pentest
INFORMATION SECURITY ASSURANCE
A Shearwater Group plc Company

Report URI

Penetration Testing Report

4239

Annual App & API Test

04/12/2025

Author: Dan Dinculeana

22 Great James Street, London, WC1N 3ES
www.pentest.co.uk

Table of Contents

1 Document Revision History

Name	Date	Version	Comment
Dan Dinculeana	28/11/2025	0.1	Initial Document
Rodger Campbell	03/12/2025	0.2	QA by Senior Consultant
Dan Dinculeana	04/12/2025	1.0	Final Draft

2 Introduction

Report URI was founded to take the pain out of monitoring security headers like HSTS, CSP, and other modern browser protections. Report URI are the best real-time monitoring platform for cutting edge web standards. Their experience, focus, and exposure allow them to take the hassle out of collecting, processing, and understanding reports, giving customers just the information they need.

Report URI have indicated the need for a updated security assessment of their 'Report URI' application in order to identify vulnerabilities to attacks that could be launched across a computer network, and to provide security assurances regarding their systems. Such a test will allow Report URI to undertake remediation efforts and increase their overall security posture.

The Report URI application had been previously tested under the reference "3756" and the results of that specific test could be found in the *R3756_271124_v2.0.pdf* document.

2.1 Scope & Duration

This assessment included the following phase of work:

- Phase 1 – Web application and API assessment of the Report URI application

The duration included 5 days effort (including reporting). Work commenced on 24/11/2025 and concluded on 28/11/2025.

2.2 Scenarios Included

Testing was performed from both the perspective of an anonymous and authenticated attacker. For the purpose of the engagement, the following accounts were registered on the application:

Email Address	Role	Team
dand-test1@pentest.co.uk	Owner	Team 1 (active subscription)
dand-test1+invite2admin@pentest.co.uk	Admin	
dand-test1+invite1@pentest.co.uk	User	
dand-test2@pentest.co.uk	Owner	Team 2 (active subscription)
dand-test3@pentest.co.uk	Owner	Team 3 (no subscription)

2.3 Target(s)

- https://*.report-uri.com (limited to created subdomains only)
- https://cdn.report-uri.com
- https://docs.report-uri.com
- https://helios.report-uri.com
- https://worker.report-uri.com
- https://dash.report-uri.com

3 Executive Summary

Pentest Limited performed a penetration test for Report URI over the course of 5 days. The focus of this assessment included a security assessment for the Report URI application.

Overall, the application demonstrated strong performance, with most areas effectively mitigating common vulnerabilities such as access control issues and injection attacks. This reflects the security posture of a mature application. However, some portions of the application did not receive the same degree of security hardening found elsewhere.

The assessment revealed one (1) medium risk issue, three (3) low risk issues and one (1) informational risk issue. The following summarises the most impactful of these issues:

CSV Formula Injection (Medium): Certain parameters within the NEL and Crash requests were found to lack adequate validation or sanitisation. As a result, an attacker, whether an authenticated user within the organisation or an external party with sufficient understanding of the domain and configured filters, could submit specially crafted payloads through these parameters to target organisation users. If these payloads were later exported to CSV, they could potentially lead to the compromise of user devices, providing attackers a pathway to access additional information or system capabilities.

Vulnerabilities in Outdated Dependencies Detected (Low): Two outdated and vulnerable library versions were seen to be used by the application. These could allow an attacker to exploit cross-site scripting flaws in a situation where the configuration was changed or they would successfully bypass application restrictions.

Insufficient Session Expiration (Low): The web application set a long expiry time, of twenty-four (24) hours, for the session cookie. This could increase the chance of an attacker gaining physical access to an unprotected device, having an active session on the web application, and could allow them to gain access to sensitive information or functionality.

