

University of Essex | Online

Executive Summary

Course: MSc Computer Science

Module: Network and Information Security Management

Assignment: Unit 11 Assignment

Date: Thursday 14th September 2021

Team: C (Kieron Holmes, Victor Hernandez, Sergio Zavarce, George Cowie)

1. Methodology:

During a comprehensive review, our testing has identified multiple security issues with the e-Health site, ranging from immediate vulnerabilities to advisory notices providing a pathway for improvement. The Black-box testing process was adopted, allowing testing to be performed from an outside attackers' perspective, with no prior knowledge of the system or infrastructure surrounding it. If a malicious user gains access to the internal network, there may be unreported issues with the codebase or infrastructure configurations that cannot easily be identified with this method of testing (Sharma, 2021).

The scoring system used throughout this document is the Common Vulnerability Scoring System (CVSS) maintained by the Forum of Incident Response and Security Teams (FIRST). This system allows security vulnerabilities to be ranked on a scale of 0 (Lowest) to 10 (Highest) based on their perceived severity against key exploitability characteristics. However, some issues identified within this report are not considered security vulnerabilities; thus, the CVSS standard could not be used for scoring. In such cases, the testing team assigned a hypothetical risk value based on the issue's potential severity if an exploit became available.

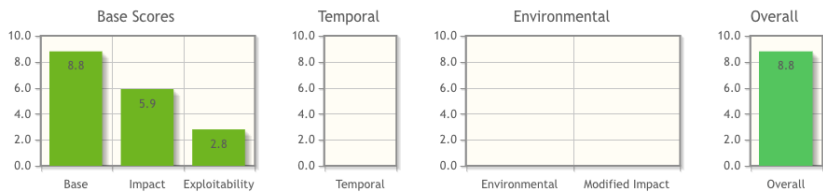
To identify vulnerabilities and security issues with the e-Health site, a series of manual and automated tests were conducted using freely available tools such as NMap, OWASP ZAP, Nikto and SecurityHeaders. During both phases, likely vulnerabilities such as Distributed Denial of Service (DDoS) attacks were avoided as

they would have breached the Terms of Service of the upstream hosting provider (Amazon Web Services).

2. Security Issues:

During the testing phase, a total of nine vulnerabilities or potentially exploitable items were identified within the e-Health site. The below vulnerabilities have been ranked on CVSS v3 scoring and the perceived threat level for potentially exploitable items. When rectifying identified issues, the items should be prioritised in risk order.

2.1. Absence of Cross-Site Request Forgery (CSRF) tokens:

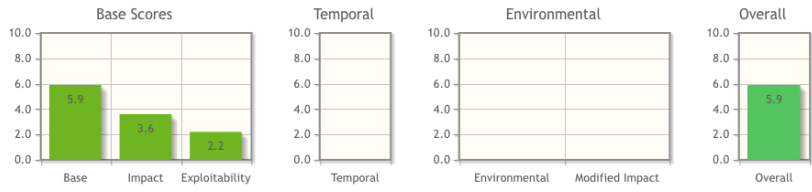
Risk	High																
CVSS Score	v3: 8.8 / v2: 6.8  <p>The chart displays the CVSS score breakdown for this vulnerability. It includes four sub-charts: 'Base Scores' showing Base (8.8), Impact (5.9), and Exploitability (2.8); 'Temporal' which is empty; 'Environmental' which is empty; and 'Overall' showing the final score of 8.8.</p> <table border="1"><thead><tr><th>Category</th><th>Score</th></tr></thead><tbody><tr><td>Base</td><td>8.8</td></tr><tr><td>Impact</td><td>5.9</td></tr><tr><td>Exploitability</td><td>2.8</td></tr><tr><td>Temporal</td><td>-</td></tr><tr><td>Environmental</td><td>-</td></tr><tr><td>Modified Impact</td><td>-</td></tr><tr><td>Overall</td><td>8.8</td></tr></tbody></table>	Category	Score	Base	8.8	Impact	5.9	Exploitability	2.8	Temporal	-	Environmental	-	Modified Impact	-	Overall	8.8
Category	Score																
Base	8.8																
Impact	5.9																
Exploitability	2.8																
Temporal	-																
Environmental	-																
Modified Impact	-																
Overall	8.8																
Description	A typical CSRF attack causes the user to do unintended actions (PortSwigger, n.d.). A malicious user can manipulate requests to impersonate a user and perform actions on their behalf, from a simple SPAM action to a privilege escalation. Generating unique tokens at the server-side prevents the attacker from forging a false request (Dizdar, 2021).																
Recommendations	<ul style="list-style-type: none">• Adding a session key or token to the URL will lower the risk, as it is unlikely that the attacker could know or guess the key/token structure.• Many Frameworks and CMS systems include CSRF protection. In addition, many open-source repositories are available to combat this issue, which can be used to prevent this issue within a purpose-built system.																

