Executive Summary for

allmytype.co.uk

Group 2

Elizabeth Cook

Jurbe Jubet

Isioma Vina Okafor-Tolofari.

Osarodion Samuel Tolofari

Contents

1.0 Introduction

Website security is vital to any website, and organisations must ensure they implement appropriate levels of security when designing them. All websites that will process or store European or UK personal information must be compliant with the General Data Protection Regulation (GDPR) and ensure appropriate organisational and technical measures are put in place to ensure the security of this data (GDPR, 2018). This report uses recommended vulnerability scanning tools to assess the website for the potential weaknesses (Daud, 2015) discussed in the design document.

2.0 Scanning Tools

Vulnerability scanning can be conducted through manual testing or available tools (Daud, 2014). For the scanning performed on allmytype.co.uk, several tools were reviewed and selected based on criteria that assessed the tools' ease of use, flexibility, reputation, licensing, and ease of installation.

2.1 Kali Linux

Kali Linux can be installed as a virtual machine on a device allowing users access to multiple applications and frameworks (Bhatt, 2018). While some of the tools offered as part of the package are more complex and suited to experienced penetration testers, the variety available means there is also an offering for beginners with little knowledge of Kali or security testing (Leroux, 2020), making the tool as straightforward or as complex as the user requires. While downloading Kali Linux from their website is not complicated, the user must first have some virtualisation software to install Kali on, making the process more complex. For scanning allmytype.co.uk, Kali Linux was chosen to gain access to a range of tools.

Through Kali Linux we utilised Nikto, a command-line vulnerability scanner that scans for both security vulnerabilities and misconfigured web servers. Dmitry for port scanning which is a good option for gathering as much information as possible about a host (Aar & Sharma, 2017). We combined this with Network Mapper (Nmap) a free open-source tool (Kaur & Kaur, 2017), often used for network discovery and auditing by using IP packets to determine what hosts are available (Bhingardeve & Franklin, 2018).

2.2 Dmitry

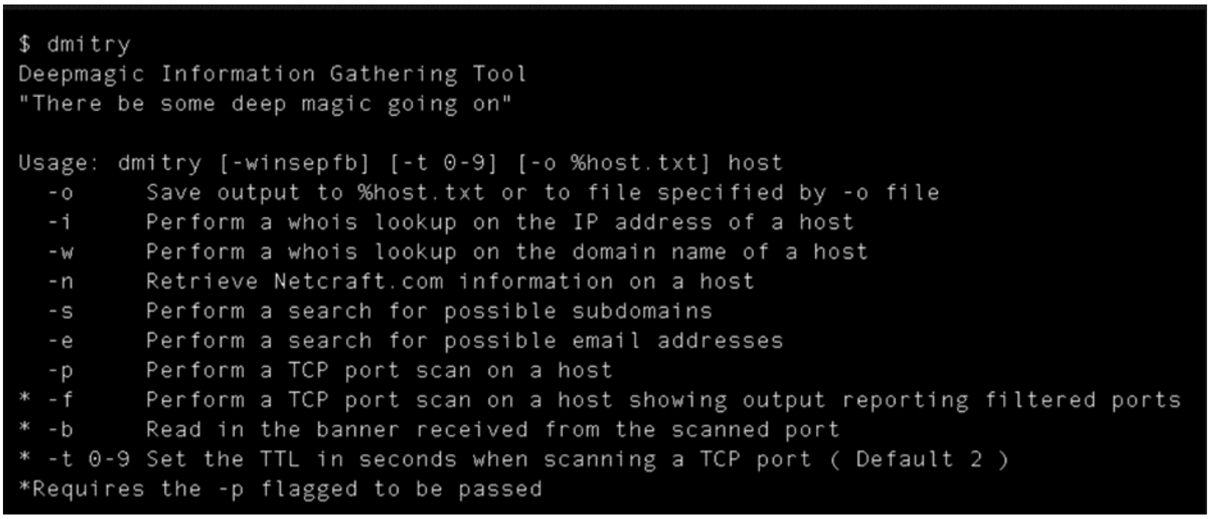
We chose Dmitry from Kali for the information gathering scan because it can (as shown in the scan)

A. Obtain information about an IP address by using the WHOIS lookup system,

B. Use the WHOIS tool to check the WHOIS information on a domain name, the Subdomain Finder to check the different records available, and so forth

C.Detect email addresses within WHOIS records or the start of authoritative DNS records and perform port scanning and banner grabbing regarding found open ports by scouring the web for 'low-hanging fruit' information.