Key high-level process improvements that can be made are summarised as follows:

- Implement strict input validation and output sanitisation for all data included in CSV exports to prevent spreadsheet formula injection. Where formula-type content is unavoidable, ensure the application safely neutralises it before export.
- Regularly review and update third-party libraries and dependencies to ensure that outdated or vulnerable components are promptly patched or replaced, reducing exposure to known security weaknesses.
- Introduce an account lockout or timeout mechanism to limit repeated failed login attempts. This helps mitigate brute-force attacks and unauthorised access attempts.
- Enforce appropriate session expiration policies by shortening idle session timeouts and invalidating sessions upon logout or extended inactivity, reducing the risk of session hijacking.
- Harden the application's TLS configuration by disabling weak protocols and cipher suites, enforcing modern standards, and ensuring certificates are properly managed to maintain secure transport of data.

3.1 Next Steps

A complete writeup of every issue is available in the body of this report. It includes required steps to confirm and replicate each issue, along with recommended remedial actions. Pentest recommend taking time to review the findings before arranging a triage meeting to determine the order of priority for remedial work. As a rule of thumb:

- **Medium Risk Items** – Plan to address these within 3 months of discovery.
- **Low and Info Risk Items** – Track these within a risk register and discuss remediation versus acceptance.

If recommendations within this report are followed Pentest believe that the target's security posture will improve.

3.2 Caveats

Pentest provides no warranty that the target(s) are now free from other defects. Security is an ever-evolving field and consultancy is based on the opinions of the consultant, their understanding of the goals of Report URI as well as their individual experience.

The findings of this project are based on a time-limited assessment and by necessity can only focus on approved targets which are in scope. An attacker would not be constrained by either time or scope limits and could circumvent controls which are impractical to assess via structured penetration testing.

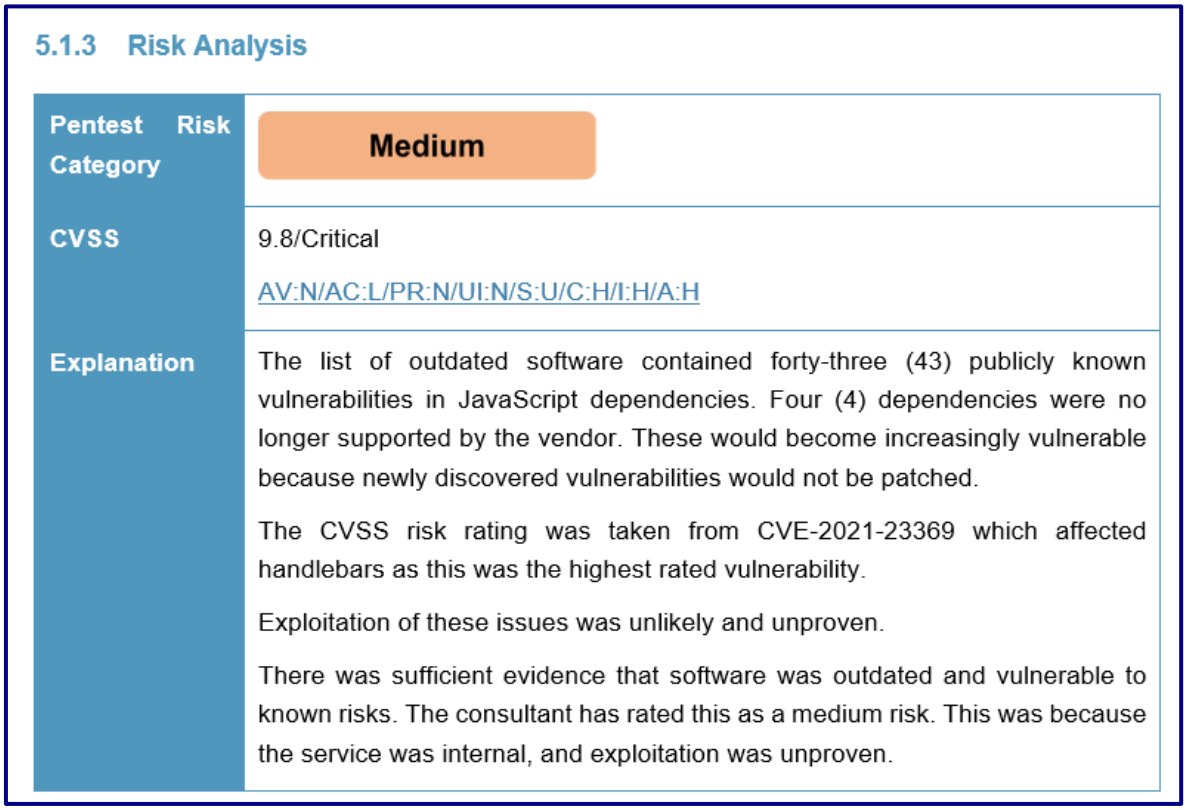
To appropriately secure assets Pentest encourage a cyclical approach to assessment. Each cycle should include:

