**Executive Summary**

Within the web application HackThisSite, the business has requested a penetration test of the current methodology and procedures within the businesses web applications to identify vulnerabilities and test potential exploits. Both ISO 27005 and 27001 were used to provide foundational structure to the pen testing methodology.

During the investigation, activities were conducted to test the security of password structure and it was found that there were numerous procedural issues lacking industry standards. There website contain vulnerabilities related to Hypertext markup language (HTML) source code via the user end of the application.

**Level 1 breach**

Password identified with the web application source code which could indicate other sensitive information/notes could be available

Action

]Need to scrub search users facing source code of information from password to sensitive system information.

**Level 2 breach**

Password auth poor lack of password procedural requirement.

Susceptible to brute force attack

Action

Review within testing environment.

**Level 3 breach**

Able to identify a readable file (php) in the front end of the application

Directory listing and file and folder permissions

**Level 4 breach (email script)**

Able to identify notes and email to assist with password access to source code

Action

Remove hard code in HTML Java web applications use PHP as it is on the server side of the application and end users are unable to access it directly.

**Level 5 breach (emails)**

web application not to hard code sensitive information as client side code as vulnerable to attack.

**Mediation**

Web developers dev

**Level 6 (Cryptography)**

Used reversable encryption use modern methods

**Mediation**

Use of modern encryption –

Single Secret Key – encryption and decryption and the same key is shared between both the sender and receiver.

Hashing- using SHA-2 AES 256 (Advanced encryption standards)

Multi-factor Authentication- two verification methods used to access application and servers.

**Task 7 breach (command injection)**

User can access a file contain data in this case the password to access the system using the ; (semi-colon) to brute force vulnerabilities

Mediation

Sanitisation of identifiable items such as cookies

Remove direct injection of user supplied input

Use of Firewalls to stop SQL injections

To assist refer to OWASP for further advice (<https://owasp.org/www-project-top-ten/>)

**Task 8 (web form security)**

Clean treat un trusted data **<!--#exec cmd="ls ../" -->**

/var/www/hackthissite.org/html/missions/basic/8/

**Task 9 (Script)**

**<!--#exec cmd="ls ../../9" -->**

 p91e283zc3.php

**Task 10 (Java Script)**

Use session cookies encrypted still login as clear cookies after application is closed.

**Task 11 ( Directory Listing)**

Stored .htaccess apache (linux) website security