California State University, San Bernardino
CSUSB ScholarWorks

Electronic Theses, Projects, and Dissertations | Office of Graduate Studies

12-2021

CYBER SECURITY VULNERABILITY ASSESSMENT IN LEARNING MANAGEMENT SYSTEMS

Mohammad Rabie

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd

 Part of the Management Information Systems Commons, and the Technology and Innovation Commons

Recommended Citation

Rabie, Mohammad, "CYBER SECURITY VULNERABILITY ASSESSMENT IN LEARNING MANAGEMENT SYSTEMS" (2021). *Electronic Theses, Projects, and Dissertations*. 1376.
https://scholarworks.lib.csusb.edu/etd/1376

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

CYBER SECURITY VULNERABILITY ASSESSMENT IN LEARNING MANAGEMENT SYSTEMS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information Systems and Technology:

Cyber Security

---

by

Mohammad Hassan Rabie

December 2021

CYBER SECURITY VULNERABILITY ASSESSMENT IN LEARNING MANAGEMENT SYSTEMS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Mohammad Hassan Rabie

December 2021

Approved by:

Dr. William Butler, Committee Chair

Dr. Conrad Shayo, Committee Member

Dr. Javad Varzandeh, Department Chair

© 2021 Mohammad Hassan Rabie

ABSTRACT

With online learning becoming in high demand to deliver training and education during the COVID-19 pandemic, cybercriminals have more opportunities to take advantage of vulnerable Learning Management Systems to steal information like training materials, and students' private information, or they try to make easy money by deploying ransomware. Regardless of the cybercriminal motivation, the compromised system has consequences on the organization that affects it financially, legally, and reputationally. This requires the organization to invest in choosing the most secure LMS and apply the required security controls to avoid such consequences that may cost them much more than expected.

This project highlights the vulnerabilities that are found in a selected list of Learning Management Systems. This may help organizations in the selection phase of their LMS, and also blue teams can use this project's result to harden their systems.

ACKNOWLEDGEMENTS

I would like to thank my family for their support during the master's journey, and especially during this project, also I highly appreciate the guidance that I received from my supervisor Dr. William Butler, and Professor Conrad Shayo.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Problem Statement

Online learning became the main delivery channel of education and training, so maintaining the security of the Learning Management System is essential through finding and fixing the vulnerabilities. This assessment should be conducted before launching and publishing the LMS to the public, however, the vulnerability assessment is an ongoing process that should discover any newly released vulnerabilities due to outdated versions of software or after any configuration change. Other studies have covered the vulnerability assessment in generic Content Management Systems, however, fewer studies have been conducted on vulnerabilities in Learning Management Systems specifically.

Purpose Statement

The purpose of this quantitative study is to determine the common vulnerability score of each LMS set to the default? In addition, what are the common and unique vulnerabilities in the different Learning Management Systems such as Moodle (*About Moodle*), SAP Litmos (*Litmos LMS: Learning Management System* 2021), and TalentLMS (*Talent LMS - About us* 2021) set to the default configuration? The assessment will be based on the Open Web

Application Security Project (OWASP) top 10 vulnerabilities in web applications, version 2017 (current version).

By answering the above questions, the education/training organization will be able to choose the best LMS according to their security measures, and they will be able to make a remediation plan to harden the security of their LMS.

CHAPTER TWO

LITERATURE REVIEW

Common Vulnerabilities and Exposures

Common vulnerabilities and exposure (CVE) gives a uniqe names to known vulnerabilities. The objective of CVE is to facilitate sharing information over different databases and make available a common platform to evaluate security tools. Security researchers scan the applications to find vulnerabilities, and when they find a new vulnerability they give a unique identifier to each one to help the security analysts to deal with them. The format of the CVE consists of three portions: the first one is fixed “CVE”, the second one is the year of release, and the third one is a serial number like CVE-2020-11023 as an example of a vulnerability in jQuery versions greater than or equal to 1.0.3 and before 3.5.0 that may cause execution of an untrusted code. CVE makes a unique definition of each vulnerability to allow sharing this information between tools and services, when a new vulnerability is discovered it is assigned an ID according to the CVE Numbering Authority (CAN) that writes a description and references, and then this information is posted on CVE website, the description includes the software versions that are affected and the impact of the vulnerability CVE is designed to allow vulnerability databases. The US Department of Homeland Security funded MITRE to copyright the CVE list for the benefit of the community to assure that

this database is available for everyone as an open-source through their website https://cve.mitre.org.