2.2. Sensitive public-facing ports:

Risk	High
-------------	-------------

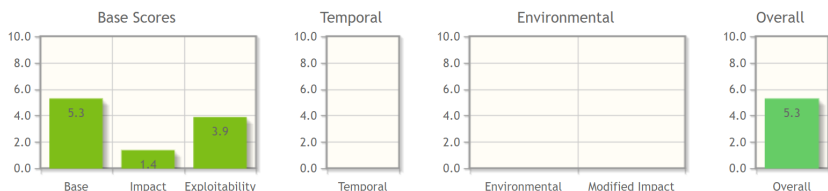
CVSS Score	N/A
Description	The server hosting the e-Health website has multiple ports open to public access, which are not essential for the operation of the public-facing e-Health site. However, if security vulnerabilities are identified within one of the services listening on those ports, a malicious user could gain unauthorised access to the system or cause DOS attacks (Tunggal, 2021).
Recommendations	<ul style="list-style-type: none"> • Ensure that a correctly configured firewall is in place, preventing outside access to all non-essential system ports. • Any services which are publicly accessible should regularly be updated to reduce the chance of malicious access in the event of new vulnerabilities being identified.

2.3. Application does not support HTTPS:

Risk	Medium																
CVSS Score	<p>v3: 6.8 / v2: 6.5</p>  <p>The figure displays four bar charts for CVSS scoring. The 'Base Scores' chart shows three bars: Base (5.9), Impact (3.6), and Exploitability (2.3). The 'Temporal' chart is empty. The 'Environmental' chart is empty. The 'Overall' chart shows a single bar for Overall (5.9). All charts have a y-axis from 0.0 to 10.0.</p> <table border="1"> <thead> <tr> <th>Category</th> <th>Score</th> </tr> </thead> <tbody> <tr> <td>Base</td> <td>5.9</td> </tr> <tr> <td>Impact</td> <td>3.6</td> </tr> <tr> <td>Exploitability</td> <td>2.3</td> </tr> <tr> <td>Temporal</td> <td>-</td> </tr> <tr> <td>Environmental</td> <td>-</td> </tr> <tr> <td>Modified Impact</td> <td>-</td> </tr> <tr> <td>Overall</td> <td>5.9</td> </tr> </tbody> </table>	Category	Score	Base	5.9	Impact	3.6	Exploitability	2.3	Temporal	-	Environmental	-	Modified Impact	-	Overall	5.9
Category	Score																
Base	5.9																
Impact	3.6																
Exploitability	2.3																
Temporal	-																
Environmental	-																
Modified Impact	-																
Overall	5.9																
Description	The application lacks an installed SSL certificate to ensure that communication between the web server and browser is encrypted (DigiCert, 2019). Due to this, any information sent between the user and the application is vulnerable to MITM attacks (Catchpoint, 2017) as the communication is not private. There are three types of SSL certificates with varying levels of trust and security; each business should consider the level of trustworthiness their users require (Lowry, 2021).																

Recommendations	<ul style="list-style-type: none"> • If the application or website is on a managed service, asking the host administrator to issue and install an SSL certificate would be enough. Most modern hosting companies have free and paid options. • If the hosting is self-managed, the Certbot from LetsEncrypt manages the installation on different OS and Web Servers with minimal or no complications.
------------------------	--