Figure 1 - Dmitry scan



```
$ dmitry
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered ports
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
```

3.0 Vulnerabilities

Our initial design document explored a potential 12 vulnerabilities that the site may be susceptible to. Once appropriate vulnerability scanning tools were selected, the vulnerabilities were explored, and the ones confirmed to exist are discussed in more detail throughout this report. We then used the Open Web Application Security Project top 10 vulnerabilities (OWASP, 2021) to rate the vulnerability according to its most critical security risks to web applications (NGINX, 2022).

3.1 Broken Access Control

GDPR (2018) requires allmytype.co.uk to protect the integrity confidentiality and availability of personal data, access controls must be implemented to manage who has access to the data in allmytype.co.uk. Broken access control is up from fifth place in OWASP due to a number of notable Common Weakness Enumeration (CWEs). Information Disclosure to Unauthorised Personnel, GWE-201: Inserting Sensitive Data into Sent Data, and GWE-352: Access to Sent Data by Unauthorised

Personnel. Cross-Site Request Forgery (CSRF). A scan of allmytype.co.uk showed the presence of the CSRF vulnerability which allows an attacker to force a user's web browser to conduct malicious activity on the site (Siddiqui & Verma, 2011).

These vulnerabilities include:

1. Access should only be allowed for specific capabilities, roles, or users, but is open to everybody in violation of the concept of least privilege or deny by default.

2. Modifying the URL (parameter tampering or force browsing) or internal application state, or using an attack tool to manipulate API requests, to get around access control checks.

3. By giving a unique identity, you can read or edit someone else's account

3.1.2 Mitigations

1. Deny access by default.

2. Log access control failures, alert admins when appropriate for example, repeated failed login attempts.

3. Implement access control mechanisms once and re-use them throughout the application. Access controls should enforce ownership rather than accept that the user is permitted to create, read, update, or delete any record.

4. Disable directory listing for the web server and ensure that backup files and file metadata are not present within the web root.

5. Limit API and controller access to minimise the harm from automated attack tooling.

6. After logging out, stateful session identifiers should be invalidated. Tokens based on stateless JWT are recommended to be short-lived in order to minimise the window of opportunity for an attacker. The OAuth revoke access standards should be followed for longer-lived JWTs.

3.2 Cryptographic Failures

Article 32 of GDPR suggests using appropriate encryption to secure any personal information being stored or while in transit (GDPR, 2018). A Nikto scan of allmytype.co.uk revealed the Transport Layer Security (TLS) certificate is encrypted, as shown in figure 5, using AEAD Advanced Encryption Standard (AES) with 256bit key in Galois/Counter mode (AES 256 GCM) and Secure Hash Algorithm 384 (SHA384) which is more secure than other algorithms like SHA-1 (TECHCOMMUNITY.MICROSOFT.COM, 2021).

AES encryption encrypts plaintext inputs and cryptographic keys and yields an encrypted block of byte data that is decrypted by the receiver with the same key. This AES encryption supports the block cipher mode GCM which uses a cryptographic key of 128 or 256 bits in length (Google Cloud, N.D).