Table 1. Comparison Between Vulnerability and Exposures (CVE Explained 2019)

Vulnerability	Exposures
Allows the hacker to intrude a system or network due to an error in the software code.	Make the data accessible to the attacker to be misused or sold.
Allows the hacker to execute commands with unauthorized permissions.	Facilitate data gathering activities for the attacker.
Allows the hacker to get information that is restricted.	Allows the hacker to conceal activities.
Allows the hacker to act like another entity.	Is considered as the main entry point by an attacker to access the information.
Allows the hacker to deny service.	Is an issue in the security policy.

CVE community. Below is a list of the major contributors to the CVE community, according to beyondsecurity.com - CVE Explained 2019:

- CVE board – The CVE Board incorporates individuals from various cybersecurity-related associations globally, like government offices, research organizations, and other security specialists. Through open discussions, the board decides the entries on the CVE List.
- CVE sponsor – US-CERT sponsors CVE at the U.S. Department of Homeland Security. The sponsors' page consists of all the past sponsors.
- CVE Numbering authorities – CVE numbering authorities (CNAs) allocate CVE identifiers to newly found problems without including MITRE.
- CVE-compatible products and services – various organizations globally have incorporated CVE identifiers to make their cybersecurity products and services "CVE-compatible".

Common Vulnerability Scoring System

Common Vulnerability Scoring System (CVSS) is an open framework that was launched in 2005 to measure the characteristics and severity of CVE's. It is considered a universal language so cybersecurity admins can understand the severity level of each software they deal with, then they can prioritize their remediation plans. This scoring system is ranged from 0 to 10, where 10 is the most severe.

CVSSv3. Common Vulnerability Scoring System version 3 (current version) started being used in 2016 (*Vulnerability Metrics. NATIONAL VULNERABILITY DATABASE*). There are three metrics to calculate CVSSv3:

1. The base metrics: exploitability and impact regardless of time and place.
2. The Temporal metrics: adjustment based on the current situation like if there is a workaround available.
3. Environmental metrics: based on the deployment of the software or hardware.

Some of these metrics are objective like, “Does the exploitation need user credentials?”, and some metrics are subjective like, “it is easy to exploit that vulnerability?” The base and temporal metrics are calculated by someone who should be knowledgeable about the vulnerable system, usually the author of that software or the one who found this vulnerability. On the other hand, the environmental metrics are calculated by someone who knows how that software is deployed, that’s why the environmental metrics may vary from one customer to another.

Table 2. Vulnerability Severity Levels

Severity Level	Score Range
Critical	9 - 10

High	7.0 – 8.9
Medium	4 – 6.9
Low	0.1 – 3.9
Information	0

OWASP Top 10 Vulnerabilities

Open Web Application Security Project (OWASP) of the Top 10 Web Application Security Risks. This report is released usually every three to four years. The current version is 2017, and the next version will be released later in 2021. The top 10 security risks are based on vulnerabilities gathered from thousands of web applications and ranked based on their exploitability, detectability, and impact on organizations (*OWASP Top Ten*). A list of the top 10 is sorted below:

1. Injection. Injection flaws, such as SQL, NoSQL, OS, and LDAP injection.
2. Broken Authentication. Allowing attackers to compromise passwords, keys, or session tokens.
3. Sensitive Data Exposure. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.

4. XML External Entities (XXE). External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
5. Broken Access Control. Restrictions on what authenticated users are allowed to do are often not properly enforced.
6. Security Misconfiguration. insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information.
7. Cross-Site Scripting (XSS). XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites.
8. Insecure Deserialization. Insecure deserialization often leads to remote code execution.
9. Using Components with Known Vulnerabilities. Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application.
10. Insufficient Logging & Monitoring. Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

Vulnerability Scanning Tools

Many tools automate the vulnerability scanning process. Some of them are open-source and some are commercial tools. Each tool has a library of plugins that define the VCE's, and the function of these tools is to scan the target system using the predefined plugins to detect what vulnerabilities exist in the target system. In this section, we will introduce some of these tools.

Nmap.

It is a free tool to scan and IP address or URL, the first version of Nmap was released in 1997.

Legion

Legion is a semi-automated penetration testing tool, it works in the reconnaissance and exploitation phases, it is open-source with a graphical user interface, it can discover CVEs and it allows scheduled Scripting. The scan target can be either an IP address or a domain name / URL, also the user can specify a range or multiple targets with a parallelization feature to save the time of the multi-target scan. The user can also control the scan speed to be as fast as possible or sneaky to not be detectable by the target Intrusion Detection System. The scan configuration can be exported and edited manually to be used in a scheduled script.

OpenVAS

OpenVAS stands for Open Vulnerability Assessment System (*Open Vulnerability Assessment Scanner*) which is the open-source framework tool that

is developed by Green Bone Networks. It uses Nessus attack scripting language, and has almost all features of Legion in addition to reporting feature.