2.4. Exposure of identifiable server information:

Risk	Medium																
CVSS Score	<p>v3: 5.3 / v2: 5.0</p>  <p>The chart displays the CVSS score breakdown for this vulnerability. It includes four sub-charts: Base Scores, Temporal, Environmental, and Overall. The Base Scores chart shows a Base score of 5.3, an Impact score of 1.4, and an Exploitability score of 3.9. The Temporal, Environmental, and Overall charts all show a score of 5.3.</p> <table border="1"> <thead> <tr> <th>Category</th> <th>Score</th> </tr> </thead> <tbody> <tr> <td>Base</td> <td>5.3</td> </tr> <tr> <td>Impact</td> <td>1.4</td> </tr> <tr> <td>Exploitability</td> <td>3.9</td> </tr> <tr> <td>Temporal</td> <td>5.3</td> </tr> <tr> <td>Environmental</td> <td>5.3</td> </tr> <tr> <td>Modified Impact</td> <td>5.3</td> </tr> <tr> <td>Overall</td> <td>5.3</td> </tr> </tbody> </table>	Category	Score	Base	5.3	Impact	1.4	Exploitability	3.9	Temporal	5.3	Environmental	5.3	Modified Impact	5.3	Overall	5.3
Category	Score																
Base	5.3																
Impact	1.4																
Exploitability	3.9																
Temporal	5.3																
Environmental	5.3																
Modified Impact	5.3																
Overall	5.3																
Description	<p>The web server hosting the e-Health application exposes the software and versions used when responding to web requests. Security researchers can use this information during the server fingerprinting/reconnaissance stages, allowing them to identify potential vulnerabilities with a sites underlying software (OWASP, n.d.b).</p>																
Recommendations	<ul style="list-style-type: none"> • Update the Apache configuration file to disable the Server Signature and Server Token headers (Kiprin, 2021). This will prevent the server type and version from being leaked with HTTP responses. • Implement a reverse proxy to intercept and forward requests to a backend node. This will help disguise the servers' identity and software packages from a malicious user whilst providing enhanced functionality such as caching and load balancing. (Cloudflare, n.d.). 																

2.5. Missing content security policy:

Risk	Medium
-------------	---------------

CVSS Score	v3: 4.3 / v2: 6.1 <div> <p>The figure displays four bar charts for CVSS scoring. The 'Base Scores' chart shows Base (6.1), Impact (2.7), and Exploitability (2.8). The 'Temporal' chart is empty. The 'Environmental' chart is empty. The 'Overall' chart shows an overall score of 6.1. All charts have a y-axis from 0.0 to 10.0.</p> </div>
Description	This vulnerability is a general lack of security policy on a given website or application. For example, the CSP policy prevents XSS, Clickjacking and other kinds of scripting attacks.
Recommendations	Different tools like Google CSP evaluator can help apply a CSP policy on your site. However, awareness about the codebase, API, or other third-party services is needed to implement a non-restrictive and secure policy.

2.6. Sensitive public-facing files:

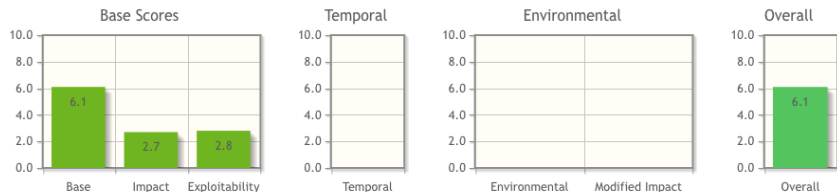
Risk	Medium
CVSS Score	N/A
Description	Identifying sensitive assets and setting the appropriate permissions to protect non-public information is a basic requirement of any Infosec strategy. Having information leaks could lead to different kinds of problems depending on the non-public information that has been compromised.
Recommendations	<ul style="list-style-type: none"> Scanning websites and applications to find any file that could be misplaced or inappropriately labelled as 'public' is a good first step. Having a good categorisation of files to identify private and public information must be applied

2.7. Lack of request throttling:

Risk	Medium
CVSS Score	N/A

Description	<p>During testing, it has been identified that the web forms located on the e-Health site are not subject to rate limiting. As a result, users with malicious intent (or infected with Malware) would be able to execute high volumes of form completions in a short period.</p> <p>If the form requires complex database/background tasks, this form of attack could lead to a Denial of Service (DoS) attack (Emergent Software, 2021), causing extended response times for genuine users. Alternatively, this attack could also be used to insert large volumes of false data into a customer's accounts to hinder their application usage.</p>
Recommendations	<ul style="list-style-type: none"> • All public-facing forms should use a challenge-response test to determine whether the user accessing the form is a human or an automated tool. If this check fails, the service should reject the form completion attempt. • Logged-in users should be subject to a session and time-based rate limit to prevent high volumes of form completions. However, this value should be set to a realistic figure which is unlikely to be hit by a genuine user.

