Development Team Project: Executive Summary (Group 4)

Brief Summary of the work performed:

The purpose of the penetration test was to find any vulnerabilities on the e-commerce website. This activity helped to give more vision about the current security risks and guide to operate the website with the most secure and safe standards. Performing the penetration test provides an understanding of security weaknesses and threats that malicious staff or unauthorised users could use to theft or alter the data within the web application and its database.

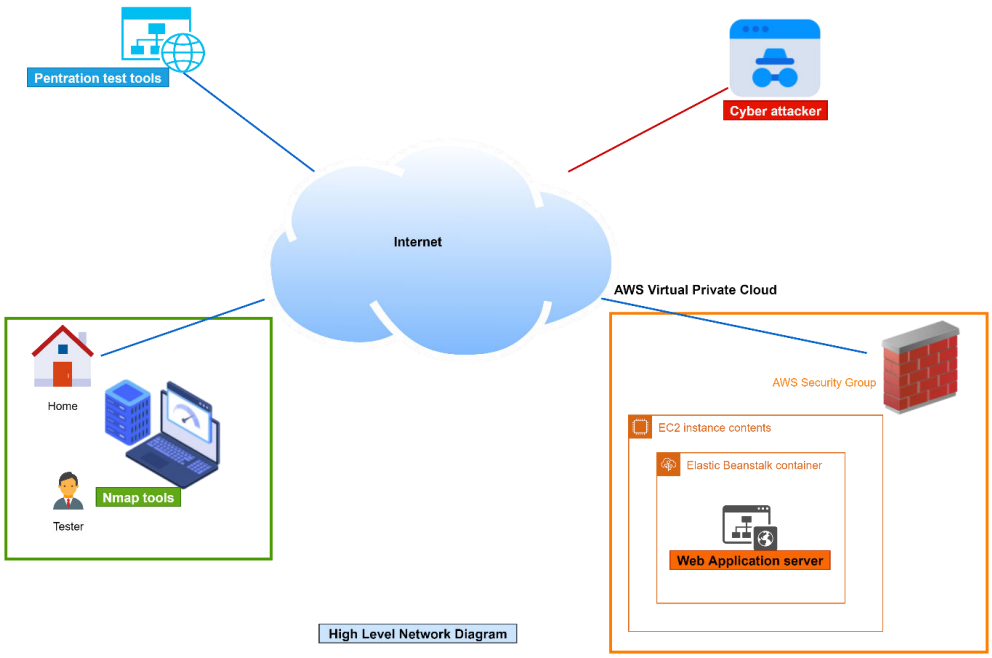
In completing a comprehensive penetration test, scanning was performed from different areas using multiple tools. First, we run tests such as fingerprinting, DNS forward, reverse lookup, DNS Zone transfer, ping and traceroute during the reconnaissance phase. Afterwards, penetration tools were used to scan the software, application, and operating system. For the server software, we used the Penetration-tools.com website to give us more information about version-based vulnerabilities detection, the insecure HTTP headers, common insecure cookie settings and deprecated server software.

ImmuniWeb Community tools were used to scan and get more information and verify the webserver security; these tools also helped us to fingerprint the CMS components and vulnerabilities. Additionally, it enabled us to check and gather more details about the GDPR and PCI DSS compliance, HTTP Headers, Content Security Policy, Cookies, and external content security tests.

In discovering the open «doors» at the network level, port scan tools like Penetration-tools.com and Nmap tools were used to detect the TCP and UDP ports available from the public internet side to the web application server. Nmap tools identified the ports and their version information. The traceroute identified the web application server and details about the host server like the host's name, IP address, uptime, and the operating system.

Some additional tools like SHHGIT and SkyArt were installed in our AWS environment. SHHGIT gave us some excellent results about secret leaks across Bitbucket, GitHub and Gitlab platforms. SkyArt was used to discover the sensitive and risky users' permission, but the limited AWS privileges given prevented us from completing the scan.

Finally, with the fingerprinting and scans performed to the website, we generated a report that includes the security weaknesses and all potential vulnerabilities that attackers and malicious staff can utilise and an analysis of the output data. Furthermore, we sorted out the vulnerabilities in the software, operating system, network access level, GDPR and DSS-PCI compliance test. Lastly, we included our recommendations and advice that can be implemented to mitigate the security risks and help to avoid any problems with the confidentiality, integrity, or availability agreements.