Tenable.io

Tenable.io (*Tenable.io* 2021) is a commercial cloud-based product of Nessus company. The tool being Software-as-a-Service makes it immediately updated with the latest plugins and CVEs. This is a critical feature to detect the most recent and even zero-day vulnerabilities.

Learning Management System (LMS)

LMS is a web application that is used by education institutes, and other organizations to manage the whole learning process and deliver e-Learning material, and it can be even used by any organization to deliver training for internal employees, LMS has a lot of benefits in terms of cost reduction, flexibility, and mobility, especially during the COVID-19 pandemic. There are many Learning Management Systems. We will select three of them which are widely used to do our study. The three LMS's are: Moodle, SAP Litmos, and TalentLMS.

CHAPTER THREE

METHODOLOGY

This is a quantitative experimental study. To answer the problem statement questions, we will build a virtual lab composed of three different LMS's, and conduct vulnerability assessments using a 3rd party tool (data collection), and identify the unique and common vulnerabilities between the different LMSs (Analysis), then identify the recommended configuration to harden the LMS security (conclusion).

Our Virtual Lab

The lab consists of three web applications (Learning Management Systems). The three LMS's that we have chosen are Moodle, SAP Litmos, and TalentLMS. In addition to the vulnerability scanning tool (Tenable.io)

Moodle

Moodle is an open-source Learning Management System, it's one of the oldest and widely used LMS. The first version of Moodle was released in August 2002. The number of registered users exceeds 190 million in 2020 (*Moodle Documentation*).

SAP Litmos

SAP Litmos was founded in 2007. Litmos was acquired by CallidusCloud in 2011 and acquired again by SAP in 2018. It is one of the most reliable Learning Management Systems, used in 150 countries, and supports 35 languages (*Litmos LMS: Learning Management System* 2021).

TalentLMS

TalentLMS was released in 2012. Because of its ease of deployment, there are 11 millions students around the world who use TalentLMS (*Talent LMS - About us* 2021).