2.8. Lack of X-Frame-Options headers:

Risk	Low																
CVSS Score	<p>v3: 3.1 / v2: 2.6</p>  <p>The figure displays four bar charts for CVSS scores. The 'Base Scores' chart shows Base (6.1), Impact (2.7), and Exploitability (2.8). The 'Temporal' chart is empty. The 'Environmental' chart is empty. The 'Overall' chart shows an Overall score of 6.1.</p> <table border="1"> <thead> <tr> <th>Category</th> <th>Score</th> </tr> </thead> <tbody> <tr> <td>Base</td> <td>6.1</td> </tr> <tr> <td>Impact</td> <td>2.7</td> </tr> <tr> <td>Exploitability</td> <td>2.8</td> </tr> <tr> <td>Temporal</td> <td>-</td> </tr> <tr> <td>Environmental</td> <td>-</td> </tr> <tr> <td>Modified Impact</td> <td>-</td> </tr> <tr> <td>Overall</td> <td>6.1</td> </tr> </tbody> </table>	Category	Score	Base	6.1	Impact	2.7	Exploitability	2.8	Temporal	-	Environmental	-	Modified Impact	-	Overall	6.1
Category	Score																
Base	6.1																
Impact	2.7																
Exploitability	2.8																
Temporal	-																
Environmental	-																
Modified Impact	-																
Overall	6.1																
Description	<p>This option prevents a browser from loading the website as an iframe. The level of protection can be configured by the website owner (Hawthorne, 2020), for example, allowing loading on a particular domain to denying any domain from loading the website.</p> <p>Clickjacking (Also known as User Interface Redressing) can</p>																

	be prevented by implementing these headers. This particular attack tricks the user into clicking an external source loaded into the host webpage without their awareness (IBM, n.d.).
Recommendations	Setting up the <code>x-frame-options</code> directive at the webserver level denies the site to be loaded on a <code><frame></code> or <code><iframe></code> tag on a third-party website if it is not allowed to.

2.9. Use of external JavaScript:

Risk	Low
CVSS Score	N/A
Description	<p>The frontend code used by the e-Health application makes use of a JavaScript file hosted by a third-party provider. Therefore, to fully leverage this vulnerability, the third-party website hosting the JavaScript file would also need to be exploited to allow modification of the identified resource.</p> <p>If exploited, the JavaScript file could be edited to modify the webpage contents to cause annoyance to the user or generate revenue for the attackers in the event of advert insertion. In a worst-case scenario, this exploit could allow an attacker to retrieve session cookies belonging to genuine users and gain complete control of the users' account (OWASP, n.d.a).</p>
Recommendations	<ul style="list-style-type: none"> • In line with the MDN Web Docs (n.d.), the developers of the e-Health application should ensure that a subresource integrity attribute is added to the <code><script></code> HTML tag. Web browsers will compare the expected and actual cryptographic hashes to ensure no prior file manipulation. • Alternatively, the e-Health website could ensure all external dependencies (Such as JQuery) are hosted on their servers and compiled in line with other core site assets.

3. Security Standard Evaluation:

The National Institute for Health and Care Excellence (NICE) recommends several steps for improving the service health providers deliver. Following these guidelines, several assessments were performed on the website to identify security gaps and improvement areas. It is crucial to make a plan with a list of steps containing deadlines, a person responsible, and KPIs to measure achievements. Patient safety, privacy, and clinical effectiveness are vital for compliance with related health standards (NICE, 2018).

The main standards that the company should comply with are GDPR and the Data Protection Act of 2018 (DPA 2018), assuming most relationships will be UK or European-based. There are subtle differences between these, but important in terms of personal health information. For example, processing personal sensitive data in health and social care use cases has more legal bases in DPA than in GDPR (Swinhoe, 2019). The DPA 2018 has increased the ICO power to impose fines for non-compliance with the regulations (Spencer & Patel, 2019).

There is a specific standard for US-based interactions, which is HIPAA. It is important to distinguish between these regulations since they enforce different rules in some respects such as breach notifications; HIPAA requires issuing breach notifications to those affected within 60 days, while GDPR demands 72 hours. Personal health information (PHI) is another crucial difference since it can be transferred without the patient's consent if it is required by treatment, while GDPR always requires consent. Additionally, GDPR gives the patient the right to have its

data deleted and forgotten in some cases, whereas HIPAA does not provide this ability.

4. Conclusion:

After being approached by a health provider to gather evidence of any possible security gaps in their website, this paper explained the outcome of following the proposed plan or design. The tools used have successfully found nine (9) security issues of different severities that compromise the financial and reputational health of the company. Furthermore, these issues can lead to service denial and system hijacking, compromising the user's private data, thus risking GDPR compliance.

