

**Vulnerability Audit and Assessment**

Ales Tekavcic PG. Dip. CS. 30.10.2023

Table of Content

[1. Brief summary with the list of security issues found and comparison with baseline. 3](#_Toc149497919)

[1.1 The list of security Issues found by Nessus Software 3](#_Toc149497920)

[1.2 The list of security Issues found by Kali Linux Software 4](#_Toc149497921)

[1.3 The list of issues found by Wireshark 6](#_Toc149497922)

[1.4 The list of issues found by immuniweb – online tool 7](#_Toc149497923)

[1.5 Comparisson of the issues found 7](#_Toc149497924)

[2. Methodology used, (with limitations encountered). 7](#_Toc149497925)

[3. Summary of data (graphical where possible). 8](#_Toc149497926)

[4. Comparison against a security standard, plus an evaluation of how well the business meets its GDPR requirements. 9](#_Toc149497927)

[5. Summary of conclusions (based on/linked back to data) 9](#_Toc149497928)

[6. Summary of recommendations (in descending order, based on risk). 10](#_Toc149497929)

[7. Applicable citations and references. 11](#_Toc149497930)

# Brief summary with the list of security issues found and comparison with baseline.

The initial phase of work was comprised of information gathering and overall exploration and reconnaissance of the target website. This allowed me to get the basic information about the website, for example, IP, hosting services, programming languages used, security assets (SSL, Certificates, Encryption). The main part of the work consisted of advanced network scanning using Nessus and Kali Linux in the final part at which the additional knowledge and manual approach was required.

## The list of security Issues found by Nessus Software

For initial step I ran basic Network scan, which took approximately 35 minutes. The findings were as follows:

Host Information DNS Name: pchelpme.org.uk IP: 68.66.247.187 OS: Ubuntu 16.04 Linux Kernel 4.4