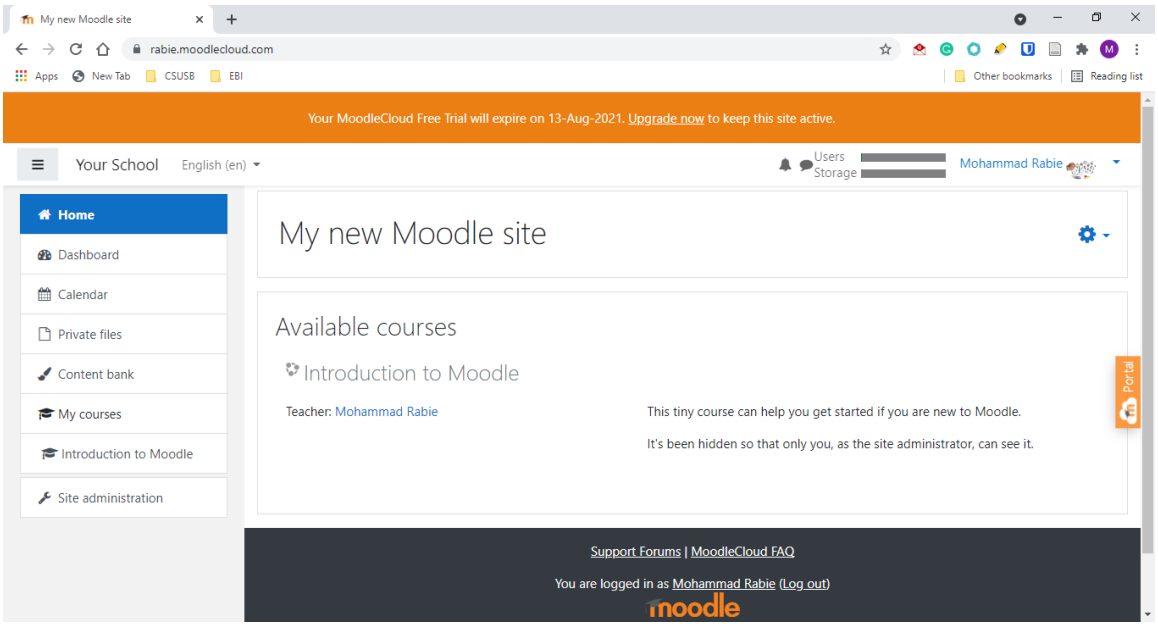

Figure 1. My Moodle Site

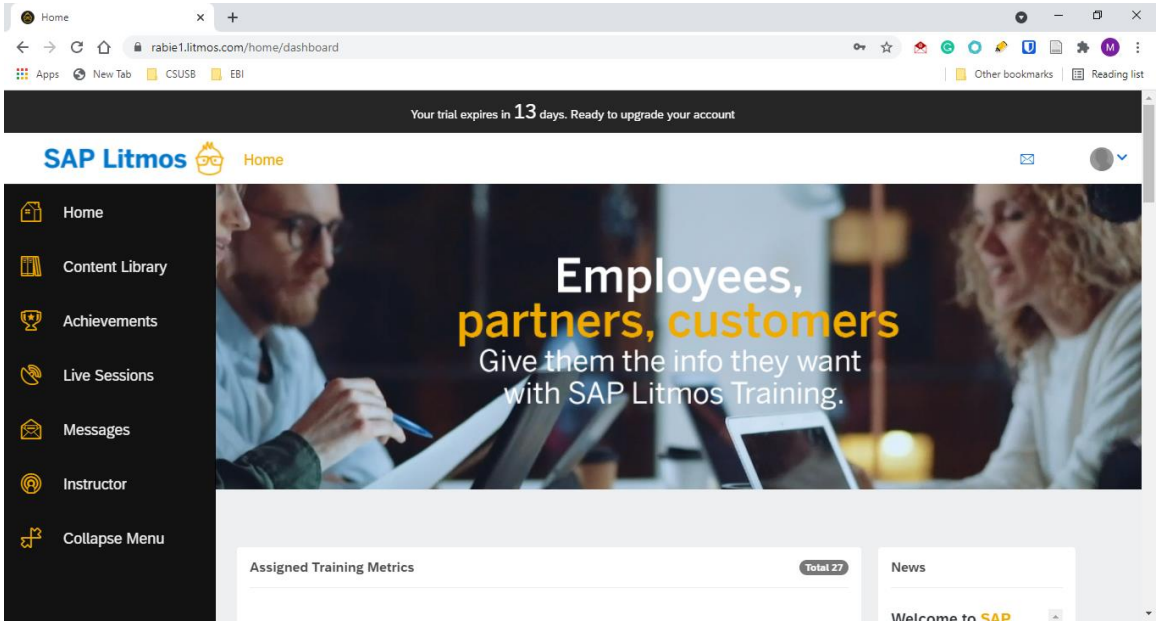

Figure 2. My SAP Litmos Site

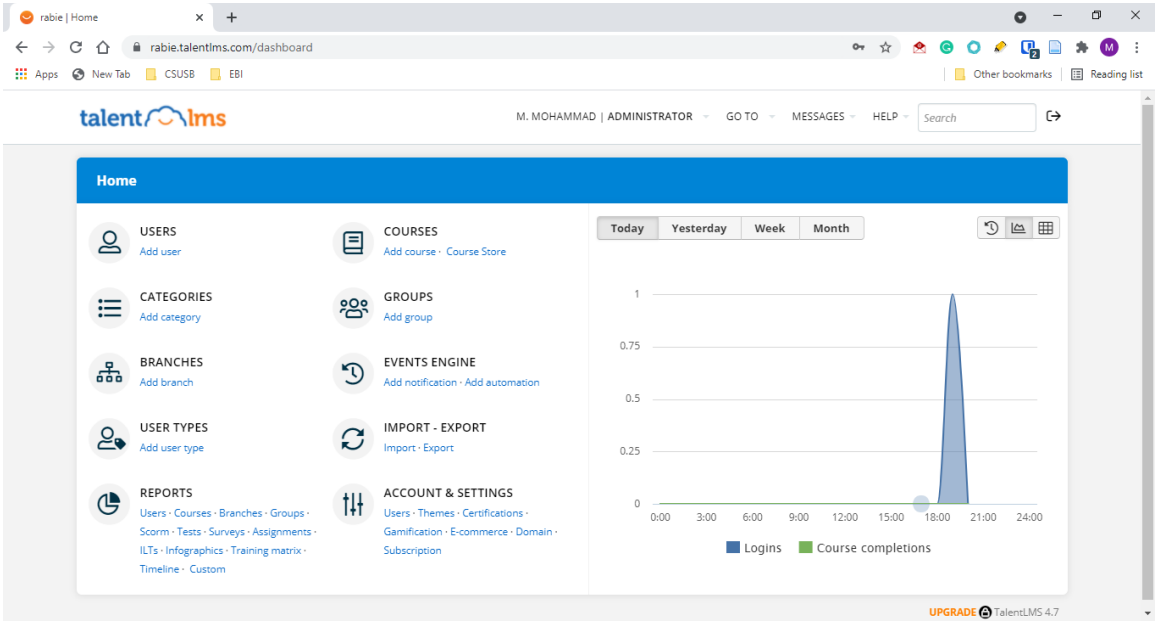

Figure 3. My TalentLMS Site

Hypothesis Testing

This is the hypothesis testing: H1 is the opposite of H0 and if proven DISPROVES the null hypothesis (H0).