The proposed methodology has been very effective by prioritising the severity of the issues, providing a more detailed line of action for the company. The scanning and testing results have been documented, and recommendations have been detailed for every security issue. These findings remark the necessity of refactoring the software by incorporating design patterns and good practices, paying particular attention to the OWASP list of vulnerabilities. Performing scheduled penetration tests periodically to ensure security and privacy compliance after applying the recommended corrections or other software updates is also recommended.

More extensive research on standards and regulations has been conducted in this report, which extends the initial scope of the design. Some of the vulnerabilities found on the website pose threats to data privacy, which are the responsibility of the data processor according to GDPR/DPA regulations, and recommendations have been added to address every vulnerability found. It is vital to follow the suggested

guidelines to prevent data breaches, as standards require organisations to assess and mitigate the risks of processing personal data (ICO, 2020).

5. References:

Catchpoint. (2017) Is HTTPS the Answer to Man in the Middle Attacks?. Available from: <https://www.catchpoint.com/blog/https-man-in-the-middle> [Accessed 24th October 2021].

Cloudflare. (n.d.) What is a reverse proxy? | Proxy servers explained. Available from: <https://www.cloudflare.com/en-gb/learning/cdn/glossary/reverse-proxy/> [Accessed 24th October 2021].

Data Protection Act 2018 c. 12. United Kingdom. Available from: <https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted> [Accessed 24th October 2021].

Digicert (2019) Beginner's Guide to TLS/SSL Certificates: Available from: <https://www.digicert.com/resources/beginners-guide-to-tls-ssl-certificates-whitepaper-en-2019.pdf> [Accessed 20th October 2021].

Dizdar, A. (2021) CSRF tokens: What is a CSRF token and how does it work?. Available from: <https://www.neuralegion.com/blog/csrf-token/> [Accessed 20th October 2021].

Emergent Software. (2021) A Crash Course to Combating Web Form Spam in 2021. Available from: <https://www.emergentsoftware.net/blog/a-crash-course-to-combating-web-form-spam-in-2021/> [Accessed 24th October 2021].

Hawthorne. (2020) What Does X-Frame-Options Do?. Available from: <https://www.technipages.com/x-frame-options> [Accessed 24th October 2021].

IBM (2019) Web Application is susceptible to Clickjacking (User Interface Redress Attack). Available from: <https://www.ibm.com/support/pages/web-application-susceptible-clickjacking-user-interface-redress-attack> [Accessed 20th October 2021].

ICO (2020) Guide to the General Data Protection Regulation (GDPR).

Kiprin, B. (2021) How to Prevent Web Server Information Leakage. Available from: <https://crashtest-security.com/server-version-fingerprinting/> [Accessed 24th October 2021].

Lowry, R. (2021) The dangers of non-secure HTTP. Available from: <https://www.deptagency.com/insight/the-dangers-of-non-secure-http/> [Accessed 24th October 2021].

MDN Web Docs. (n.d.) Subresource Integrity. Available from: https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity [Accessed 24th October 2021].

NICE (2018) How to Use Quality Standards. Available from:
<http://www.nice.org.uk/standards-and-indicators/how-to-use-quality-standards>
[Accessed 21st October 2021].

OWASP. (n.d.a) Cross Site Scripting (XSS). Available from:
<https://owasp.org/www-community/attacks/xss/> [Accessed 24th October 2021].

OWASP. (n.d.b) Fingerprint Web Server. Available from:
https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server [Accessed 24th October 2021].

Portswigger. (n.d.) Cross-site request forgery (CSRF). Available from:
<https://portswigger.net/web-security/csrf> [Accessed 24th October 2021].

Sharma, S. (2021) Black-Box Penetration Testing: Benefits, Drawbacks, Techniques, & Tools. Available from:
<https://www.getastra.com/blog/security-audit/black-box-penetration-testing/>
[Accessed 24th October 2021].

Spencer, A. and Patel, S., 2019. Applying the data protection act 2018 and general data protection regulation principles in healthcare settings. *Nursing Management*, 26(1).

Swinhoe, D. (2019) GDPR vs UK Data Protection Act 2018: What's the difference? Available from:
<https://www.csoonline.com/article/3410039/gdpr-vs-uk-data-protection-act-2018-what-s-the-difference.html> [Accessed 20th October 2021].

Tunggal, A. (2021) What is an Open Port?. Available from:
<https://www.upguard.com/blog/open-port> [Accessed 24th October 2021].