

1809693-WebTechnology- Assessment2_Report

by Mark Huntly

FILE	11495_MARK_HUNTLY_1809693-WEBTECHNOLOGY-ASSESSMENT2_REPORT_241229_695099945.DOCX (46.03K)		
TIME SUBMITTED	05-MAY-2020 10:31AM (UTC+0100)	WORD COUNT	1576
SUBMISSION ID	126250778	CHARACTER COUNT	9171

This report looks at the topic of website security, the types of attacks that can happen and how to mitigate these.

Website Security

ACCA5022 – Web Technologies

1809693 - Mark Huntly
Web Development
UWTSD School of Applied Computing

Table of Contents

1.0	Introduction	2
2.0	Website Security	2
2.1	Reasons for website attacks	2
2.1	Security Issues	3
2.2	Current Security Measures Taken	3
2.3	Future Security Measures.....	4
3.0	Conclusion	4

1.0 Introduction

Websites are vulnerable to constant attacks in many different guises. Sucuri.net define website security to be “ *the measures taken to secure a website from cyberattacks*”. [1] This report looks at the topic of website security, the types of attacks that can happen and how to mitigate these.

2.0 Website Security

By its nature, it is important to see website security as an ongoing website management process that needs to be monitored and reviewed constantly. To understand the importance of this topic, and how to secure websites, it is important to firstly understand why websites are breeched, penetrated or hacked, and how they are attacked.

2.1 Reasons for website attacks

The goals vary for hacking websites, but these include

- **Exploitation of visitors to the site**

Being able to exploit visitors to the site for multiple reasons including signing to new services, downloading malware or trojans, redirecting visitors for reasons such as fraud and social engineering is common for this style of attack. This can be done through duplicating websites or urls, or pop-ups asking for click through or download.

- **Information theft from the servers**

The ability to be able to hack into the server-side through code injections and cross-site scripting gives the opportunity to attack website vulnerabilities and steal information from databases and servers. This has been seen through well documented public data breaches over the years including companies like Sony [2], Ashley Madison [3], Adobe, Ebay, Marriott [4] and Facebook [5] .

- **Bots, web crawlers and scrapers**

Web crawlers, also known as spider bots or just spiders, are both good and bad. These crawlers are used every day by search engines to create search content and website indexing and are good for search engine optimisation (SEO). There is, however, another side to this. Scrapers can steal and duplicate website content and this content could be used to intercept web traffic to exploit visitors or sabotage e-commerce websites with cart abandonment or false leads for example. [6] [7] [8] [9] [10]

- **Abuse of server resource**

With the rise in service-to-service communications, and their reliance on server-side requests (SSR's), it can leave server resources vulnerable. This can lead to attacks such as 'Denial of Service' or 'Man in the Middle', information loss or theft and forgery. [11] [12]

- **Sabotage**

Sabotage can be done in any of the above ways and for the same reasons. A common one is black hat SEO and content stuffing for example as well as 'Denial of Service' and 'Man in the Middle'.

There are vulnerabilities that can be found within websites that can be taken advantage of for any of the above reasons.

2.1 Security Issues

Prevention is better than patching when it comes to website security and to be able to prevent attacks, it is useful to understand the vulnerabilities and security issues and to factor these in at the planning stage. Some of the most common vulnerabilities to consider are:

- **Injection flaws**

This happens when data can be entered without any filtering. Unfiltered data could be passed to the database through SQL injection, through the lightweight directory access protocol (LDAP), LDAP injection and through the browser with cross-site-scripting (XSS) using Javascript or HTML injections.

- **XSS**

This method is widespread and takes injection on a step, more commonly using Javascript tag injections to get a user's browser to execute the command to craft a link for the user to click on as an example sending cookie information to an attacker when the script is run on page load.

- **Broken or bad authentication**

There are multiple problems that can stem from this vulnerability:

- i. The url may contain a session id.
- ii. Passwords may not be encrypted either in transit or storage.
- iii. Session fixation or session hijacking.