H01 LMS #1 displays MORE Common Vulnerability Score than LMS#2 or LMS#3.

H1 LMS #1 displays LOWER Common Vulnerability Score than LMS#2 or LMS#3.

H02 LMS #2 displays MORE Common Vulnerability Score than LMS#1 or LMS#3.

H2 LMS #2 displays LOWER Common Vulnerability Score than LMS#1 or LMS#3.

H03 LMS #3 displays MORE Common Vulnerability Score than LMS#1 or LMS#2.

H3 LMS #3 displays LOWER Common Vulnerability Score than LMS#1 or LMS#2.

CHAPTER FOUR

RESULTS

According to the National Institute of Standards and Technology, the Common Vulnerability Scoring System (CVSS) is a framework to measure the severity of a software security vulnerability. The National Vulnerability Database provides a score for almost all known vulnerabilities (*Vulnerability Metrics. NATIONAL VULNERABILITY DATABASE*).

Moodle Vulnerabilities

According to the Common Vulnerability Scoring System version 3 (CVSSv3), the overall Common Vulnerability Score is 92.4. The following table is a list of discovered vulnerabilities sorted by severity level. And the number of instances refers to the number of pages that the same vulnerability exists.

Table 3. Moodle Vulnerabilities

Severity	Name	Family	Instances
Medium	Cross-Site Request Forgery	Cross Site Request Forgery	4
Medium	jQuery 1.2.0 < 3.5.0 Cross-Site Scripting	Component Vulnerability	1
Medium	Missing HTTP Strict Transport Security Policy	HTTP Security Header	1

Low	Cookie Without SameSite Flag Detected	Web Applications	5
Low	Cookie Without HttpOnly Flag Detected	Web Applications	5
Low	Cookie Without Secure Flag Detected	Web Applications	4
Low	HTTP Header Information Disclosure	HTTP Security Header	1
Low	Missing 'Expect-CT' Header	HTTP Security Header	1
Low	Missing 'X-Content-Type-Options' Header	HTTP Security Header	1
Low	Missing Content Security Policy	HTTP Security Header	1
Low	Missing 'Cache-Control' Header	HTTP Security Header	1

SAP Litmos Vulnerabilities

According to the Common Vulnerability Scoring System version 3 (CVSSv3), the overall Common Vulnerability Score is 20.1. The following table is a list of discovered vulnerabilities sorted by severity level. And the number of instances refers to the number of pages that the same vulnerability exists.

Table 4. SAP Litmos Vulnerabilities

Severity	Name	Family	Instances

Medium	Cross-Site Request Forgery	Cross Site Request Forgery	1
Medium	HTTP to HTTPS Redirect Not Enabled	SSL/TLS	1
Low	Cookie Without SameSite Flag Detected	Web Applications	3

TalentLMS Vulnerabilities

According to the Common Vulnerability Scoring System version 3 (CVSSv3), the overall Common Vulnerability Score is 62.6. The following table is a list of discovered vulnerabilities sorted by severity level. And the number of instances refers to the number of pages that the same vulnerability exists.

Table 5. TalentLMS Vulnerabilities

Severity	Name	Family	Instances
Medium	Permissive HTTP Strict Transport Security Policy Detected	HTTP Security Header	1
Medium	jQuery 1.12.4 < 3.0.0 Cross-Site Scripting	Component Vulnerability	1
Medium	jQuery 1.2.0 < 3.5.0 Cross-Site Scripting	Component Vulnerability	1
Medium	jQuery < 3.4.0 Prototype Pollution	Component Vulnerability	1

Low	Cookie Without SameSite Flag Detected	Web Applications	3
Low	Cookie Without HttpOnly Flag Detected	Web Applications	2
Low	Cookie Without Secure Flag Detected	Web Applications	2
Low	Missing 'X-Frame-Options' Header	HTTP Security Header	1
Low	HTTP Header Information Disclosure	HTTP Security Header	1
Low	SSL/TLS Weak Cipher Suites Supported	SSL/TLS	1
Low	Missing Content Security Policy	HTTP Security Header	1
Low	Missing 'Expect-CT' Header	HTTP Security Header	1

The Common and Unique Vulnerabilities

The following table shows the number of instances of each vulnerability in each LMS, and we can find which vulnerabilities are common in more than one LMS and which vulnerabilities are unique in one LMS.