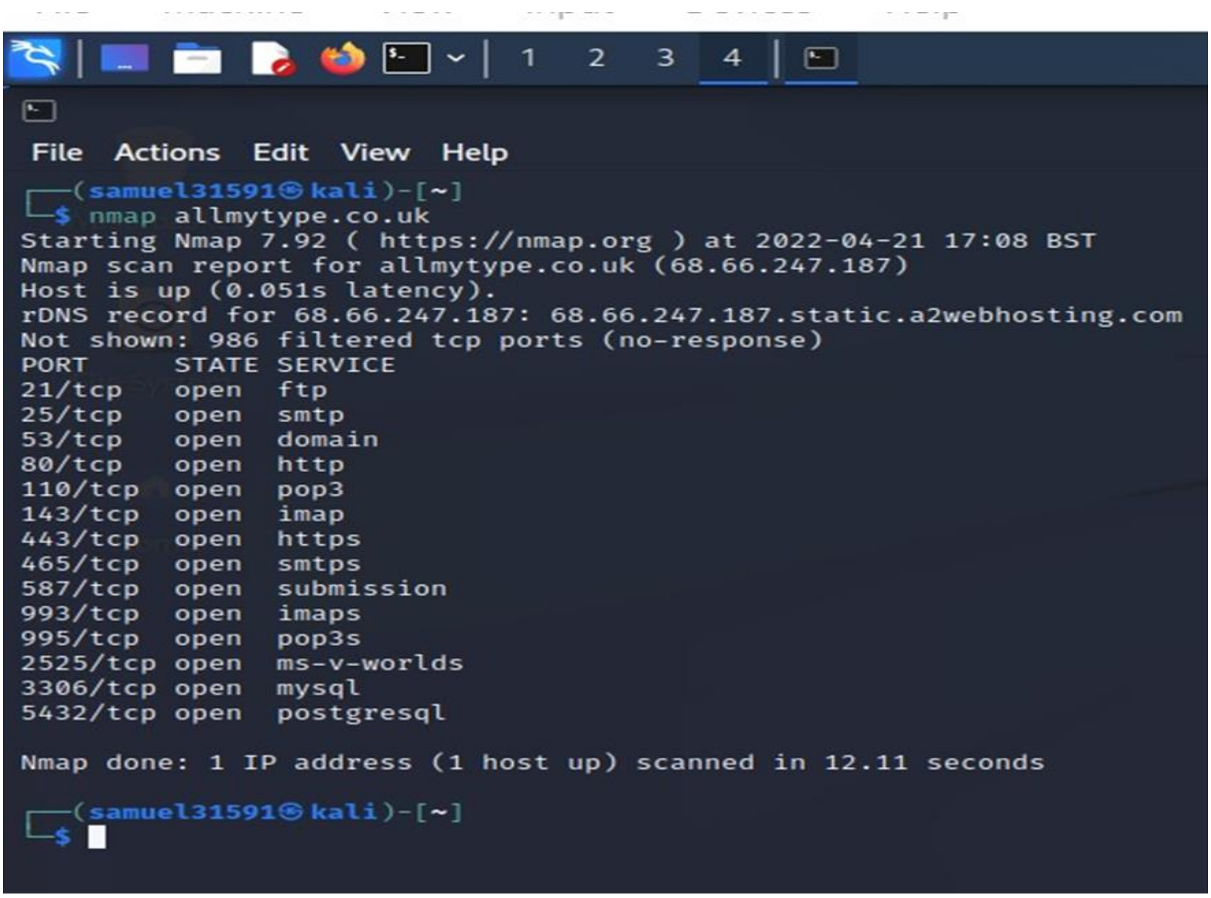
Security Hash Algorithm 2 (SHA384) produces a hash value of 384 bits as a 96-digit hexadecimal number. This helps verify if a file has been altered (WEB Tools, N.D).

Based on the above, the site is not likely to be vulnerable to SSL TLS attacks such as Advanced Persistent Malware, SSL Stripping Attacks, Man-in-the-Middle attacks, and brute force attacks and AES is known to be more resistant to the recent quantum attacks, which makes it a preferred encryption solution (Rao et al., 2017).