- **Security Misconfiguration**

These misconfigurations could be:

- i. Running a web app in production with debug enabled.
- ii. Outdated software, such as well documented outdated wordpress plugins for example.
- iii. Keeping default keys or passwords.

- **Confused Deputy Attack**

This occurs when a browser is fooled by another party into misusing its authority, allowing it to do something for an attacker. This can be done using 'cross site request forgery' through HTTP get request for example. [13] [14] [15] [16]

2.2 Current Security Measures Taken

There have been some security measures considered and implemented to the 'Legere' website created. The three main ones are

- **Prepared statements using PHP**

This has been implemented to filter user input and prevent SQL injection allowing users to potentially add, change or delete data or hijack authentication to gain authorisation.

- **Password hashing**

This has been implemented to prevent sensitive password information from being seen and hijacked to gain authorisation. The implementation used currently is MD5 hashing. Whilst recognising this isn't the most secure method of hashing, it does allow for password security and can be changed in the future for a higher-level hashing method to protect sensitive information in transit and at rest.

- **Hidden text field**

This text field has been implemented to trap a spam bot as it is a hidden field, not visible to a human user, but it is an input tag that a bot would pick up, therefore if this text input is filled in, it is clear that a bot has entered the details. In this situation, the session will close and the link to the database will close for extra protection. This has been implemented in both the registration page and the login page.

Furthermore, the Hyper Text Transfer Protocol (HTTP) 'POST' method has been used as opposed to the 'GET' to protect data in transit a little more as the data will never appear in the url, be cached or remain in the browser history.

2.3 Future Security Measures

For the website to go live, further security measures will be considered. The first is to ensure that the page is secure, encrypting data through the secure socket layer (SSL) and sending the data using HTTPS protocol settings. This will protect user information being visible stopping the information from being seen in urls, hashing all data in transit protecting it from man in the middle attacks. It is also prudent to improve filtering of data to further protect against 'second order SQL injection'. [13] The next would be to sanitise the HTML so as not to return tags to the user. This can be done through regular expressions and will prevent HTML injections and XSS. To help with authentication, consideration should be given to a framework to eliminate input mistakes and broken authentication. [13] [14] [15]

3.0 Conclusion

Website security is a huge and very important topic to consider. This should be implemented from the planning and outset of the project as it should be an integrated part of the development process. It is good to understand the possible threats to a website and the internet in general and to ensure vulnerabilities are closed to attackers. Website security should also be an ongoing concern with continuous monitoring and development.

Ongoing development within this subject would be to further research PHP and server-side development and the threats to this within internet security, authorisation, and authentication.