Findings:

After performing a detailed penetration testing against Team One's PHP e-commerce website, we identified several issues of concern. This report provides brief descriptions of each testing category and offers more details where findings were negative.

The Open Web Application Security Project (OWASP) was used to identify vulnerabilities and complicated logic flaws. We used the Common Vulnerability Scoring System (CVSS) framework to measure quantitative scores and reflect the

severity of the discovered vulnerabilities. The scores were translated into a qualitative representation (i.e., low, medium, and high) to assess and prioritise the vulnerability management process.

Below is a table showing details of the identified vulnerabilities established on category and severity of the risk. Following the table is a detailed breakdown outlining each testing category.

Table 1: Vulnerability and Security Risk

SECURITY RISK	COUNT	VULNERABILITY SEVERITY/SECURITY RISK		
		HIGH	MEDIUM	LOW
A1. Injections				
A2. Broken Authentication				
A3. Sensitive Data Exposure	1	X		
A4. Insufficient Logging & Monitoring				
A5. Insecure Deserialization				
A6. Security Misconfiguration	64			X
A7. Cross-Site Scripting (XSS)	2		X	
A8. Server-Side Request Forgery (SSRF)				
A9. Known Vulnerabilities	3		X	
A10. Broken Access Control				

A3. Sensitive Data Exposure

The website uses an insecure web protocol that transmits data in plain text without encryption. Confidentiality, one of the tenets of the CIA (Confidentiality, Integrity, Availability) triad, is a set of rules designed to ensure that information access is limited to the authorised subjects. The impact on an individual or an entity resulting

from unauthorised access to sensitive information usually determines the risk rating of the data to vary (Wesley, 2021).

Communication is not secure

URL	Evidence
http://nismphp-env.eba-2mwmqiam.us-east-1.elasticbeanstalk.com/	Communication is made over unsecure, unencrypted HTTP.

Details

Risk description:
The communication between the web browser and the server is done using the HTTP protocol, which transmits data unencrypted over the network. Thus, an attacker who manages to intercept the communication at the network level, is able to read and modify the data transmitted (including passwords, secret tokens, credit card information and other sensitive data).

Figure 1 on the finding further to assist with the understanding concerning compliance with standards and regulations.