3.3 Security misconfiguration, improper error handling and open ports

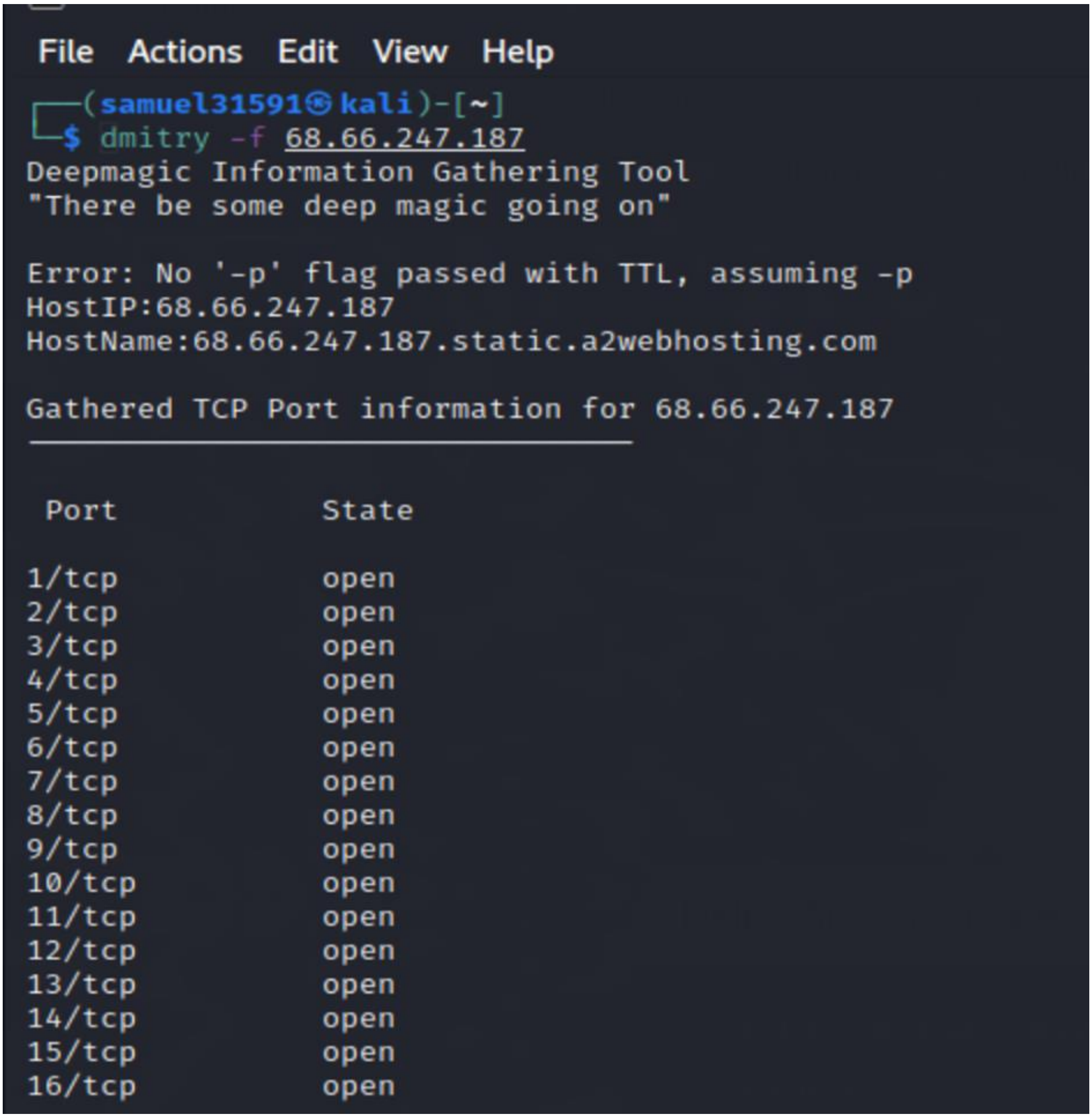
There can be security misconfigurations at any level of the application stack, including network services, platforms, web and application servers, databases, Frameworks, custom code, pre-installed virtual machines, containers, or storage. Misconfigurations, default accounts, unpatched flaws, unused access files, directories, and unprotected files, along with attackers' tactics to gain control of your systems, can be detected by automatic scanners. An attacker can exploit these flaws to gain access to certain system data or functionality, potentially compromising the whole system. As shown on the scan of allmytype.co.uk in Figure 2.

Figure 2 Open ports (Nmap scan)



