**Vulnerabilities within computers Group 1 team 2**

Vulnerabilities are weaknesses in a system that could leave it open to attack by threat actors such as hackers. According to Charles, Shari and Jonathan (2015, pp. 5) “For instance, a particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user’s identity before allowing data access.”

**Weak passwords**

These are passwords that are easily figured out by an attacker due to circumstances such as follows:

* The use of Personally-Identifiable Information as password

Due to having trouble remembering a random string of alphanumeric characters people tend to use an easy to remember password such as date of birth or name.

This makes a weak password since it is easy to guess WHOA.com (2019).

* Use of simple sequences – This is whereby a user uses a simple string of numbers or letters such as “123456789” and 0000 so as to remember easily. Although nowadays a lot of business software does not accept such kind of passwords.
* Use of words from the dictionary WHOA.com (2019).

Charles, Shari and Jonathan (2015, pp. 5) This makes a weak password because attackers could use a technique called dictionary attack by systematically entering every word in a dictionary as a password. This method can also be used to decrypt an encrypted message.

* Storing of passwords.

This is whereby a user stores his many passwords in a folder within the computer. Unfortunately, this is where an attacker will firstly look to get hold of critical passwords so as to gain access. Therefore, storing of passwords makes them weak AuthAnvil (2018).

There are several steps that can be taken to prevent weak passwords. Some of them are as follows Beyond Trust (2019).

* Application of password encryption

Even if passwords are stolen they still remain secure and safe.

* Creation of strong, long passphrase

Strong passwords are considered to consist of over eight characters with numbers, symbols and both upper- and lower-case letters.

* Implementation of Two-Factor Authentication methods

This is a technique mostly used by organisations whereby in addition to username and password, users are expected to confirm their identity by a one-time ode sent to them.

* Avoiding the use of dictionary words.

There are programmes used by hackers that search through thousands of dictionary words in the process of trying to crack a password. Hence dictionary words are risky.

* Avoiding storage of passwords

A stored password could be stolen by attackers. Therefore, avoid storing passwords either digitally or on paper.

* Testing of password

Testing of passwords helps to generate passwords that are less likely to be hacked. For instance, Microsoft’s Safety and Security Centre has a tool that is used for testing passwords.

**Complicated user interface**

Due to complicated user interface, the user might seek online help that could be from an attacker. In the process of gaining help, the user might provide sensitive information such as names, phone number, emails and even passwords that an attacker could use for malicious reasons.

In order to avoid this, Graphical User Interfaces should be simple and easy to use and understand. Also, organisations should provide user manuals that are written in a simple language and not complicated words.

**Missing authorisations, hidden backdoor programs and buffer overflows**

There is difference between authentication and authorisation. Authentication is verifying that “a person is (or at least appears to be) a specific user, since he/she has correctly provided their security credentials.” (Kalman, n.d.)

Authorisation is confirming that a user has access to a specific resource or has been given permission to perform a particular action.

What is missing authorisation?

Missing authorisation vulnerability is when a hacker interacts with the system and performs an action that is restricted, and the system lacks the capacity to put a stop to that action. Maybe the system may perform insufficient error checking and may give an attacker the ability to extract sensitive data.

Hidden Backdoor programs

A backdoor in computing “refers to the method by which authorized and unauthorized users are able to get around normal security measures and gain high level user access on a computer system.” (Alwarebytes, n.d.)

To prevent backdoor attacks, here are a list of measures you can take.

* Change your default passwords.
* Monitor network activity
* Choose applications and plugins carefully
* Use a good cybersecurity solution

Buffer overflows

A buffer overflow is a software coding mistake that an attacker can exploit to gain access to a system. This usually happens when there is more data in buffer than it is supposed to handle, which causes data to overflow into adjacent storage. This vulnerability can cause the system to crash or in this case create an entry point for a cyberattack.

Here are a few ways you can prevent buffer overflow attacks.

* Address space layout randomisation
* Buffer overflow protection
* Executable space protection
* Bounds checking
* Static code analysis tools

(kiuwan, 2020)

**SQL injection**

SQL injection attackers target databases through the front-end of a web from where they will take advantage of the flaws within the validation logic of the web components comments such as CGI scripts. (Keromytis, 2004). SQL is a vulnerability that occurs when the attackers have access to a SQL query through which they will leverage the syntax and the capabilities of the SQL. It has an impact on business as it leads to them, exposing personal information stored in databases such as username, password, name and more.(Clarke-Salt, 2009). SQL is the one of the most common we hacking techniques. It can be used to add, change and delete data within a database. SQL injections can affect many website or web application that used sql database.