- **Comprehensive Assessment** – where a full list of findings is produced with the widest scope possible.
- **Focused Verification Testing** – where solutions to the initial assessment's findings are verified.

Depending on how important the target is to the concerns of Report URI, Pentest recommend repeating the cycle every 6-months or 12-months at least.

3.3 Risk Categories & Rationales

The following screenshot shows an example of how we document risks in our reports:



5.1.3 Risk Analysis

Pentest Risk Category	Medium
CVSS	9.8/Critical AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Explanation	The list of outdated software contained forty-three (43) publicly known vulnerabilities in JavaScript dependencies. Four (4) dependencies were no longer supported by the vendor. These would become increasingly vulnerable because newly discovered vulnerabilities would not be patched. The CVSS risk rating was taken from CVE-2021-23369 which affected handlebars as this was the highest rated vulnerability. Exploitation of these issues was unlikely and unproven. There was sufficient evidence that software was outdated and vulnerable to known risks. The consultant has rated this as a medium risk. This was because the service was internal, and exploitation was unproven.

Figure 1 - Example Risk Rating

This risk rating was from a vulnerability describing multiple outdated JavaScript Dependencies affecting an application that was restricted to an Internal network. The risk analysis section contains three parts:

- **Pentest Risk Category**: this is a risk categorisation generated by the consultant. It is based on their experience and knowledge of the context of the vulnerability.
- **Common Vulnerability Scoring System (CVSS)**: an industry standard system which generates a numeric score between 0.0 and 10.0.
- **Explanation**: this is a statement from the consultant which explains the reasoning used to generate the “Pentest Risk Category” which may disagree with the CVSS score.

There are two scoring systems. Some customers prefer to use CVSS exclusively, while others rely on the expert opinion of the consultant.

3.3.1 Pentest Risk Category

Pentest use a risk categorisation for each vulnerability referred to as “Pentest Risk Category” throughout our reports. This is based on the experience of the consultant and their knowledge of the environment it presents itself in and is used to prioritise remediation efforts as documented in the table below:

Pentest Risk Category	Rationale
Critical	Poses a severe risk which may be easy to exploit. Begin the process of remediating immediately after the issue has been presented.
High	Poses a significant risk and can be exploited. Address these as soon as possible after any critical risks have been remediated.
Medium	Poses an important risk but may be difficult to exploit. Pentest recommends remedial work within 3 months of discovery.
Low	Poses a minor risk or may be exceedingly difficult to exploit. Address these over the long-term during testing cycles.
Info	Loss of sensitive information, or a discussion point. These are not directly exploitable but may aid an attacker. Remediate these to create a true defence-in-depth security posture.
Fixed	No further remediation is required because the issue has been verified as fixed. This can occur due to early reporting of issues during a project, or as part of a follow up verification test.

Wherever the “Pentest Risk Category” is used you will see the category icons in the table above.

3.3.2 Common Vulnerability Scoring System (CVSS)

The Common Vulnerability Scoring System (CVSS) is an industry standard formula. It generates a risk score between 0.0 and 10.0, and has its own qualitative rating system as shown below:

CVSS Score	Qualitative Rating
0.0	None
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 – 10.0	Critical

CVSS is an excellent tool used to standardise risk scores. However, it is not applicable to all risks. For example, it is incapable of capturing the risk of a “flat network design”. Experience has told us that this is a “high” risk in most cases.