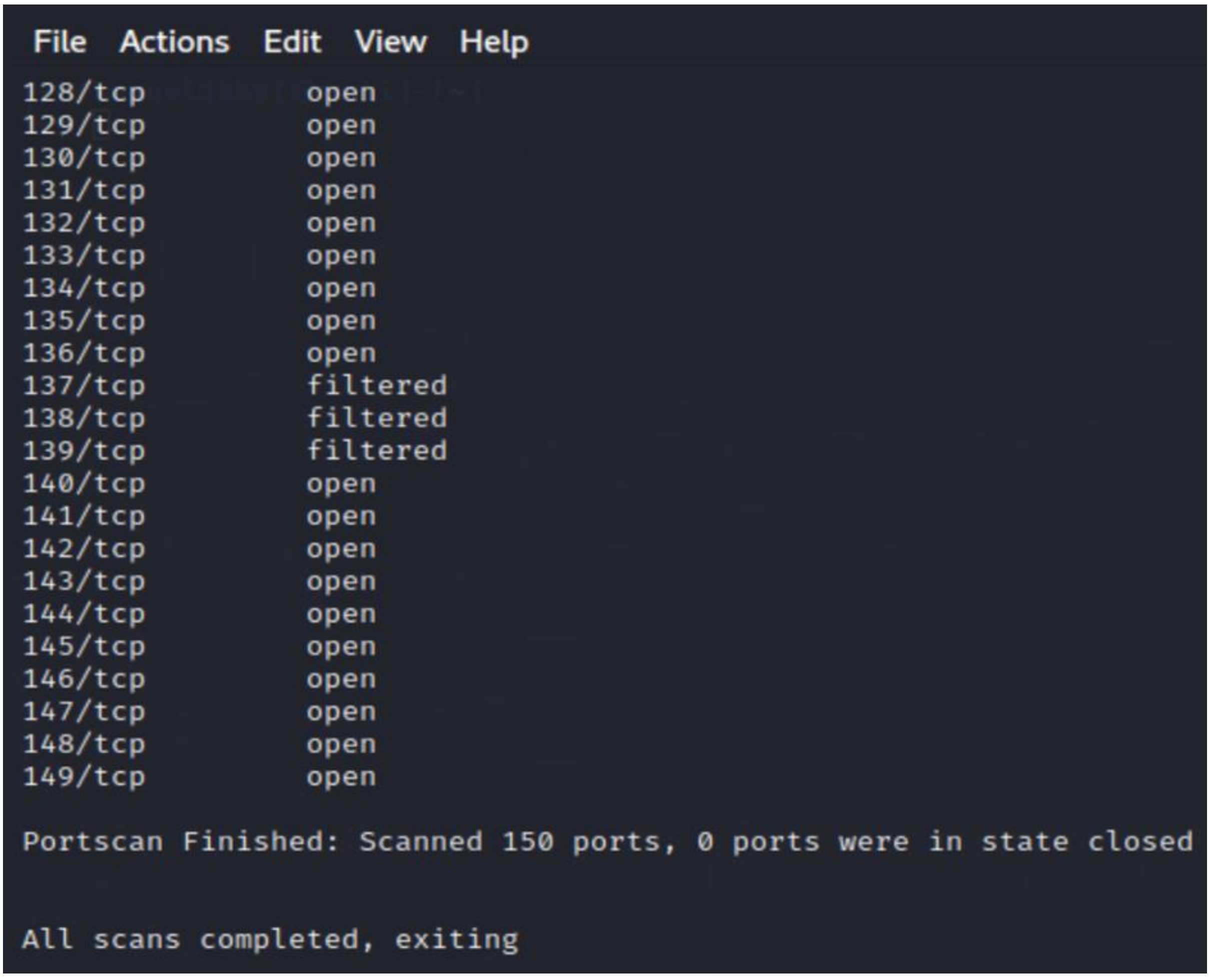
```
┌──(samuel31591㉿kali)-[~]
└─$ nmap allmytype.co.uk
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-21 17:08 BST
Nmap scan report for allmytype.co.uk (68.66.247.187)
Host is up (0.051s latency).
rDNS record for 68.66.247.187: 68.66.247.187.static.a2webhosting.com
Not shown: 986 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
2525/tcp open  ms-v-worlds
3306/tcp open  mysql
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 12.11 seconds

┌──(samuel31591㉿kali)-[~]
└─$ 
```

Figure 3 - Open ports (dmitry)



```
┌──(samuel31591㉿kali)-[~]
└─$ dmitry -f 68.66.247.187
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Error: No '-p' flag passed with TTL, assuming -p
HostIP:68.66.247.187
HostName:68.66.247.187.static.a2webhosting.com

Gathered TCP Port information for 68.66.247.187
---------------------------------


 Port           State

1/tcp           open
2/tcp           open
3/tcp           open
4/tcp           open
5/tcp           open
6/tcp           open
7/tcp           open
8/tcp           open
9/tcp           open
10/tcp          open
11/tcp          open
12/tcp          open
13/tcp          open
14/tcp          open
15/tcp          open
16/tcp          open
```

Figure 3 - Open ports (dmitry) continued



```
128/tcp         open
129/tcp         open
130/tcp         open
131/tcp         open
132/tcp         open
133/tcp         open
134/tcp         open
135/tcp         open
136/tcp         open
137/tcp         filtered
138/tcp         filtered
139/tcp         filtered
140/tcp         open
141/tcp         open
142/tcp         open
143/tcp         open
144/tcp         open
145/tcp         open
146/tcp         open
147/tcp         open
148/tcp         open
149/tcp         open

Portscan Finished: Scanned 150 ports, 0 ports were in state closed


All scans completed, exiting
```

The website might be vulnerable if;

1. There are no or incorrectly specified security configurations in any section of the application stack.

2. Features that aren't required are enabled or installed (open ports, services, pages, accounts, or privileges)

3. usage of default passwords and accounts

4. Error handling that reveals stack traces or other data which can provide useful information to an attacker

5. The most recent security settings are either disabled or improperly configured.

6. Application servers, application frameworks (e.g., Struts, Spring, ASP.NET), libraries, and databases do not have their security settings configured to safe values.

