Design Document: Group 1

A company known as ‘Loaded Commerce Store’ operates an e-commerce website called "https://loadedwithstuff.co.uk/". The company’s product catalogue includes digital accessories, laptops, mobile phones and more. Customers visiting the website would browse for a product and proceed to perform a purchase. Customers can also create accounts by entering their personally identifiable information (PII) such as names, e-mail addresses, contact details and address information for shipping purposes. The company operates from the United Kingdom and is therefore subjected to the following governing bodies and associated regulations:

- UK GDPR (General Data Protection Regulation of the European Union)
- DPA 2018 (Data Protection Act of 2018)
- PCI DSS (Payment Card Industry Data Security Standard)
- ICO (Information Commissioners Office)
- BEIS (Department for Business, Energy, and Industrial Strategy)
- NIS UK (Network & Information Systems Regulations 2018 - UK)

The Information Commissioners Office (ICO) is responsible for upholding information rights for the UK public as well as to provide guidance in adherence and compliance to the UK GDPR, DPA of 2018 and the NIS UK (ICO, 2021). It is important to note that the UK GDPR does not dictate or define security measures to have in place. The responsibility lies with the company based on their risks presented when processing data (ICO, 2021). Since the e-commerce website accepts credit cards for payments, it will need to comply with the PCI

DSS. Furthermore, the website is an online store which is also governed by the NIS and is classified as a Relevant Digital Service Provider (RDSP) (GOV.UK, 2021).

Being an online store accessible from the Internet, the website is vulnerable to cyber-attacks. Cyber-attacks in most cases occur when vulnerabilities are detected and exploited. According to the National Cyber Security Centre (NCSC, 2021) a vulnerability is "a weakness, or flaw, in software, a system or process. An attacker may seek to exploit a vulnerability to gain unauthorised access to a system". The following vulnerabilities in order of preference may exist on the 'Loaded Commerce Store' e-commerce website:

- Weak account passwords
- Embedded Passwords
- Default service and admin accounts on databases
- Spoofing (Phishing & Pretexting attacks)
- Exploiting open ports
- Operating System Vulnerabilities
- E-Skimming
- SQL (Structured Query Language) Injection
- Cross-Site-Scripting (XSS)
- Denial of Service Attacks

Table 1 below represents the vulnerabilities, mitigation actions and recommendations applicable to a website with a database. The mitigation and recommendation techniques aim to maintain Confidentiality, Integrity and Availability (CIA) of information.

Table 1: Vulnerabilities for an e-commerce website

Threat/ Vulnerability	Definition	Mitigation Action	Recommendations
Weak Account Passwords	Weak passwords used for customer accounts can easily be hacked (dictionary attacks)	Use strong complex passwords	1. Define a mandatory password policy (minimum 8 characters) with a combination of numbers, upper case, lower case and special characters 2. Implement MFA (Multi-Factor Authentication)
Embedded Passwords	Applications hosting websites save passwords in configuration files (Lokhande & Meshram, 2013)	Do not store passwords in configuration or registry files	Perform regular configuration file audits to ensure secure coding practices (Howard & Leblanc, 2003)
Default service and admin accounts on databases	When databases are installed, a default admin and/or service account exists with elevated privileges	Disable default database and service accounts	1. Create new accounts and implement least a privilege access model 2. Perform regular audits of database accounts
Spoofing (Phishing & Pretexting attacks)	Phishing e-mails sent to users to disclose personal information. Pretexting - hackers pretend to need information to confirm user identity (Anderson, 2020)	1. Security awareness campaign 2. Mandatory cybersecurity training 3. Using spam filters can mitigate against phishing attacks, but cannot protect against spear phishing	1. Informing customers about the company's contact channels (e.g. the company will not contact you unsolicited. Do not pass on any login data. Mails sent by us have no hyperlinks. Never use them if you supposedly receive such an email from us, it is not from us). 2. Random phishing e-mail simulations to determine which users still require cybersecurity training.
Exploiting Open Ports	Basic scan tools (NMAP, SATAN) can reveal open ports to websites, which can be exploited	1. Close/block all unused ports 2. Block ICMP	1. Implement a WAF (Web Application Firewall) to filter and block Layer 7 traffic (e.g. HTTP) 2. Implement a Next Generation Firewall to filter and block network traffic 3. Always use encryption in transit (e.g. HTTPS) for websites with TLSv1.2/TLSv1.3 and strong ciphers
Operating System Vulnerabilities	Implementing of security measures on the operating systems that hosts the website	1. Patch operating systems regularly 2. Harden operating systems according to CIS (Center for Internet Security) standards	1. Define and maintain a patching policy 2. Use endpoint security software (e.g. Symantec Endpoint Protection) 3. Install reporting and monitoring agents (e.g. FireEye)
E-Skimming	Hackers compromise payment card processing pages and steals credit card information (Rouge et al., 2020)	Update browser plugins (plug-ins that is not used should not be installed)	Use browser based tools to detect and prevent e-skimming (Bower, 2019)
SQL (Structured Query Language) Injection	Injecting SQL code into a database in an attempt to extract and steal data	1. Avoid using "select *" statements 2. Perform SQL filtering via a WAF	1. Use Stored Procedures (Wei et al., 2006) 2. Perform regular database auditing (Lokhande & Meshram, 2013) 3. Implement SQL detection technology such as machine learning algorithms (Sivasangari et al., 2021)
Cross-Site-Scripting (XSS)	Injecting malicious code into a web browser	Scan website using a web vulnerability scanner to detect XSS vulnerabilities (Lokhande & Meshram, 2013)	1. Implement an IDS (Intrusion Detection System) on the client side to detect malicious java scripts. 2. Implement a BEEP (Browser-Enforced Embedded Policy) (Nithya et al., 2015)
Denial of Service Attack	Attack on a website making it unusable and unresponsive (jeopardize availability)	Automatic scaling of server resources to cater for additional web traffic, however this is not sustainable	1. Implement load-balancers across regions or Datacenters 2. Implement a WAF (Web Application Firewall) to filter and block Layer 7 traffic 3. Implement a Next Generation Firewall with IPS (Intrusion Prevention System) and IDS (Intrusion Detection System)