For this reason, the reader may find vulnerabilities which have no CVSS rating in our reports. In these cases, the CVSS rating will be set as “N/A” (Not applicable). We endeavour to provide the reason for omitting in the “Explanation” when that is the case, and to provide CVSS by default in all applicable cases.

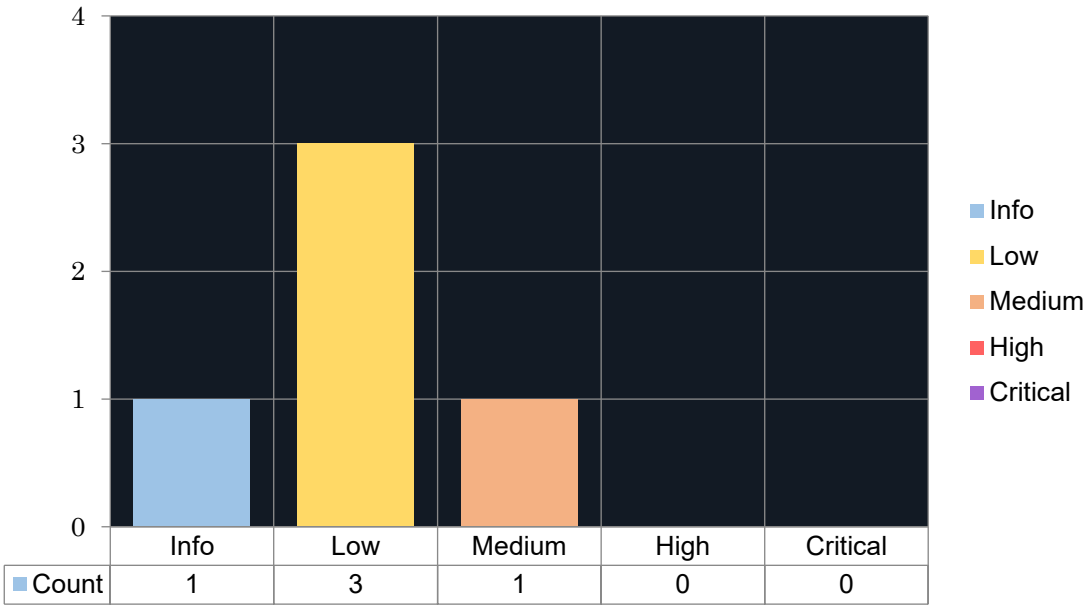
3.3.3 Rule-of-thumb Equivalency

The Pentest Risk Category contains “Fixed”, and “Info” categories which do not exist in CVSS. Therefore, there is no direct mapping from a CVSS score back to a specific Pentest Risk Category.

The table below explains the risk categories and demonstrates rule-of-thumb equivalency:

Pentest Risk Category	CVSS Score	Rationales
Critical	8.1 – 10.0	Poses a severe risk which is easy to exploit. Begin the process of remediating immediately after the issue has been presented.
High	6.1 – 8.0	Poses a significant risk and can be exploited. Address these as soon as possible after any critical risks have been remediated.
Medium	4.1 – 6.0	Poses an important risk but may be difficult to exploit. Pentest recommends remedial work within 3 months of discovery.
Low	2.1 – 4.0	Poses a minor risk or may be exceedingly difficult to exploit. Address these over the long-term during testing cycles.
Info	0.0 – 2.0	Loss of sensitive information, or a discussion point. These are not directly exploitable but may aid an attacker. Remediate these to create a true defence-in-depth security posture.