The SQL database can protect from SQL injection in couples of ways (Acunetix, n.d.)

* Including additional validation and parameterised quire within the prepared statement.
* Development needs to make sure that the application code never to use the input directly.
* Developer should make sure that the inputs are sanitised.
* Potential malicious code elements such as single quotes shouldn’t be available.
* Use stored ways which will protect from SQL injection, but not against others.

**Missing data encryption**

Missing data encryption is when a data within a software are not encrypted before being stored. When is occurring, it led to the passing’s up right of confidentiality, integrity, and accountability that properly implemented data encryption. (CWE, 2019)

There are many ways to prevent missing data encryption such as:

* Encryption most be integrated correctly within the designed system, which included at least an encryption that will store private data of the system user’s and an encryption that will protect the system itself from unauthorised disclosure. (CWE, 2019)
* Identify the separate needs for encryption: an example will be one-way which only a user will need a key. There is also the Two-way which automatically performed on behalf of a user, a key must be available so that the plaintext can automatically recoverable by that user. (CWE, 2019)

**Path traversal**

Path traversal or directory traversal is a form of attack that occur through stored files and directories outside of the web folder been accessed.(OWASP, 2020). It is a web security vulnerability which enables attack to read files within a server that’s been running an application. These applications might be coded and data. The attack can also edit the files within the server which will enable them to change the data within the application which can lead them to take over the server. (PortSwigger, 2020)

There are ways to protect path traversal some of which are (PortSwigger, 2020):

* Avoiding passing user-supplied input to filesystem APIs.
* The application should confirm the user input before processing it.
* Vulnerabilities within a website and web application can be checked using a scanner.
* Installing latest version of web server software and make sure that all patches have been applied.

# References

Acunetix. (n.d.). *What is SQL Injection (SQLi) and How to Prevent It*. Retrieved from acunetix.com: https://www.acunetix.com/websitesecurity/sql-injection/

Alwarebytes. (n.d.). *Backdoor*. Retrieved February 13, 2020, from https://www.malwarebytes.com/backdoor/

Clarke-Salt, J. (2009). SQL Injection Attacks and Defense. In J. Clarke-Salt, *SQL Injection Attacks and Defense.* Syngress.

CWE. (2019, june 20). *CWE-311: Missing Encryption of Sensitive Data*. Retrieved from Common Weakness Enumeration (CWE): https://cwe.mitre.org/data/definitions/311.html

Kalman, G. (n.d.). *10 Most Common Web Security Vulnerabilities*. Retrieved February 13, 2020, from https://www.toptal.com/security/10-most-common-web-security-vulnerabilities

Keromytis, S. W. (2004). SQLrand: Preventing SQL Injection Attacks. In S. W. Keromytis, *International Conference on Applied Cryptography and Network Security* (pp. pp 292-302). Columbia: Springer, Berlin, Heidelberg.

kiuwan. (2020). *Prevent Buffer Overflow Attacks with Kiuwan*. Retrieved February 13, 2020, from https://www.kiuwan.com/prevent-buffer-overflow/?creative=406742328406&keyword=%2Bbuffer%20%2Boverflow%20%2Bprevention&matchtype=b&network=g&device=c&utm\_source=google&utm\_medium=cpc&utm\_campaign=uk\_irl\_en\_generic&utm\_term=%2Bbuffer%20%2Boverflow%20%2Bprev

OWASP. (2020). *Path Traversal*. Retrieved from OWASP: https://owasp.org/www-community/attacks/Path\_Traversal#

PortSwigger. (2020). *Directory traversal*. Retrieved from PortSwigger web security : https://portswigger.net/web-security/file-path-traversal

WHOA.com (2019) Top 5 Password Mistakes and Their Impact on Your Data Security. Available at: https://www.whoa.com/top-5-password-mistakes-and-their-impact-on-your-data-security/ (Accessed 30 January 2020).

AuthAnvil (2018) Top 10 Signs Your Password is Weak. Available at: https://authanvil.com/blog/top-10-signs-your-password-is-weak (Accessed 31 January 2020).

Beyond Trust (2019) Top 15 Password Management Best Practices. Available at: https://www.beyondtrust.com/blog/entry/top-15-password-management-best-practices

(Accessed 6 February 2020).

Charles,P. Shari,L. and Jonathan,M. (2015) Security in Computing. 5th. New Jersey: One Lake Street, Upper Saddle River.