Table 6. The Common and Unique Vulnerabilities

Vulnerability Name	Number of Instances

	Moodle	SAP Litmos	TalentLMS
Cross-Site Request Forgery	4	1	
jQuery 1.2.0 < 3.5.0 Cross-Site Scripting	1		1
Missing HTTP Strict Transport Security Policy	1		
Cookie Without SameSite Flag Detected	5	3	3
Cookie Without HttpOnly Flag Detected	5		2
Cookie Without Secure Flag Detected	4		2
HTTP Header Information Disclosure	1		1
Missing 'Expect-CT' Header	1		1
Missing 'X-Content-Type-Options' Header	1		
Missing Content Security Policy	1		1
Missing 'Cache-Control' Header	1		
HTTP to HTTPS Redirect Not Enabled		1	
Permissive HTTP Strict Transport Security Policy Detected			1
jQuery 1.12.4 < 3.0.0 Cross-Site Scripting			1
jQuery < 3.4.0 Prototype Pollution			1
Missing 'X-Frame-Options' Header			1
SSL/TLS Weak Cipher Suites Supported			1

Description of Each Discovered Vulnerability

Missing HTTP Strict Transport Security Policy

The HTTP protocol is clear text, which means that any data transferred using HTTP protocol can be intercepted by cybercriminals using the “Man in the middle” technique. To keep data private and encrypted, HTTP is often tunneled through either Secure Sockets Layer (SSL) or Transport Layer Security (TLS), which is referred to as HTTPS.

HTTP Strict Transport Security policy is an optional header that instructs the browser to only communicate through HTTPS that the browser enforces even if the user tried to use HTTP. The scanner discovered that the affected application is using HTTPS however does not use the HSTS header.

Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) occurs when an authenticated user is tricked into clicking on a link that would automatically submit a request without the user’s consent. An anti-CSRF token can be used to prevent this, that token is generated each time the request is intiated and expires when the request is submitted, the web application backend can use the anti-CSRF token technique to verify of that the request is ligitmate.

Cross-Site Request Forgery implies different factors:

- Sensitive action is perfeormed.
- The victim must have an active session.
- The victim click on a malicious link to send the.

The source code of the web application contains a request, available only to authenticated users that may perform a sensitive action, such as reset a password, modify user profiles, post content on a forum, etc.. which is increases the likelihood of CSRF vulnerability.

jQuery Version 1.2.0 < 3.5.0 Cross-Site Scripting

Cross-Site Scripting is a known vulnerability in all versions of jQuery below 3.5.0. The scanner did not test the Cross-Site Scripting vulnerability, but it relied only on the self-reported version number of jQuery.

Cookie Without Secure Flag Detected

The “Secure” flag instructs the web browser to send a cookie over an encrypted HTTPS tunnel instead of HTTP.

Although the initial connection was HTTPS, the existence of a cookie without a secure flag may cause an unencrypted transmission of cookies in the case of an HTTP link to the same server. The risk of this vulnerability depends on the sensitivity of the information contained in this cookie.

Missing ‘Expect-CT’ Header

The Expect-CT header allows sites to opt into reporting and or enforcement of Certificate Transparency requirements, which prevents the use of wrong certificates for that site from going unnoticed.

HTTP Header Information Disclosure

The HTTP header of the webpage includes sensitive information about the webserver, like server version and technologies. An attacker can use this information in the reconnaissance stage.

Missing ‘X-Content-Type-Options’ Header

The non-existence of ‘X-Content-Type-Options’ header puts the website at risk of a cross-site scripting attack.

Missing Content Security Policy

Content Security Policy is a web security standard that helps to mitigate attacks like cross-site scripting (XSS), clickjacking, or mixed content issues. Content Security Policy restricts content that browsers will be allowed to load.

Missing’ Cache-Control’ Header

The web browser uses the HTTP ‘Cache-Control' header to specify caching mechanisms. The server did not return 'Cache-Control' header or returned an invalid 'Cache-Control' header, which means that the web browser can store a page containing sensitive information like (password, credit card, personal data, social security number, etc.). Then unauthorized persons can access sensitive information on the client-side disk.

Cookie Without SameSite Flag Detected

SameSite is an attribute that the application sets on a cookie to help prevent Cross-Site Request Forgery (CSRF) attacks.

The scanner did not find the SameSite attribute on cookies set by the application or a misconfiguration.

HTTP to HTTPS Redirect Not Enabled

HTTPS is enabled on the website; however, the application does not redirect the HTTP requests to HTTPS. Communications are not encrypted if users do not explicitly access to HTTPS version of the website.