3.4 Visual Summary



4 Recommended Actions

ID	Vulnerability	Recommendation	Pentest Risk Category	CVSS
1	CSV Formula Injection	Ensure parameter values are validated or sanitised before being used to construct reports.	Medium	7.9/High
2	Vulnerabilities in Outdated Dependencies Detected	Update the affected libraries to the latest stable versions.	Low	6.4/Medium
3	Insufficient Session Expiration	Set an appropriate expiry time for the session cookie.	Low	4.1/Medium
4	Insecure TLS Configuration	Review the TLS configuration and ensure only strong algorithms and ciphers are allowed.	Low	3.7/Low
5	No Account Lockout or Timeout Mechanism	Implement an account lockout mechanism as a defence-in-depth measure.	Info	4.3/Medium

5 Technical Findings

5.1 CSV Formula Injection

5.1.1 Background

CSV Injection, also known as Formula Injection, occurs when a CSV file uses untrusted input. Given web applications use CSV generation to export data, a CSV file containing unsanitised data can be used as a method for system exploitation.

The vulnerability occurs due to the way in which spreadsheet clients interpret fields containing Formula. When a spreadsheet program opens a CSV, the application interprets any cells starting with an equals sign (=) as a Formula. This allows execution of commands in the context of the spreadsheet application user.

Since users tend to ignore security warnings in spreadsheets and when a victim downloads the malicious CSV file, the injected code will execute when the CSV file opens on the victim's computer. Thereby allowing the attacker to gain a foothold on the victim's network.

5.1.2 Details

The web application did not implement a sufficient input validation or sanitisation on the parameters of two requests (NEL, Crash). This could allow either an authenticated attacker or an anonymous attacker with knowledge of an organisation’s subdomain and filter settings to send a specially crafted payload part of the request, and target users in the specific organisation with access to the section’s report generation functionality (Export to CSV).

As an example, the consultant identified that the NEL JSON body, more specifically the “type” parameter was not strictly validated, and could allow arbitrary values to be submitted:


```
[
    {
        "age": 0,
        "type": "network-error",
        "url": "https://www.example.com/",
        "body": {
            "sampling_fraction": 0.5,
            "referrer": "http://example.com/",
            "server_ip": "123.122.121.120",
            "protocol": "h2",
            "method": "GET",
            "status_code": 200,
            "elapsed_time": 823,
            "type": "TEST"
        }
    }
]
```

It was seen that this arbitrary value would reflect in the NEL→ Reports section of the application, if the request had been accepted. To demonstrate this, the consultant injected a CSV Injection Formula payload in the parameter, as seen in the following request:

```
POST /a/t/g HTTP/2
Host: aaf0c8bee64cd02af72e7e56ecb7f633.report-uri.com
Content-Length: 342
Content-Type: application/reports+json
Origin: https://report-uri.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/142.0.0.0 Safari/537.36
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=4, i

[
    {
        "age": 0,
        "type": "network-error",
        "url": "https://www.example.com/",
        "body": {
            "sampling_fraction": 0.5,
            "referrer": "http://example.com/",
            "server_ip": "123.122.121.120",
            "protocol": "h2",
            "method": "GET",
            "status_code": 200,
            "elapsed_time": 823,
            "type": "=cmd|'/C calc'!'A1'"
        }
    }
]
```

Request Response

```
HTTP/2 201 Created
Date: Wed, 26 Nov 2025 11:17:35 GMT
[...]
```

As evidenced above, the request did not require any authentication tokens, but in order to be accepted, the correct subdomain would need to be sent. Additionally, the domain submitted in the NEL JSON body would need to match a filter that had been previously created in the organisation that had the specific subdomain submitted above.

Once accepted, the payload would be displayed in the NEL → Reports section for a period of time:

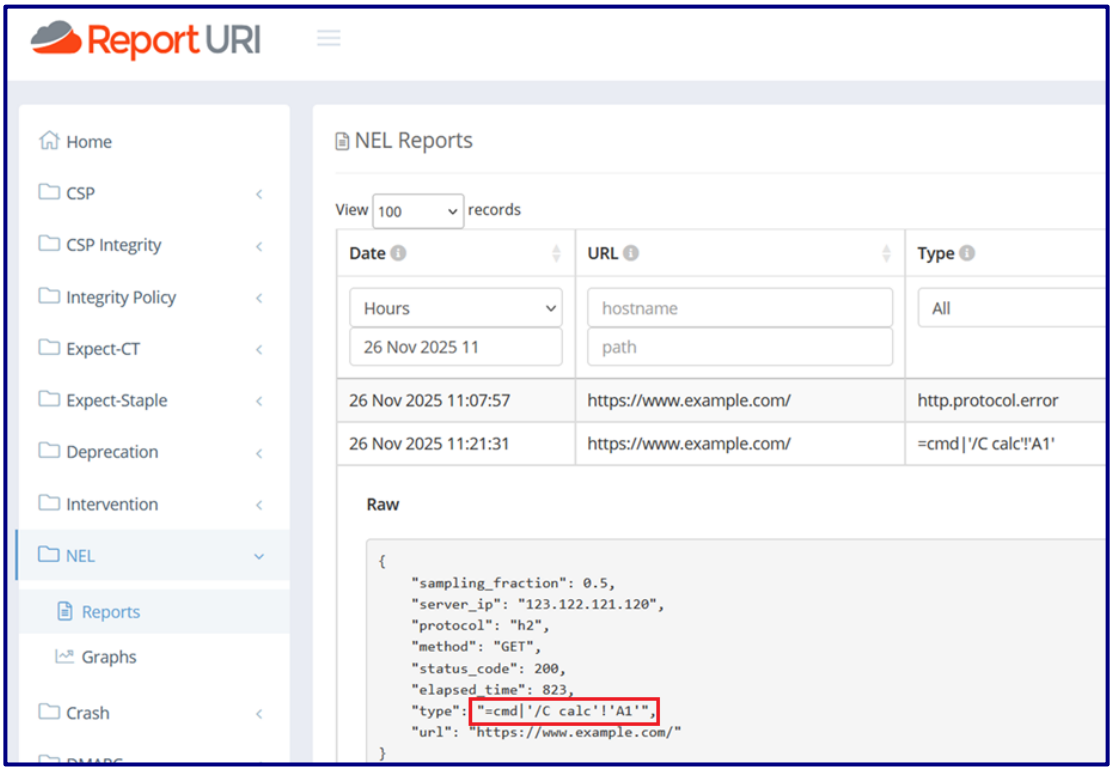


Figure 2 - NEL Request Accepted

A user reviewing the entry could then export this using the Export (CSV) button on the right side of the Reports section:

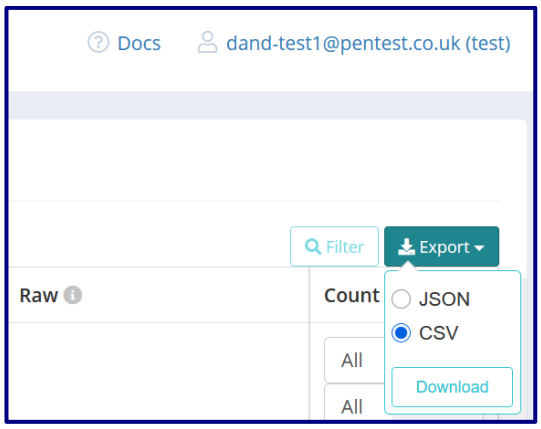


Figure 3 - Export To CSV/JSON

Clicking on the button triggered the following request:

```
POST /ajax/time/nel/ HTTP/2
Host: report-uri.com
[...]
csrf_token=6ad6bb20f3381bab8b2b1cec46372be5&length=-
1&order%5B0%5D%5Bcolumn%5D=5&order%5B0%5D%5Bdir%5D=desc&timezoneOffset=0&report_table_1_len
gth=100&export-
format=csv&unit=hours&calendar=26+Nov+2025+11&hostname=&path=&type=all&phase=all&browser=al
l&platform=all&report_table_1_length=100
```