7. Software is updated or susceptible; systems are more vulnerable without a systematic application security configuration process.

3.3.1 Mitigations

Installation processes that are secure should be implemented, including:

1. Establish a repeatable hardening procedure that can be implemented fast and efficiently in a secure environment. All environments should be established up the same way, with different credentials for each domain which should be automated to save time.

2. Retain a platform simple by eliminating unnecessary features, components, documentation, and samples. To minimise the attack surface, remove or do not install new features and frameworks that are not required.

3. Implement a Patch Management process to review required updates and configuration changes.

4. implementation of segmentation, containerization, or cloud security groups, as segmented application design allows effective and secure separation between components or tenants (ACLs).

5. To ensure the effectiveness of setups and settings across all contexts, an automated procedure should be used.

3.4 Vulnerable and Outdated Components

One of the top application security threats that web developers must address, according to the OWASP 2021, is vulnerable and obsolete components (OWASP, 2021). This includes using out of support operation systems, databases, and applications as part of the website development and not ensuring identified vulnerabilities are patched.

The Nikto scan (Figure 5) showed the website is running on an Apache server and uses the Imunify360 web server security suite. Due to being open source and multi-platform (Vlad, 2022), Apache is one of the most popular choices of web server. In 2019, an update was released to patch CVE-2019-0211, which gave attackers full root access to the server (Narang, 2019). In 2022, Apache released an update to address CVE-2022-23943, a denial-of-service vulnerability (Moussalli, 2022). The release of updates must be kept up with so as not to become susceptible to a breach through a known vulnerability and ensure there is an appropriate process in place to alert the operations team for critical patches so as not to cause a delay (Luszcz, 2018).

Imunify360 offers website protection and web server security capabilities, including intrusion prevention and detection systems, network firewalls, antivirus protection, and patch management. Allmytype.co.uk uses Imunify 360, a security platform designed for web hosting servers. Researchers from Cisco's Talos unit discovered that Imunify360 could exploit a high-severity vulnerability. A remote attacker can execute arbitrary code on a server using a specially crafted file. This has been reported to the vendor and is now patched in a new version, 5.11.3 (Kovacs,2021.3).

3.4.1 Mitigations

1. Although the Apache version being used is unclear, the organisation must make sure it is patched to the latest version (2.4.46), which includes some security fixes.

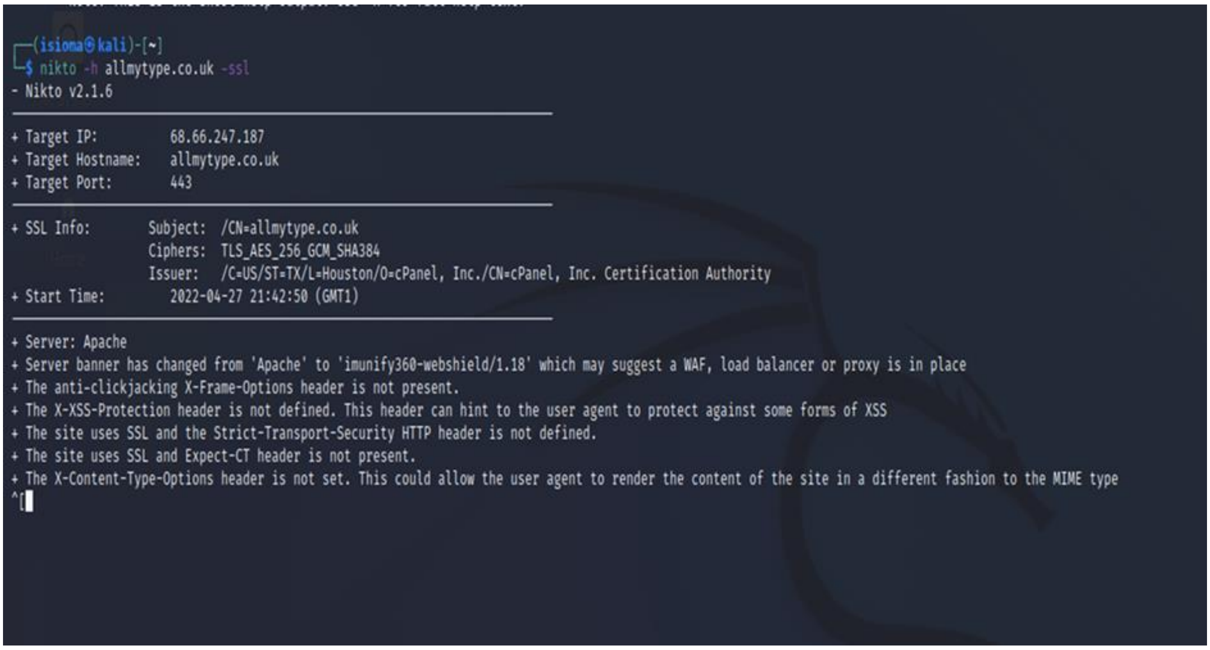
2. Update with a version of Imunify360 at least 5.11.3 or higher.