Permissive HTTP Strict Transport Security Policy Detected

HTTP Strict Transport Security (HSTS) is a header that should be configured on the server to enforce only HTTPS communication. The detected HSTS policy either does not have a long max-age value determining the time the browser will adhere to the header policy or does not cover subdomains via the includeSubDomains directive.

jQuery Version 1.12.4 < 3.0.0 Cross-Site Scripting

According to its self-reported version number, the jQuery version is at least 1.12.4 and before 3.0.0. Therefore, it may cause a cross-site scripting vulnerability. Note that the scanner relied only on the application's self-reported version number instead of testing these issues.

jQuery < 3.4.0 Prototype Pollution

According to its self-reported version number, the jQuery version is below 3.4.0. Therefore, it may be affected by a prototype pollution vulnerability. Note that the scanner has relied only on the application's self-reported version number instead of testing these issues.

Missing 'X-Frame-Options' Header

Clickjacking is known as (user Interface redress attack). It is a malicious technique of tricking a user into clicking on a link different from what they perceive they are clicking on, thus potentially exposing confidential information or taking control of it their computer while clicking on a seemingly innocuous link.

The server did not return an `X-Frame-Options` header which means that this website could risk a clickjacking attack.

SSL/TLS Weak Cipher Suites Supported

The application supports using SSL/TLS ciphers that offer weak encryption (including RC4 and 3DES encryption).

Table 7. Number of Vulnerabilities in Each LMS by Severity Level

Severity	Moodle	SAP Litmos	TalentLMS
Critical	0	0	0
High	0	0	0
Medium	6	2	4
Low	19	3	12
Total	25	5	16

Table 8. The Common Vulnerability Score in Each LMS

	1. Moodle	2. SAP Litmos	3. TalentLMS
CVSSv3	92.4	20.1	62.6

Table 9. CVE's in Each LMS

Moodle	SAP Litmos	TalentLMS
		CVE-2015-9251
		CVE-2019-11358
CVE-2020-11022		CVE-2020-11022
CVE-2020-11023		CVE-2020-11023

Table 10. The Result of the Hypnosis Testing

Hypothesis Number	Hypothesis Description	Proven / Not Proven
H01	Moodle displays MORE Common Vulnerability Score than SAP Limos or TalentLMS	Proven
H1	Moodle displays LESS Common Vulnerability Score than SAP Litmos or TalentLMS.	Not Proven
H02	SAP Litmos displays MORE Common Vulnerability Score than Moodle or TalentLMS.	Not Proven
H2	SAP Litmos displays LESS Common Vulnerability Score than Moodle or TalentLMS.	Proven
H03	TalentLMS displays MORE Common Vulnerability Score than Moodle or SAP Litmos.	Not Proven
H3	TalentLMS displays LESS Common Vulnerability Score than Moodle or SAP Litmos.	Not Proven

CHAPTER FIVE

DISCUSSION AND RECOMMENDATIONS

Conclusion

Software as a Service (SaaS) deployment is not fully secure, organizations may need to have more control over the systems to be able to apply higher security measures, this control can be gained by going to the Platform-as-a-Service (PaaS) deployment to have control over the application, or may further control is needed by using Infrastructure-as-a-Service (IaaS) deployment that allows the organization to manage also the operating system, lastly the most manageable deployment is the On-site, which allows the organization to even manage the physical security and avoid the multi-tenant issues, that can be used by hackers to make a lateral movement from a malicious virtual machine to another virtual machine.

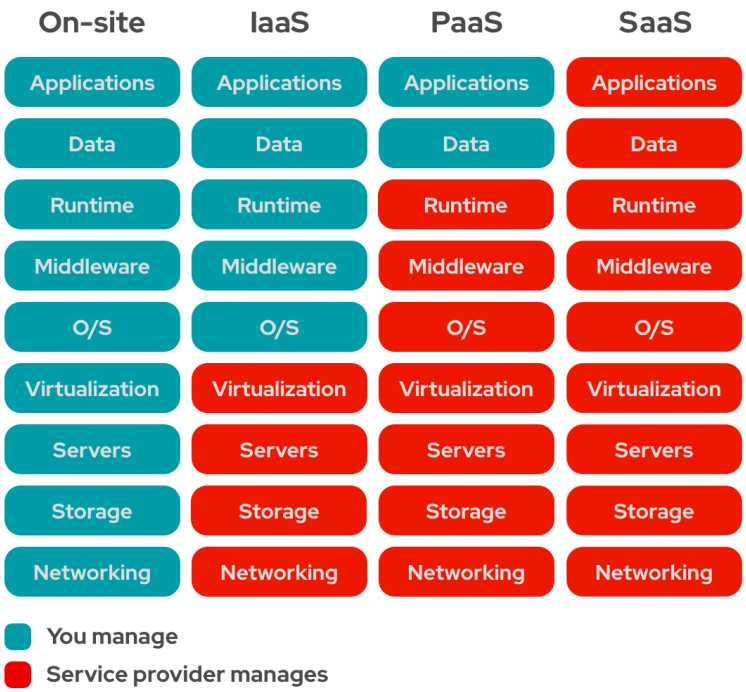

Figure 4. The Difference Between Deployment Scenarios

Moodle is an open-source application, with its default configuration, it has the highest vulnerability score which needs a lot of configurations to harden its security.