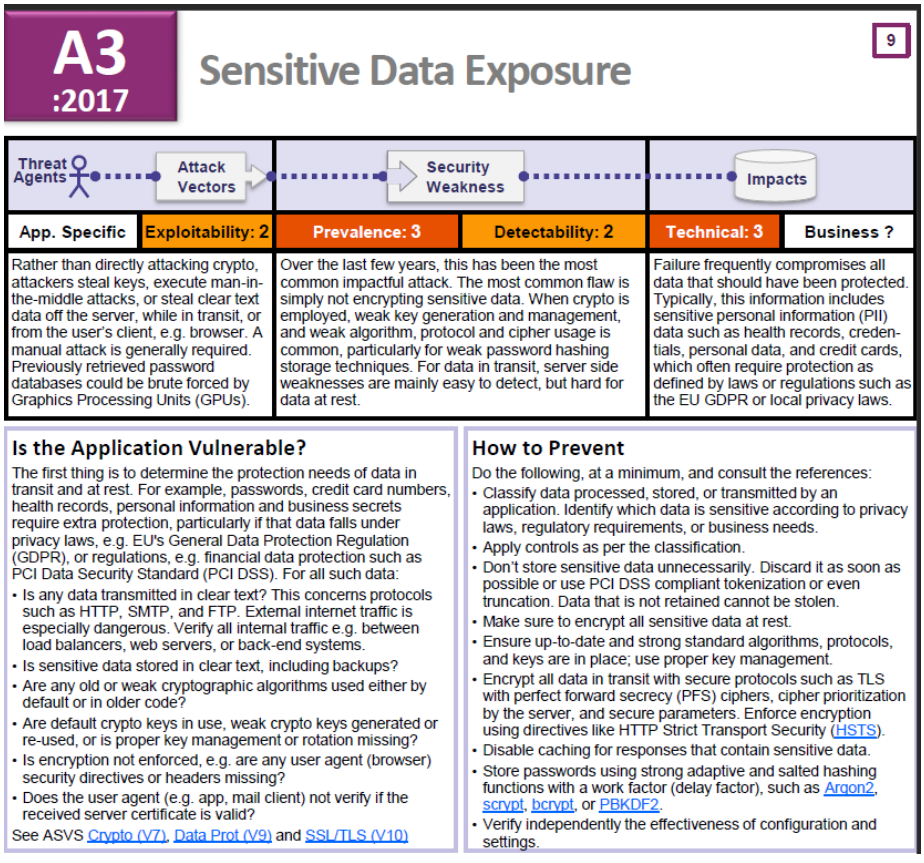


Figure 1: OWASP A3 Sensitive Data Exposure (Dehalwar et al., 2018).

A6. Security Misconfiguration

This can include a default account, unpatched or unmaintained server code, references to old versions of services, and so on. Attackers can exploit any security misconfiguration to gain access, elevate privileges, or violate the confidentiality or integrity of the data.

Missing security header: Content-Security-Policy

URL	Evidence
http://nismphp-env.eba-2mwmqiam.us-east-1.elasticbeanstalk.com/	Response headers do not include the HTTP Content-Security-Policy security header

Details

Risk description:
The Content-Security-Policy (CSP) header activates a protection mechanism implemented in web browsers which prevents exploitation of Cross-Site Scripting vulnerabilities (XSS). If the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Figure 2 below shows some of the causes of security misconfiguration:

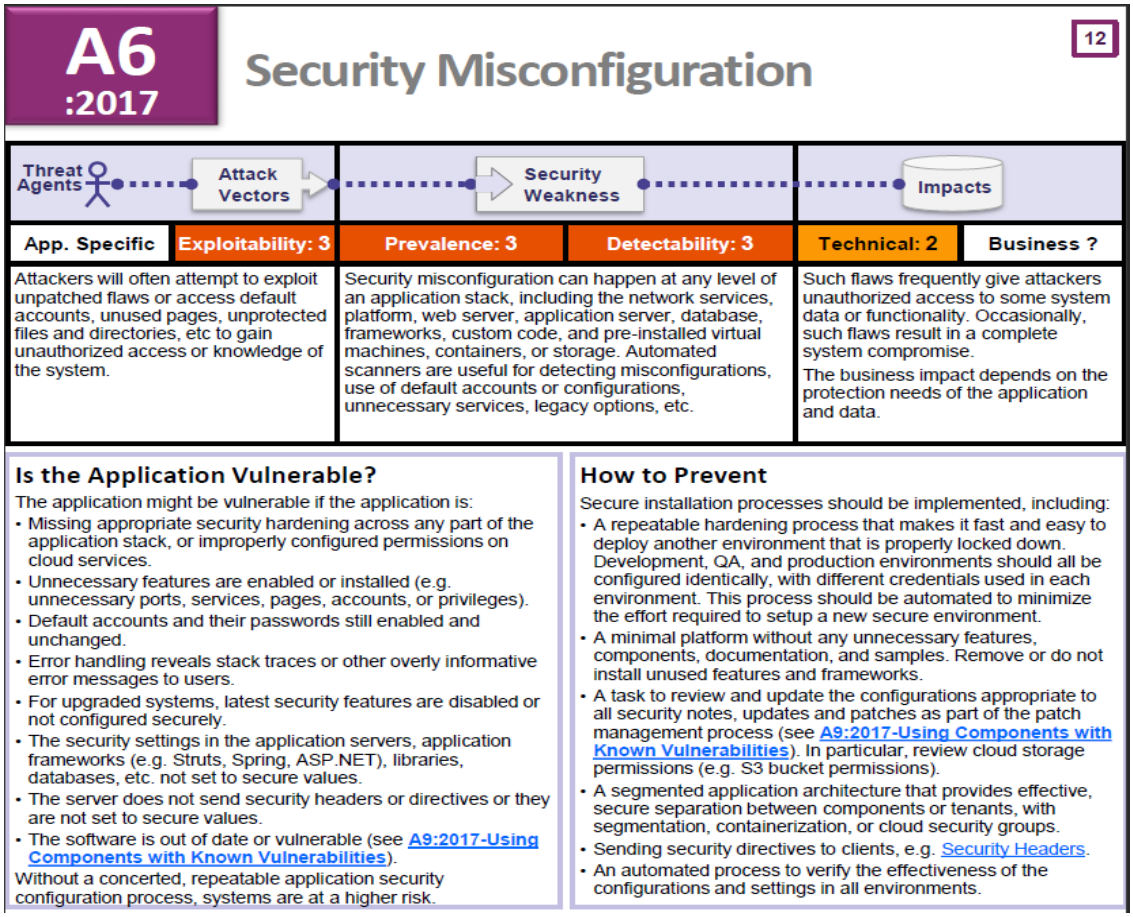


Figure 2: OWASP A6 Security Misconfiguration (Dehalwar et al., 2018).

A7. Cross-Site Scripting (XSS)

Failing of restricting the sources that are allowed to input data into the web application leads to malicious data from the victim's web browser included with dynamic content of the browser delivered to the web application. Thus, a cross-site scripting attack is successful. The usage of such attacks at other times results in defaced websites (Rosencrance, 2018).

Missing security header: X-XSS-Protection

URL	Evidence
http://nismphp-env.eba-2mwmqiam.us-east-1.elasticbeanstalk.com/	Response headers do not include the HTTP X-XSS-Protection security header

Risk description:
The X-XSS-Protection HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

Figure 3 below illustrates and details methods that malicious actors can exploit Cross-Site Scripting vulnerabilities:

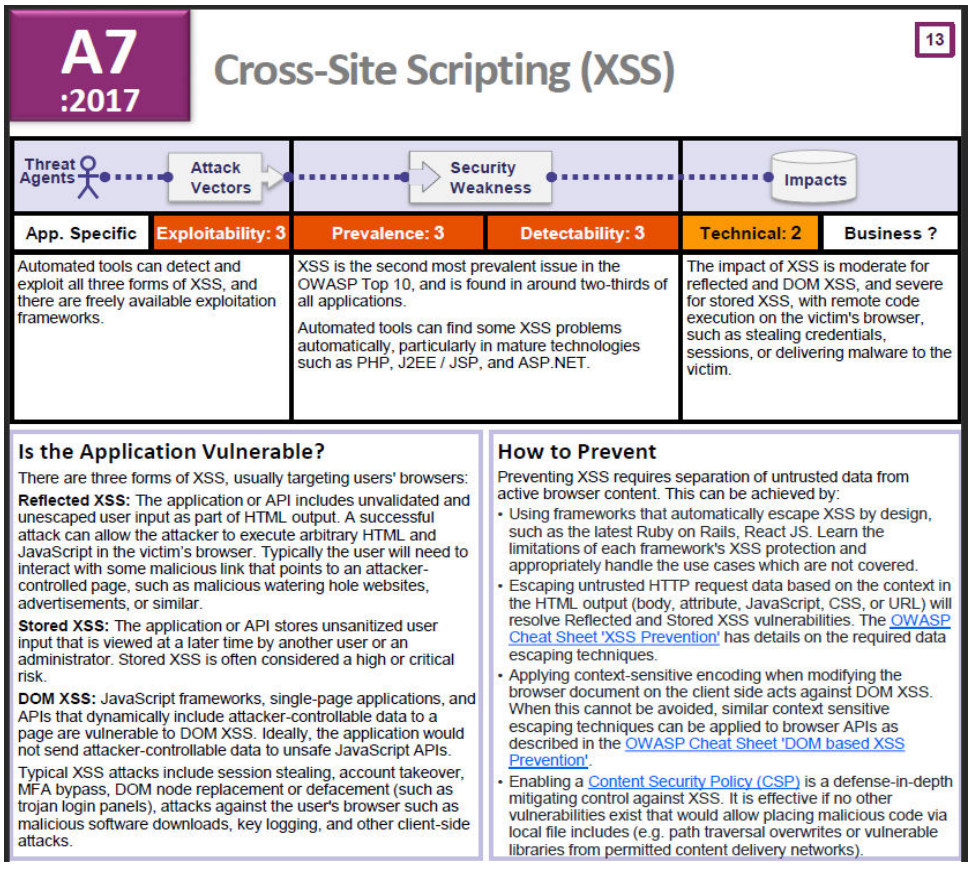


Figure 3: OWASP A7 Cross-Site Scripting (XSS) (Dehalwar et al., 2018).