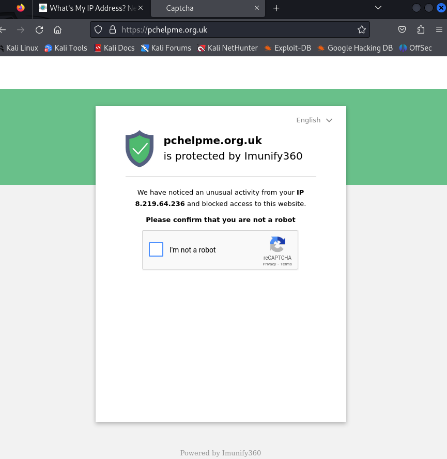


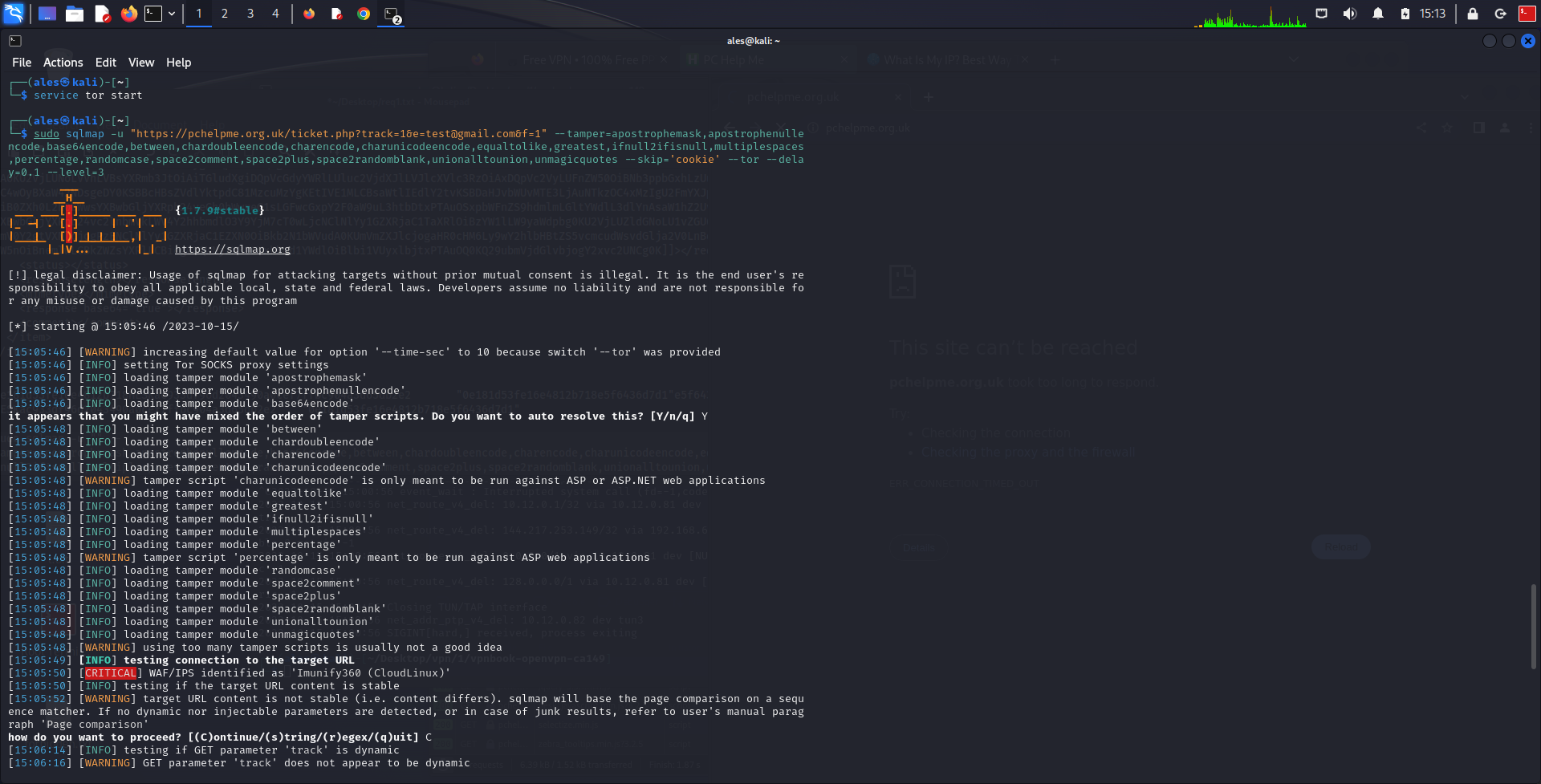
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Vulnerability | Freq. | Severity | Nesuss Plugin/  CVSS Score | Protocol | Port | Service |
| PHP Unsupported Version Detected | 2 | Critical | 58987/10.0 | 1: tcp  2: tcp | 1: 80  2: 443 | 1: http  2: https |
| SSL Medium Strength Cipher Suites Supported | 5 | High | 42873/7.5 | 1: tcp  2: tcp  3: tcp  4: tcp  5: tcp | 1: 21  2: 110  3: 143  4: 993  5: 995 | 1: ftp  2: pop3  3: imap  4: imap  5: pop3 |
| SSL Certificate with Wrong Hostname | 5 | Medium | 45411/5.3 | 1: tcp  2: tcp  3: tcp  4: tcp  5: tcp | 1: 21  2: 110  3: 143  4: 993  5: 995 | 1: ftp  2: pop3  3: imap  4: imap  5: pop3 |
| SSL Certificate Cannot Be Trusted | 5 | Medium | 51192/6.5 | 1: tcp  2: tcp  3: tcp  4: tcp  5: tcp | 1: 110  2: 143  3: 465  4: 993  5: 995 | 1: pop3  2: imap  3: /  4: imap  5: pop3 |
| SSL Self-Signed Certificate | 5 | Medium | 57582/6.5 | 1: tcp  2: tcp  3: tcp  4: tcp  5: tcp | 1: 110  2: 143  3: 465  4: 993  5: 995 | 1: pop3  2: imap  3: /  4: imap  5: pop3 |
| TLS Version 1.0 Protocol Detection | 4 | Medium | 104743/6.5 | 1: tcp  2: tcp  3: tcp  4: tcp | 1: 110  2: 143  3: 993  4: 995 | 1: pop3  2: imap  3: imap  4: pop3 |
| TLS Version 1.1 Protocol Deprecated | 4 | Medium | 157288/6.5 | 1: tcp  2: tcp  3: tcp  4: tcp | 1: 110  2: 143  3: 993  4: 995 | 1: pop3  2: imap  3: imap  4: pop3 |
| HSTS Missing From HTTPS Server | 3 | Medium | 142960/6.5 | 1: tcp  2: tcp  3: tcp | 1: 2083  2: 2087  3: 2096 | 1: www\*  2: www\*  3: www\* |
| SSL Anonymous Cipher Suites Supported | 1 | Medium | 31705/5.9 | 1: tcp | 1: 21 | 1: ftp |
| IMAP Service STARTTLS Plaintext Command Injection | 1 | Medium | 52609/4.5 | 1: tcp | 1: 143 | 1: imap |
| SSL RC4 Cipher Suites Supported (Bar Mitzvah) | 1 | Medium | 65821/5.9 | 1: tcp | 1: 21 | 1: ftp |
| SMTP Service Cleartext Login Permitted | 2 | Low | 54582/2.6 | 1: tcp  2: tcp | 1: 587  2: 2525 | 1: smtp  2: smtp |