3.5 Cross-Site Scripting (XSS)

The scan in figure 5 shows the X-XSS protection header is not defined. Data is sent to a web user as dynamic content without being validated to verify that it doesn't contain malicious software. XSS attacks occur when an untrusted source like a web browser enters data into a web application without validation. Web browsers typically receive malicious code in the form of JavaScript, but it can also be HTML, Flash, or any other code the browser can execute.

Attacks using XSS are almost limitless, but they commonly involve sending private information to the attacker and redirecting it back to the victim. An XSS attacker may use the victim to access content controlled by the attacker or perform other malicious operations on the victim's machine while posing as the infected site itself.

Figure 5 - Nikto scan



```
┌──(isioma㉿kali)-[~]
└─$ nikto -h allmytype.co.uk -ssl
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          68.66.247.187
+ Target Hostname:    allmytype.co.uk
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=allmytype.co.uk
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/ST=TX/L=Houston/O=cPanel, Inc./CN=cPanel, Inc. Certification Authority
+ Start Time:         2022-04-27 21:42:50 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache
+ Server banner has changed from 'Apache' to 'imunify360-webshield/1.18' which may suggest a WAF, load balancer or proxy is in place
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
^[
```

3.5.1 Mitigations

Effective mitigation for XSS vulnerabilities will require several effective strategies including;

1. Filter input on arrival.

2. Encode data on output at the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content.

3. Use proper headers to avoid XSS in answers that aren't supposed to contain any HTML or JavaScript.

4.0 Conclusion and Recommendations

Whilst this report identifies several vulnerabilities found within allmytype.co.uk, the organisation must be aware that these may not be the only ones to exist. There were other vulnerabilities explored in the design document that have not been found due to the tools selected and the ability to test. These are, Software and Data Integrity Failures, Weak Login Credentials, Insufficient Logging & Monitoring and SQL Injection. Further investigation should be done to test for these potential vulnerabilities.

We recommend that all the mitigation activities mentioned throughout the report be addressed to further enhance the security of the site and to reduce the risk of personal information being disclosed to unauthorised individuals. Failure to address these vulnerabilities after being made aware of their presence could negatively affect the company's position if under investigation by the ICO.

Once the current risks are mitigated, we recommend the organisation stay informed of any CVEs which could affect their site and conduct regular penetration and vulnerability testing to check for new vulnerabilities that could be introduced, followed by further mitigating actions to address any findings, thus implementing a cycle of regular checks and fixes to maintain compliance with GDPR.

In conclusion, we deem the organisation to have implemented some technical measures to ensure the security of the website and this should be improved and then maintained.

5.0 References:

Aar, P. and Kumar Sharma, A. (2017). Analysis of Penetration Testing Tools. *International Journals of Advanced Research in Computer Science and Software Engineering*, 7(9). doi:2277-128X.

Bhatt, D. (2018). Modern Day Penetration Testing Distribution Open Source Platform -Kali Linux -Study Paper. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, [online] 7(2277-8616). Available at: https://www.ijstr.org/final-print/apr2018/Modern-Day-Penetration-Testing-Distribution-Open-Source-Platform-Kali-Linux-Study-Paper.pdf [Accessed 10 May 2022].

Cloud, G. (2022). *AEAD encryption concepts* | *BigQuery*. [online] Google Cloud. Available at: https://cloud.google.com/bigquery/docs/reference/standard-sql/aead-encryption-concepts.

Daud, N.I., Abu Bakar, K.A. and Md Hasan, M.S. (2014). A Case Study on Web Application Vulnerability Scanning Tools. *2014 Science and Information Conference*, pp.595–600. doi:10.1109/sai.2014.6918247.

Franklin, S. and Bhingardeve, N. (2018). A Comparison Study of Open Source Penetration Testing Tools. *International Journal of Trend in Scientific Research and Development*, [online] Volume-2(Issue-4), pp.2595–2597. Available at: https://www.ijtsrd.com/computer-science/computer-security/15662/a-comparison-study-of-open-source-penetration-testing-tools/nilesh-bhingardeve [Accessed 12 May 2022].