Recommended Solutions to Mitigate the Vulnerabilities

Below is a list of instructions to remediate the discovered vulnerabilities, according to Tenable's Web Application Scanning 2021.

Missing HTTP Strict Transport Security Policy

Depending on the framework being used the implementation methods will vary, however it is advised that the `Strict-Transport-Security` header be configured on the server.

One of the options for this header is `max-age`, which is a representation (in milliseconds) determining the time in which the client's browser will adhere to the header policy.

Depending on the environment and the application this time period could be from as low as minutes to as long as days.

Cross-Site Request Forgery

Update the application by adding support of anti-CSRF tokens in any sensitive form available in an authenticated session.

Most web frameworks provide either built-in solutions or have plugins that can be used to easily add these tokens to any form. Check the references for possible solutions provided for the most known frameworks.

jQuery 1.2.0 < 3.5.0 Cross-Site Scripting

Upgrade to jQuery version 3.5.0 or later.

Missing 'Expect-CT' Header

Configure your webserver to include an 'Expect-CT' header with a value of 'maxage' defined therein.

Missing 'X-Content-Type-Options' Header

Configure your webserver to include an 'X-Content-Type-Options' header with a value of 'nosniff'.

Cookie Without SameSite Flag Detected

Web browser's default behavior may differ when processing cookies in a cross-site context, making the final decision to send the cookie in this context unpredictable. The SameSite attribute should be set in every cookie to enforce the expected result by developers. When using the 'None' attribute value, ensure that the cookie is also set with the 'Secure' flag.

HTTP to HTTPS Redirect Not Enabled

Enable HTTP to HTTPS redirect for all requests. Besides redirects, if HTTP Strict Transport Security (HSTS) is not implemented it's highly recommended to enable it.

Permissive HTTP Strict Transport Security Policy Detected

The max-age must be set at least to 31536000 seconds (1 year) and includeSubDomains directive must be specified.

jQuery 1.12.4 < 3.0.0 Cross-Site Scripting

Upgrade to jQuery version 3.0.0 or later.

jQuery < 3.4.0 Prototype Pollution

Upgrade to jQuery version 3.4.0 or later.

Missing 'X-Frame-Options' Header

Configure your web server to include an `X-Frame-Options` header.

SSL/TLS Weak Cipher Suites Supported

Reconfigure the affected applicatio

Future Work

This study was limited to only three Learning Management Systems, although there are a lot more available systems, some of the suggested LMS’s

that may be studied later are Blackboard and Canvas, both systems are widely used.

REFERENCES

About Moodle. About Moodle - MoodleDocs. (n.d.). https://docs.moodle.org/311/en/About_Moodle.

Bhatia, M., & Maitra, J. K. (2018). E-learning Platforms Security Issues and Vulnerability Analysis. *2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES)*. https://doi.org/10.1109/cctes.2018.8674115

Cross Site Request Forgery (CSRF). Cross Site Request Forgery (CSRF) | OWASP Foundation. (n.d.). https://owasp.org/www-community/attacks/csrf.

CVE Explained. Beyond Security. (2019, September 26). https://beyondsecurity.com/vulnerability-assessment-requirements-cve-explained.html.

CVE. (n.d.). https://cve.mitre.org/.

E-learning Platforms Security Issues and Vulnerability Analysis. Digital Object Identifier System. (n.d.). https://doi.org/10.1109/cctes.2018.8674115.

Litmos LMS: Learning Management System. SAP Litmos. (2021, April 23). https://www.litmos.com/learning-management-system.

Moodle Documentation. History - MoodleDocs. (n.d.). https://docs.moodle.org/38/en/History.

Moodle vs SAP Litmos vs TalentLMS Comparison. GetApp. (n.d.). https://www.getapp.com/education-childcare-software/a/moodle/compare/litmos-lms-vs-talentlms/.

Nmap. (n.d.). https://nmap.org/.

Open Vulnerability Assessment Scanner. OpenVAS. (n.d.).

https://www.openvas.org/.

OWASP Top Ten. OWASP. (n.d.). https://owasp.org/www-project-top-ten/.

Talent LMS - About us. TalentLMS. (2021, April 26).

https://www.talentlms.com/about.

Tenable.io. Tenable®. (2021, May 24).

https://www.tenable.com/products/tenable-io.

Vulnerability Metrics. NATIONAL VULNERABILITY DATABASE. NVD. (n.d.).

https://nvd.nist.gov/vuln-metrics/cvss.