## The list of security Issues found by Kali Linux Software

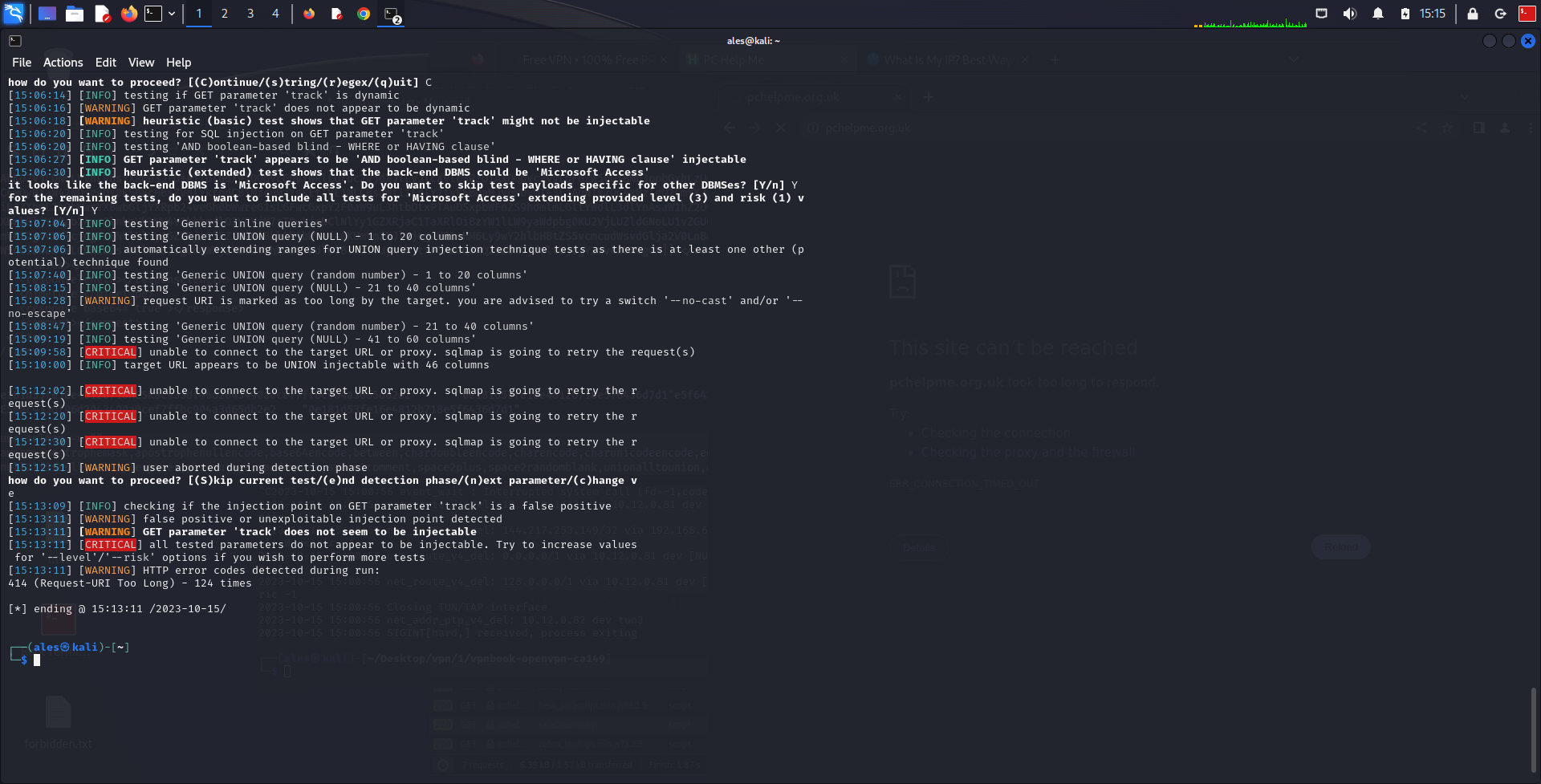
At this stage, I decided to explore vulnerabilities manually where applicable. For example, the following issues were found by using sqlmap command:

* + Bypassing the firewall with tampered Script:

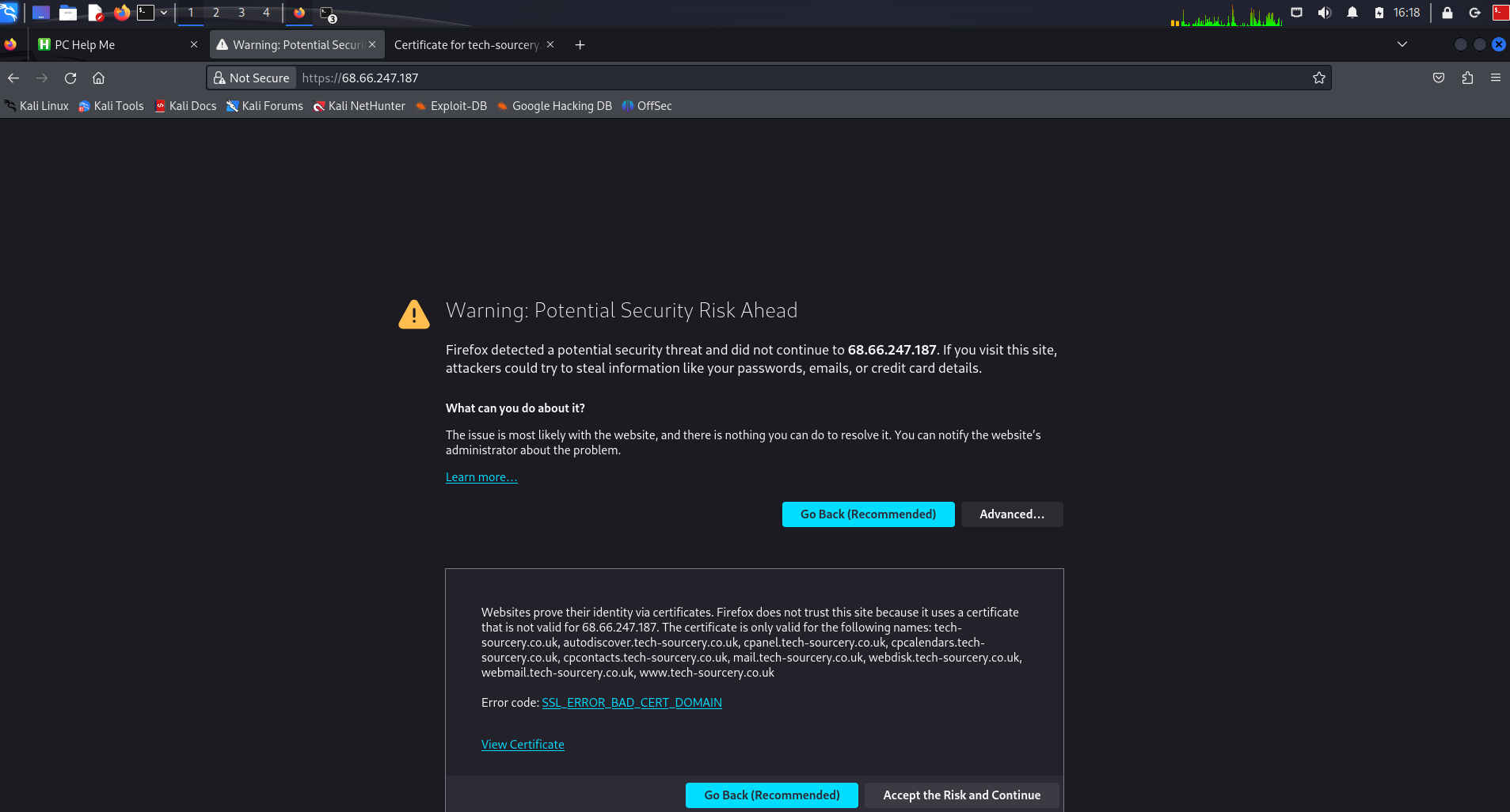




* + Found possible Injectable parameter ‘track’ And information of Database used – Microsoft Access