G, V. (2020). *Apache Server Latest Versions and Version History* | *ScalaHosting Blog*. [online] https://www.scalahosting.com/blog/. Available at:

https://www.scalahosting.com/blog/apache-server-latest-versions-and-version-history/ [Accessed 16 May 2022].

GDPR (2018). *General Data Protection Regulation (GDPR)*. [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/ [Accessed 15 May 2022].

Kaur, M.G. and Kaur, N. (2017). Penetration Testing – Reconnaissance with NMAP Tool. *International Journal of Advanced Research in Computer Science*, [online] 8(3), pp.844–846. doi:10.26483/ijarcs.v8i3.3111.

Kovacs, E. (2001). *Serious Vulnerability Found in Imunify360 Web Server Security Product* | *SecurityWeek.com*. [online] www.securityweek.com. Available at: https://www.securityweek.com/serious-vulnerability-found-imunify360-web-server-security-product [Accessed 13 May 2022].

Leroux, S. (2020). *The Kali Linux Review You Must Read Before You Start Using it*. [online] https://itsfoss.com/. Available at: https://itsfoss.com/kali-linux-review/ [Accessed 14 May 2022].

Luszcz, J. (2018). Apache Struts 2: How Technical and Development Gaps Caused the Equifax Breach. *Network Security*, 2018(1), pp.5–8. doi:10.1016/s1353-4858(18)30005-9.

Moussalli, B. (2022). *CVE-2022-23943 - Apache httpd memory corruption deeper analysis*. [online] JFrog. Available at: https://jfrog.com/blog/diving-into-cve-2022-23943-a-new-apache-memory-corruption-vulnerability/ [Accessed 16 May 2022].

Narang, S. (2019). *CVE-2019-0211: Proof of Concept for Apache Root Privilege Escalation Vulnerability Published*. [online] Tenable®. Available at: https://www.tenable.com/blog/cve-2019-0211-proof-of-concept-for-apache-root-privilege-escalation-vulnerability-published [Accessed 16 May 2022].

NGINX, F. (2022). *OWASP Top 10 2021: The New Wave of Risk*. [online] www.f5.com. Available at: https://www.f5.com/solutions/owasp-top-10-2021-ebook?utm_medium=cpc&utm_source=google&utm_campaign=PSE_EB_22Q2_noa_2021-OWASP-top-10_emea-sde_as_mav&gclid=Cj0KCQjwspKUBhCvARIsAB2IYutzGyGJ0U8ZkSCsidE2gozcqbMJw-SXTJKN7Uy_aU9FoFmwT35Vc5saAoA5EALw_wcB&gclsrc=aw.ds [Accessed 18 May 2022].

OWASP (2021a). *A01 Broken Access Control - OWASP Top 10:2021*. [online] owasp.org. Available at: https://owasp.org/Top10/A01_2021-Broken_Access_Control/ [Accessed 18 May 2022].

OWASP (2021b). *A06 Vulnerable and Outdated Components - OWASP Top 10:2021*. [online] owasp.org. Available at: https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/ [Accessed 10 May 2022].

owasp.org. (2021). *A05 Security Misconfiguration - OWASP Top 10:2021*. [online] Available at: https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ [Accessed 11 May 2022].

Rao, S., Mahto, D., YADAV, D.K. and Ali Khan, D. (2017). The AES-256 Cryptosystem Resists Quantum Attacks. *International Journal of Advanced Research in Computer Science*, Volume 8, No. 3(0976-5697), pp.404–408.

Siddiqui, Mohd.S. and Verma, D. (2011). *Cross site request forgery: A common web application weakness*. [online] IEEE Xplore. doi:10.1109/ICCSN.2011.6014783.

Techcommunity.microsoft.com, (2021). *Microsoft to use SHA-2 exclusively starting May 9, 2021*. Available at: https://techcommunity.microsoft.com/t5/windows-it-pro-blog/microsoft-to-use-sha-2-exclusively-starting-may-9-2021/ba-p/2261924 [Accessed 15 May 2022].

WEB Tools. (n.d.). *SHA-384 Hash Generator* | *Generate sha384 Online*. [online] Available at: https://wtools.io/sha384-generator-online#:~:text=SHA-384%20or%20Secure%20Hash%20Algorithm%202%20is%20one [Accessed 13 May 2022].