A9. Using Components with Known Vulnerabilities

Components such as libraries, frameworks, and other software modules, almost run with full privileges. If a vulnerable component is exploited, such an attack can facilitate severe data loss or complete server takeover. Applications vising components with known vulnerabilities may undermine application defences and enable a range of possible attacks and impacts (ISC2, 2006).

Server software and technology found

Software / Version	Category
Apache	Web Servers
Twitter Bootstrap	Web Frameworks
jQuery 1.8.3	JavaScript Frameworks

Details

Risk description:
An attacker could use this information to mount specific attacks against the identified software type and version.

Screenshot:

Classification:
OWASP Top 10 - 2013 : A5 - Security Misconfiguration
OWASP Top 10 - 2017 : A6 - Security Misconfiguration

Figure 4 below illustrates the need for developers to establish a secure coding framework:

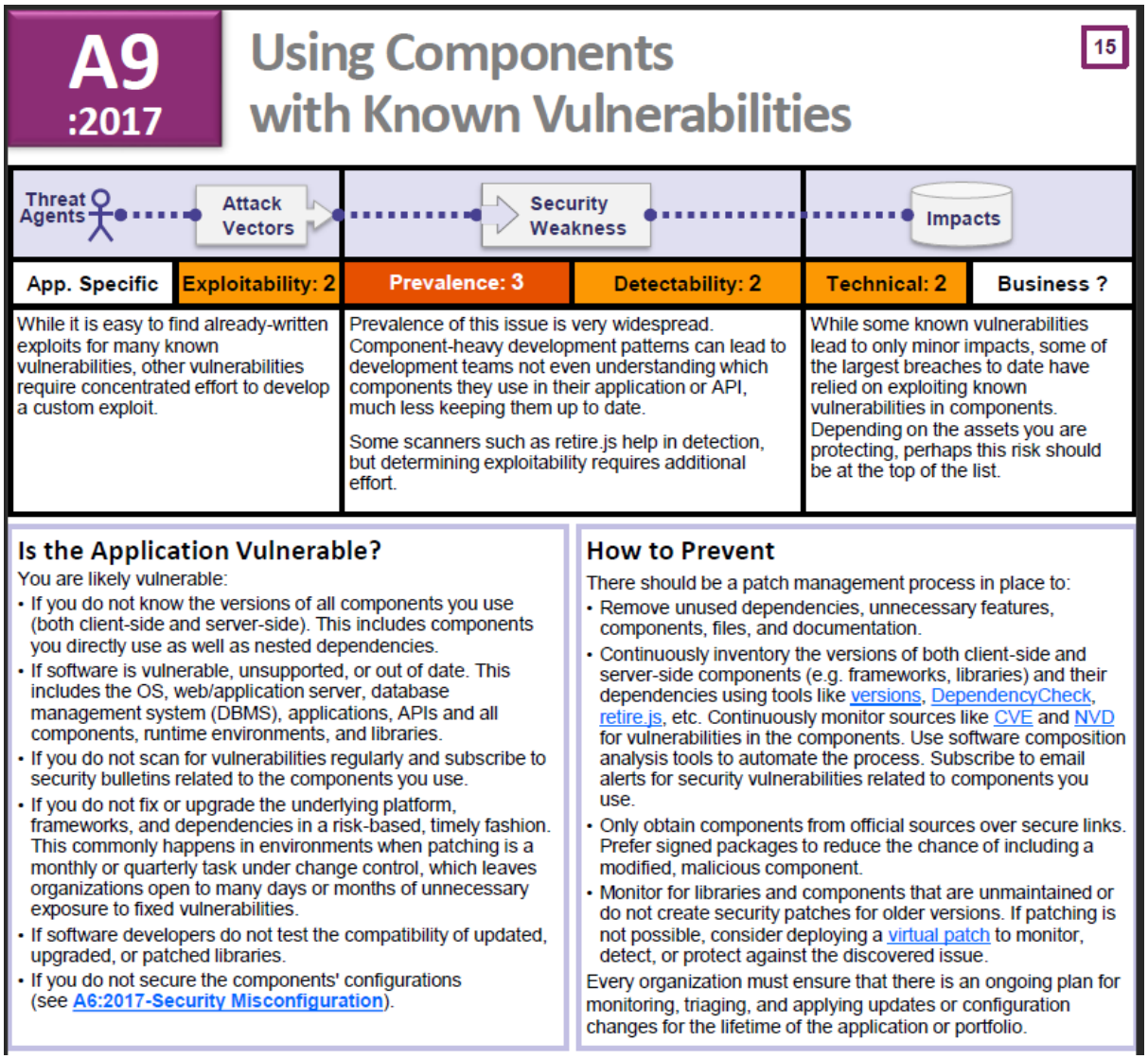


Figure 4: OWASP A9 Using Components with Known Vulnerabilities (Dehalwar et al., 2018).

Figure 5 below shows the distribution of these categories by amount of security reports, (i.e., bulletins, bug bounties, exploits):

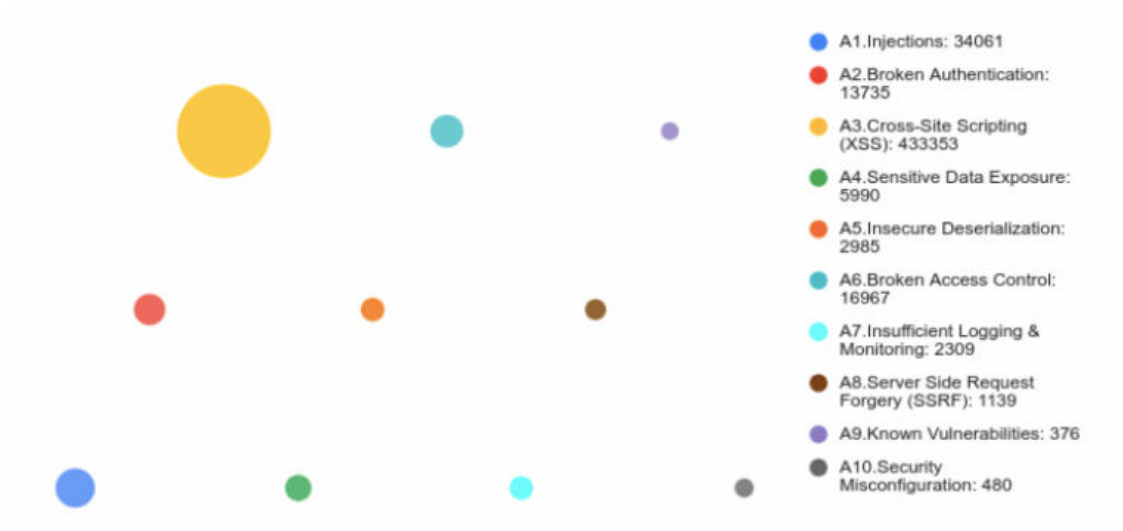


Figure 5: Distribution of OWAS categories (Wallarm, 2021).

GDPR Compliance:

The EU General Data Protection Regulation (GDPR) is a set of rules applicable to all individuals and companies that provide EU citizens services (EUR-Lex, 2016). Following a GDPR compliance test against the Group Four e-commerce website, critical compliance mistakes were discovered.

Every company and individual that provides products and services to EU citizens should have a privacy note (or notice), stating six important details (Biscoe, 2021):

- Contact Details. This includes the personal details of a person responsible for accessing and storing the data. These details include the name, address, and email address.
- The types of Personal Data stored. It is essential to highlight all the data stored in detail without using generic terms.
- How the company processes the data. A company must disclose if it transfers the data to third-party companies.

- How long the data will be kept. According to the GDPR, the data should only be stored for as long as they are needed. Data can be stored for longer only if it is for the legitimate interests of the company.
- The basis of processing the personal data. The privacy notice should include a legitimate legal reason for storing all the data. Companies and individuals should only store data required for providing their services.
- Data subject rights. The privacy notice should include all the rights an individual has regarding his/her data. This should include the right of access, right to be forgotten, right to object, etc.