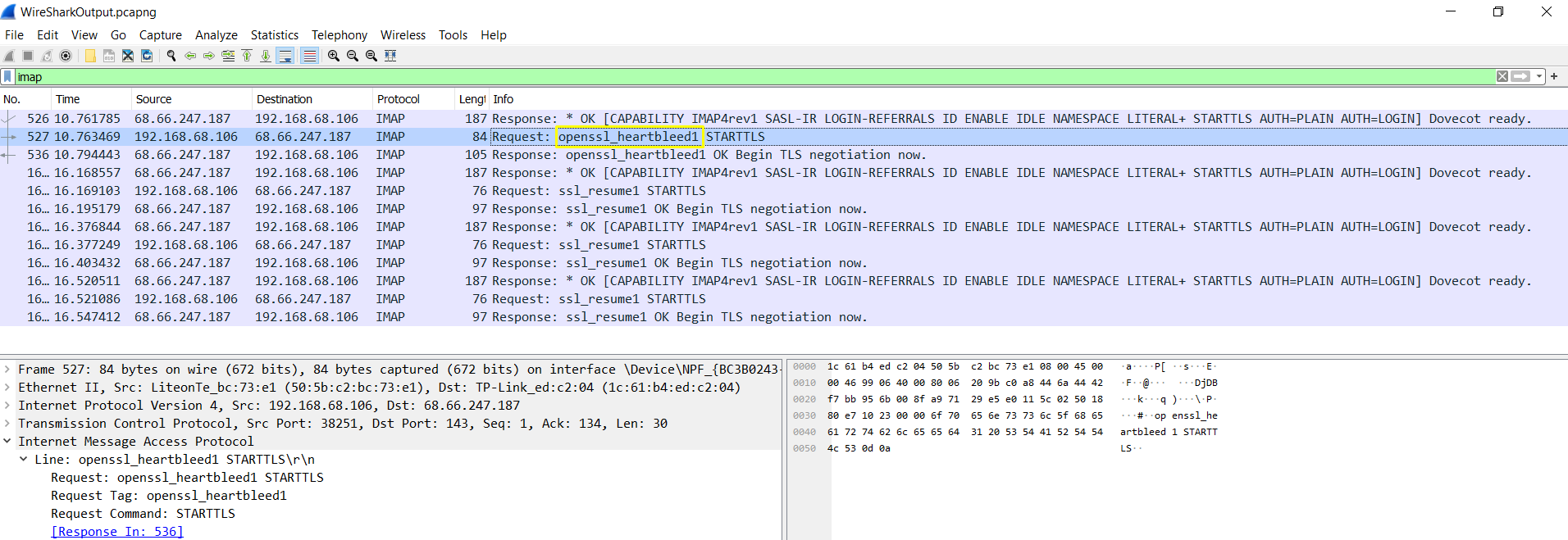
* Vulnerable certificate:



* + \*Prone to DDoS attack and phishing

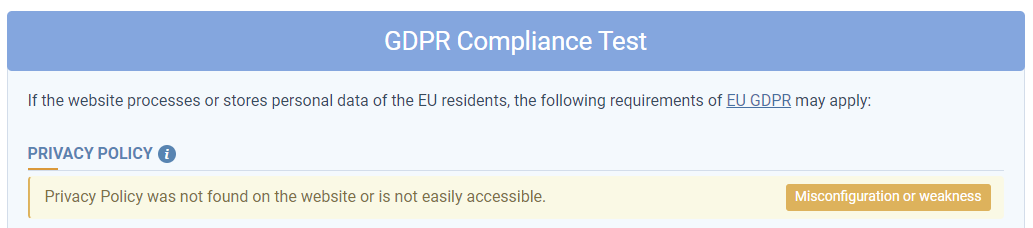
## The list of issues found by Wireshark