Bibliography

- [1] "Website Security - How to Secure & Protect Your Website," Sucuri Inc, 12th November 2019. [Online]. Available: <https://sucuri.net/guides/website-security/>. [Accessed 8th April 2020].
- [2] K. Zetter, "Sony Got Hacked Hard: What We Know and Don't Know So Far," Wired, 12 March 2014. [Online]. Available: <https://www.wired.com/2014/12/sony-hack-what-we-know/>. [Accessed March 2020].
- [3] Z. Doffman, "Ashley Madison Hack Returns To 'Haunt' Its Victims: 32 Million Users Now Watch And Wait," Forbes, 1st February 2020. [Online]. Available: <https://www.forbes.com/sites/zakdoffman/2020/02/01/ashley-madison-hack-returns-to-haunt-its-victims-32-million-users-now-have-to-watch-and-wait/#2a8db2605677>. [Accessed April 2020].
- [4] D. Swinhoe, "The 15 biggest data breaches of the 21st century," CSO, 17th April 2020. [Online]. Available: <https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html>. [Accessed April 2020].
- [5] M. Murphy, "Millions of Facebook user records exposed in data breach," The Telegraph, 3rd April 2019. [Online]. Available: <https://www.telegraph.co.uk/technology/2019/04/03/millions-facebook-user-records-exposed-data-breach/>. [Accessed April 2020].
- [6] "Bad RDF Crawlers," W3C, 23rd June 2011. [Online]. Available: https://www.w3.org/wiki/Bad_RDF_Crawlers. [Accessed April 2020].
- [7] "Think before you write Semantic Web crawlers," W3C, January 2020. [Online]. Available: <https://lists.w3.org/Archives/Public/public-lod/2011Jun/0433.html>. [Accessed April 2020].
- [8] "What Bad Bots Do (and How You Can Stop Them)," GlobalDots, 10th May 2019. [Online]. Available: <https://www.globaldots.com/blog/what-bad-bots-do>. [Accessed April 2020].
- [9] T. Haas, "Web Crawler 101: What Is a Web Crawler and How Do Crawlers Work?," WebFX, 11th July 2019. [Online]. Available: <https://www.webfx.com/blog/internet/what-is-a-web-crawler/>. [Accessed April 2020].
- [10] "The Difference Between Good Bots and Bad Bots," Radware, [Online]. Available: <https://www.shieldsquare.com/good-bots-and-bad-bots/>. [Accessed April 2020].
- [11] G. Pellegrino, . O. Catakoglu, D. Balzarotti and . C. Rosso, "Uses and Abuses of Server-Side Requests," in *Lecture Notes in Computer Science. Research in Attacks, Intrusions, and Defenses. RAID 2016*, vol. 9854, SpringerInternational, 2016, pp. 393-414.
- [12] J. Lindemann, "Towards abuse detection and prevention in IaaS cloud computing," in *10th International Conference on Availability, Reliability and Security*, Harvard, 2015.
- [13] G. Kalman, "10 Most Common Web Security Vulnerabilities," Toptal, [Online]. Available: https://www.toptal.com/security/10-most-common-web-security-vulnerabilities#CommonMistake_XSS. [Accessed April 2020].

- [14 R. Maurya, "Research Challenges and Issues in Web Security," [Online]. Available:
] https://www.academia.edu/34606795/Research_Challenges_and_Issues_in_Web_Security.pdf
. [Accessed April 2020].
- [15 A. K. Dalai and S. K. Jena, "Evaluation of web application security risks and secure design
] patterns," in *Proceedings of the 2011 International Conference on Communication, Computing
& Security, ICCCS 2011*, Odisha, 2011.
- [16 "OWASP Top Ten," 2017. [Online]. Available: <https://owasp.org/www-project-top-ten/>.
] [Accessed March 2020].

1809693-WebTechnology-Assessment2_Report

GRADEMARK REPORT

FINAL GRADE

71 / 100

GENERAL COMMENTS

Instructor

What? No campervans?! :)

Database - Generally good. It would be better if the membername in the reviews was actually memberid - it's more efficient to join on the primary key which is usually a number.

7/10

Range of coding (eg joins, searches, admin tools)

A little bit strange on accessing the review after searching - it takes me to the review drop-down and I have to choose the title again. You could just go straight to `addreview.php` and pass a parameter. I did laugh at the "long old book" review for *A Tale Of Two Cities*!

The searches provided are OK - it would be good if you'd use a wildcard, so I could search for "Rowling" as an author for example, rather than having to get an exact "J.K. Rowling"

The sign up routine needs a bit of validation - if I just hit submit it creates a blank account.

Edit and Delete facilities are there (it may not be a good idea to delete a user, especially if they've left a review for a book as you'd leave an orphan record.) Good range of facilities eg change picture, though a bit more validation on Edit would be good so I can't edit to a blank profile.

There's something a bit awry in the leave a review section, might need a bit more testing here. (It works if I get to it one way, but not from another way) Also be aware if you edit the username, the reviews will no longer join (using the primary key for joining as stated above would prevent this) Overall all though, a good effort.

22/30

Overall Integration - Be careful on the language used "Check a book" made me think it was withdrawing from a library or similar, but it was actually search. "Search for a book a lot easier". Overall I like your layout and colour choices, just make sure everything lines up eg the link back to home on the search results. 8 /10

Report

Issues identified/How issues relate to your website.

You have identified some security issues and stated what you have done to mitigate them. Is your website susceptible to some of the other measures you mentioned? Overall though, a good attempt to analyse the security of your site.

20/30

Total 57/80

Overall - 71%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