List of References:

Anderson, R. (2020) *Security engineering: a guide to building dependable distributed systems.* 3rd ed. Indiana: John Wiley & Sons Inc.

Bower, T. (2019) *Identifying JavaScript skimmers on high-value websites*. Available from: https://www.imperial.ac.uk/media/imperial-college/faculty-of-engineering/computing/public/1819-ug-projects/BowerT-Industry-project-identifying-JavaScript-skimmers-on-high-value-websites.pdf [Accessed 10 December 2021].

GOV.UK (2021). Government Website of the UK - NIS Regulations: UK digital service providers operating in the EU. Available from: https://www.gov.uk/guidance/nis-regulations-uk-digital-service-providers-operating-in-the-eu [Accessed 09 December 2021].

Howard, M & Leblanc, D, E. (2003) Writing Secure Code. 2nd ed. Redmond, WA, USA: Microsoft Press.

ICO (2021). Information Commissioners Office – Guide to the General Data Protection Regulation – Security. Available from: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/security [Accessed 09 December 2021].

Lokhande, S, P & Meshram, B. (2013)E-Commerce Applications: Vulnerabilities, Attacks and Countermeasures. *International Journal of Advanced Research in Computer Engineering & Technology.* Available from: https://www.researchgate.net/publication/235697382_E-Commerce_Applications_Vulnerabilities_Attacks_and_Countermeasures [Accessed 09 December 2021].

NCSC (2021). National Cyber Security Centre – Advice & Guidance. Available from: https://www.ncsc.gov.uk/section/advice-guidance/all-topics?topics=Vulnerabilities [Accessed 09 December 2021].

Rouge, P., Yeung, C., Salsburg, D., Calandrino, J. A. (2020). Checkout Checkup: Misuse of Payment Data from Web Skimming. Available from: https://plaintextresponse.com/static/papers/ecrime2020-rouge.pdf [Accessed 10 December 2021].

Sivasangari, A., Jyotsna, J., Pravalika, K. (2021) ‘SQL Injection Attack Detection using Machine Learning Algorithm’, *5th International Conference on Trends in Electronics and Informatics (ICOEI)*. India, 3-5 June 2021. USA: IEEE. Available from: https://ieeexplore.ieee.org/document/9452914 [Accessed 10 December 2021].

Wei, K., Muthuprasanna, M., Kothari, S. (2006)‘Preventing SQL injection attacks in stored procedures’,*Australian Software Engineering Conference (ASWEC'06)*. Australia, 18-21 April 2006. USA: IEEE. Available from: https://ieeexplore.ieee.org/abstract/document/1615052 [Accessed 10 December 2021].