* + openssl\_heartbleed1



## The list of issues found by immuniweb – online tool

* + GDPR Complience not present



## Comparisson of the issues found

Based on the software that I chose, I noticed that Nessus outputted rather generic format of results for a basic Network Scan, while on the other hand, Kali Linux produced rather runtime results displayed directly to the terminal. After all, this was expected as the tool was optimized for manual testing. Moreover, Nessus identified critical issues with PHP version and multiple medium severity issues which were mostly related to the Secure Socket Layer[[1]](#footnote-1) (SSL) vulnerabilities. Comparing these with the Kali Linux tool, I can say that I was successfully able to bypass secure encryption with the so-called Tampered scripts. Therefore the fear of vulnerable certificate issues can be justified. Another important Issue that I’ve found is the injectable parameter ‘track’. This vulnerability wasn’t detected by Nessus, as expected, since it doesn’t perform injection scanning. Another interesting thing was that to some extent, nmap found different open ports compared to Nessus. Wireshark on the other hand also exposed fairly important risk issue namely heartbleed bug.

Looking overall, I would say that Nessus was easier to use, however it didn’t reveal the issues that were otherwise revealed with the professional Kali Linux tools, designed for advanced vulnerability scanning.

# Methodology used, (with limitations encountered).

My choosen approach was using the STRIDE methodology, since it offered simple and well-structured identification of the threats with regards to the attacker goals (Department for Science, Innovation & Technology, 2023). Out of 6 steps identified by the methodology, I could say that I managed to tackle first four of them. For example:

* **Spoofing** - was identified using Wireshark (heartbleed bug).
* **Tampering** - was confirmed with the use of collection of srcipts, to evade Firewall.
* **Repudiation** - was overcame with the use of proxies and VPN, since I was able to hide my identity.
* **Information Disclosure** - was identified by exposing vulnerable parameters and database used information.

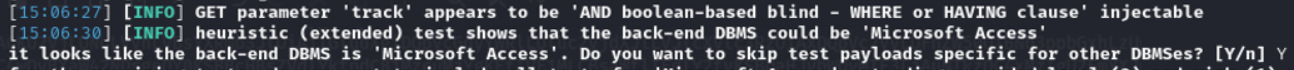
On the other hand, we could also assume that the Denial of service attack was possible, since it went for a simple web application hosted on a server. Nevertheless, I struggled to get Elevation of Privileges, due to good use of authentication and verification mechanisms. The biggest limitation that I encountered was establishing a secure connection with the server:



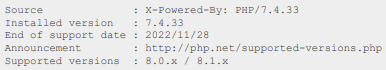
This slowed me down immensely during the penetration testing, since I had to change my public IP multiple times.

# Summary of data (graphical where possible).

Most of the potential vulnerabilities detected by Nessus software, are related to the bad use of SSL mechanisms. To some extend, I managed to exploit these vulnerabilities, since I was able to get some critical information of the website with the sqlmap command. For example, getting the injectable parameter and the name of the DBMS provider:



Looking from that perspective I would mark this flaw as a critical risk, just like the Nessus finding of the outdated PHP Version:



Looking back to the first assignment, I would reorder the list of security challenges, specific to my choosen website, since the testing identified different risks that I tought at first.

The most prominent are:

* Injection
* Vulnerable and Outdated Components
* Security misconfiguration
* Broken Access Control

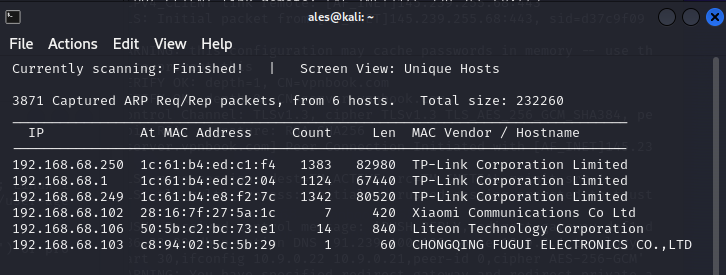
# Comparison against a security standard, plus an evaluation of how well the business meets its GDPR requirements.

During the scanning, multiple vulnerabilities were found including the ones specific to CIA Triade, which is also referenced in ISO 27001 Standard. One of these risks refer to confidentiality, since we don’t know who is able to access the data and how is stored (Irwin, 2023). Furthermore, according to Technical controls, explained in ISO27002:2022, the website doesn’t have sufficient network security, since I was able to bypass firewall. Overall, the website appears to be moderately well secured, relative to the type of the data it stores. The website is considered as simple community forum, hence there is no need for the users to enter sensitive data.