Following our GDPR compliance test, we discovered that the website did not provide a GDPR privacy note. Furthermore, no Cookies consent pop-up was utilised. Additionally, during our testing, we discovered that the website was operating using a vulnerable version of its Content management system (CMS), which includes publicly known security vulnerabilities. Furthermore, the website was not configured with HTTPS encryption. According to the GDPR, an entity storing personal data should ensure that maximum security and encryption systems are deployed (EUR-Lex, 2016).

PCI DSS Compliance:

The Payment Card Industry Data Security Standard (PCI DSS) is a set of rules aiming at creating a safe environment for online payment systems. As the website we performed the test on is an e-commerce site, we had to test further using the PCI rules.

Our testing showed that the website operates using a dated Content management system (CMS) which includes publicly known security vulnerabilities.
Furthermore, the website does not have a Web application firewall (WAF) deployed or HTTPS encryption, making the website vulnerable to common web attacks.

Conclusion:

To complete our penetration test against an e-commerce website, we used a series of various tests and scans. First, we started by running reconnaissance tests to blueprint the way the server operates. This included mostly DNS, IP, and port tests. Afterwards, we utilised penetration tools such as ZAP, ImmuniWeb and the Open Web Application Security Project (OWASP) to get more information about the software and application of the server.

After we completed the tests, we reviewed the results and came across a series of security and privacy issues. These issues included Sensitive Data Exposure, Security Misconfigurations, Cross-Site Scripting (XSS) and outdated configurations with known vulnerabilities. From the four OWASP recognised risks discovered, one was High severity, two were medium severity, and one was low severity.

Furthermore, significant GDPR compliance issues were discovered, including missing consent forms, missing privacy statements and unsafe data exposure configurations that do not meet the GDPR requirements.

In addition, as the website manages online monetary transactions, it must comply with the standards set by the PCI-DSS council. However, because the website operates in a dated Content management system (CMS) and lacks proper encryption and firewall protection, it does not match the PCI-DSS standards.

This report includes our recommendations to mitigate most security issues and comply with all security and privacy standards.

We came across certain limitations during our penetration tests, including limited time to perform further tests and limited access to AWS tools, specifically the SHHGIT and SkyArt tools.

Recommendations:

The below recommendations are divided into five sections: Software security, Network-level, PCI-DSS requirements, Server operating system and GDPR requirements. The issues are sorted in an ascending style, with "1." being the most critical to be solved.

Software security recommendations:

1. Use HTTPS to encrypt the data communicating between the web browser and the server by using an SSL certificate.
2. Set X-XSS-Protection header with block mode as that can help users to prevent any XSS attack.
3. Set X-content-Type-Options to nosniff to prevent Internet Explorer browser from reinterpreting the content of a web page and thus overriding the value of the Content-Type header, as that could lead to attacks such as Cross-Site scripting or phishing.
4. Configure Referrer-Policy header to avoid any user tacking and unintentional information leakage.
5. Eliminate web server information by obscuring web server information in heads, using a hardened reverse proxy server and ensuring it has recommended software and security patches.
6. Configure Content Security Header to prevents any exploitation of Cross-Site.
7. Keep PHP and Apache software up-to-date by checking the vendor's website and apply all recommended patches. (In our target web application, PHP should be upgraded to version 8.0 as running PHP 7.3 is outdated, and Apache software should be upgraded to version 2.4.48).
8. Add X-Frame-Options HTTP header to avoid any Clickjacking attacks.

Network-level recommendations:

1. Allow HTTPS and block HTTP to make sure all user data is encrypted.

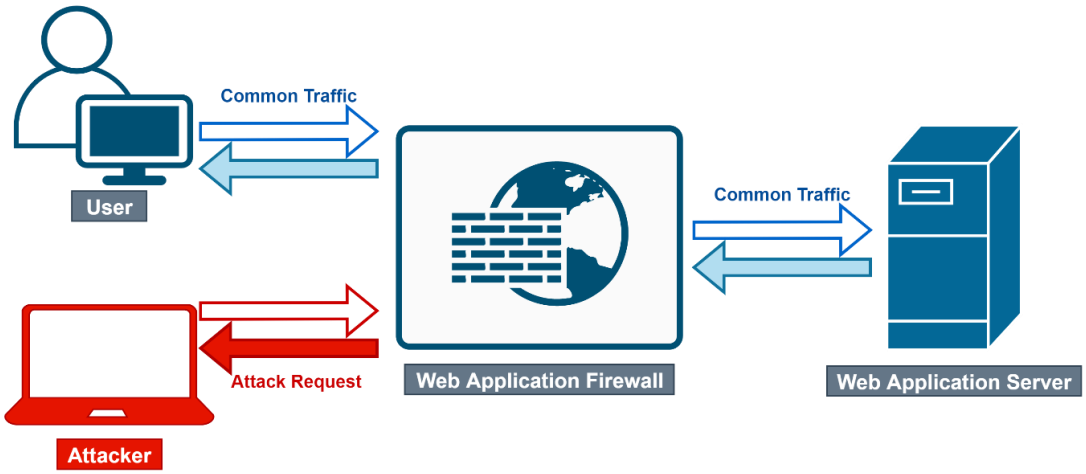
2. Block SSH port for public internet access and allow it for limited local addresses. If an SSH port is required for some customers, firewall policies should be added to allow just required customer IP addresses.
3. Allow Secure DNS to avoid cyber criminals creating any false DNS records.

Server operating system recommendations:

1. Keep the server operating system up-to-date and patched with the latest secure recommended version by working closely with the vendor and checking all critical security issues in the running version. (In the penetrated website, it is recommended to upgrade the server to the latest 46bit Amazon Linux 2 version).

PCI-DSS requirements recommendations:

1. Install applicable security patches provided by the vendor and critical security patches within one month.
2. Protect all web applications with public internet access to avoid any attack using a web application firewall to detect and block any DDoS or bad traffics.
3. Perform training for all developers to learn about secure coding and write down standardisation guidelines that all developers can follow when programming any new web application.



GDPR requirement recommendations:

1. A GDPR privacy note should be drafted, including all six important parts informing customers of how their data are processed.
2. A Cookies consent pop-up must be introduced.
3. HTTPS encryption should be used to encrypt the data transfers, including the payment details (e.g., Credit card number).
4. A Web application firewall must be deployed to shield the website from external web attacks and data leaks.
5. The website's Content management system should be kept updated to the latest version, which fixes known vulnerabilities.

References:

Biscoe, C. How to write a GDPR data privacy notice – template example Available from: https://www.itgovernance.co.uk/blog/how-to-write-a-gdpr-privacy-notice-with-documentation-template-example [Accessed 15 July 2021].

Dehalwar, V., Kalam, A., Kolhe, M., and Zayegh, A. (2018)' Review of web-based information security threats in smart grid', 2017 7th International Conference on Power Systems, ICPS 2017, pp. 849–853. DOI: 10.1109/ICPES.2017.8387407.

EUR-Lex (n.d.) Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. Available from: https://eur-lex.europa.eu/eli/reg/2016/679/oj [Accessed 15 July 2021].

ImmuniWeb (n.d) ImmuniWeb Available from: https://www.immuniweb.com [Accessed 10 June 2021].

ISC2 (2006) What is OWASP Top Ten? - Definition from WhatIs.com. Available from: https://searchsoftwarequality.techtarget.com/definition/OWASP-Top-Ten [Accessed 15 July 2021].

OWASP ZAP (n.d) OWASP Zed Attack Proxy Available from: https://www.zaproxy.org [Accessed 10 June 2021].

Rosencrance, L. (2018) What is cross-site scripting (XSS)? - Definition from WhatIs.com. Available from: https://searchsecurity.techtarget.com/definition/cross-site-scripting [Accessed 14 July 2021].

Security Standards Council (n.d.) PCI SECURITY Available from: https://www.pcisecuritystandards.org/pci_security/ [Accessed 15 July 2021].

Wallarm, I. (2021) Statistics-Based OWASP Top 10 2021 Proposal - DZone Security. Available from: https://dzone.com/articles/statistics-based-owasp-top-10-2021-proposal [Accessed 15 July 2021].

Wesley Chai (2021) What is the CIA Triad? Definition, Explanation and Examples. Available from: https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA [Accessed 14 July 2021].