To make an inquire on the website, one has to enter a name and email, which are considered as personal data under the Article 4 in GDPR compliance. The website doesn’t provide any information about the legal controller and processor, eventually the subject can’t know how the data is processed and stored (Intersoft Consulting, N.D.). This basic information is required to be compliant under the Article 12 and 13, yet the website doesn’t list any privacy policy or at least the Conditions & Terms. From that perspective, I would conclude that the website doesn’t provide basic information about privacy, hence it doesn’t meet security standards required by GDPR.

# Summary of conclusions (based on/linked back to data)

The website can be quite vulnerable if it comes under the hand of a professional cybercriminal. However, I think, that since it doesn’t expose sensitive information, clients shouldn’t be reluctant to use it, as long as they find it beneficial. In addition, I wasn’t able to perform additional advanced testing including metasploit, since I couldn’t successfully reached the website:



From that perspective, I assumed that the website refused to respond, due to unusual traffic. Additionally, I had difficult time connecting to the website, because I struggled to establish SSL connection. To solve this issue, I had to change my public IP multiple times, as I was getting blocked by the website. This exposes another moderately well secured feature of the website.

Combining all the tools used, including Wireshark and Immuniweb, I could say that the website does have some major problems. However, these issues may still be hard to exploit by ordinary users, since they usually lack the required knowledge to perform a malicious activity.

# Summary of recommendations (in descending order, based on risk).

|  |  |
| --- | --- |
| Vulnerability | Mitigation |
| PHP Unsupported Version Detection | * + Upgrade to a version of PHP that is currently supported. |
| Bypassing firewall | * + Enable logging and monitoring features on firewalls   + Regularly updating and patching firewall software   + Employ strong access control policies to restrict network access (FasterCapital, 2023) |
| SQL Injectable parameter | * + Use of Prepared Statements (with Parameterized Queries)   + Use of Properly Constructed Stored Procedures   + Allow-list Input Validation   + Escaping All User Supplied Inputs (OWASP) |
| Heartbleed bug | * + Avoid using vulnerable versions of OpenSSL and be sure to update the software, provided by the vendor. |
| SSL Medium Strength Cipher Suites | - Reconfigure the application to use up to date cipher suite algorithms. |
| HSTS Missing From HTTPS Server | - Configure the remote web server to use HSTS. |
| SSL Certificate Cannot Be Trusted | * Purchase or generate a proper SSL certificate. |

# Applicable citations and references.

* Cloudflare. (2023). What is SSL? | SSL definition. Available from: <https://www.cloudflare.com/learning/ssl/what-is-ssl/> [Accessed 21 October 2023]
* Department for Science, Innovation & Technology. (2023). Conducting a STRIDE-based threat analysis. Available from: <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1155778/Conducting_a_STRIDE-based_threat_analysis.pdf> [Accessed 21 October 2023]
* FasterCapital. (2023). Firewall Bypassing: Examining Vulnerabilities and Countermeasures. Available from: <https://fastercapital.com/content/Firewall-Bypassing--Examining-Vulnerabilities-and-Countermeasures.html> [Accessed 25 October 2023]
* Heartbleed. (2020). The Heartbleed Bug. Available from: <https://heartbleed.com/> [Accessed 20 October 2023]
* ImmuniWeb. (N.D.). AI for Application Security. Available from: <https://www.immuniweb.com/>

[Accessed 25 October 2023]

* Intersoft Consulting. (N.D.). General Data Protection Regulation (GDPR). Available from: <https://gdpr-info.eu/art-13-gdpr/> [Accessed 28 October 2023]
* Irwin, L. (2023). Demystifying the CIA Triad: Why It’s Crucial for Cyber Security. It Governance. Available from: <https://www.itgovernance.co.uk/blog/what-is-the-cia-triad-and-why-is-it-important> [Accessed 28 October 2023]
* OWASP. (N.D.). SQL Injection Prevention Cheat Sheet. Available from: <https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html> [Accessed 26 October 2023]

1. Secure Sockets Layer is an encryption-based Internet security protocol. It’s purpose is to encrypt the data transmitted across the web (Cloudflare, 2023). [↑](#footnote-ref-1)