Request *Response*

```
HTTP/2 200 OK
Date: Wed, 26 Nov 2025 11:28:54 GMT
[...]
Content-Disposition: attachment;filename="report_uri_export_NEL_26 Nov 2025 11.csv"
[...]
Date,URL,Type,Phase,Raw,Count,Browser,Platform
"26 Nov 2025
11:07:57",https://www.example.com/,http.protocol.error,unknown,"{""sampling_fraction"":0.5,
""server_ip"":""123.122.121.120"",""protocol"":""h2"",""method"":""GET"",""status_code"":20
0,""elapsed_time"":823,""type"":""http.protocol.error"",""url"":""https://www.example.com/"
"}",1,unknown,windows
"26 Nov 2025 11:21:31",https://www.example.com/,"=cmd|'/C
calc'!'A1'",unknown,"{""sampling_fraction"":0.5,""server_ip"":""123.122.121.120"",""protoco
l"":""h2"",""method"":""GET"",""status_code"":200,""elapsed_time"":823,""type"":""=cmd|'/C
calc'!'A1'"",""url"":""https://www.example.com/""}"[...]
```

Once the user opened the document, the following security notice would be displayed:

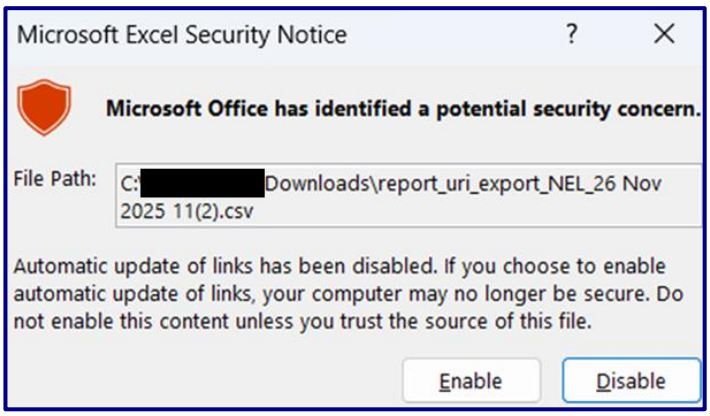


Figure 4 - Excel - Security Notice

Accepting this, would display a second warning, informing the user that Excel needs to open another application “CMD.EXE”:

Figure 5 - Excel Warning - Start Another Application

If the user accepts the warning, the payload executes, and in this case, opened Windows calculator:

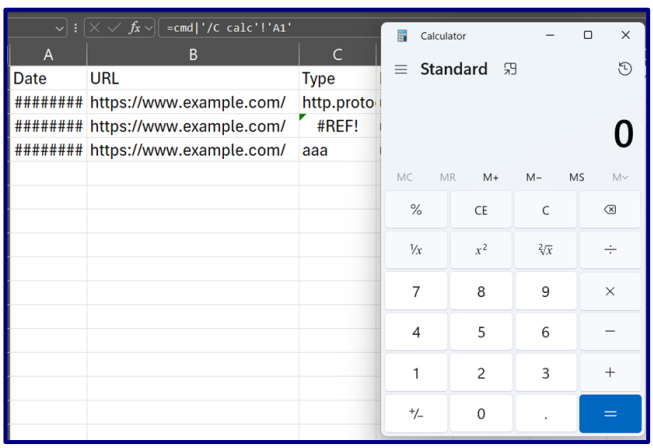


Figure 6 - CSV Injection - Payload Executed

Please note that while the parameters in the Crash request were also noted to allow arbitrary injections and were exploitable in a similar manner, there may be additional parameters across other requests that do not enforce a strict validation of the submitted values. Other requests were tested but due to the application rejecting these and consequently not displaying the requests within the report's sections, it was not possible to confirm.

It is therefore recommended that strict validation mechanisms are implemented across all requests and parameters. Where this is not possible, such as the "reason" parameter of the Crash request, the value received should be escaped or sanitised upon detection of certain control characters.

5.1.3 Risk Analysis

Pentest Risk Category	**Medium**
CVSS	7.9/High AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:L
Explanation	Initially an attacker would require access to a valid user account or have knowledge of the subdomain and filter of an organisation. The victim must interact with the vulnerable export function and accept warnings from Excel before the payload would execute. If successful, an attacker could potentially gain remote code execution on a user's machine. The impact would therefore affect the user's computer and not the backend server. This attack exploits the victim's trust in the source of the CSV file and therefore risks reputational damage. Additionally, the user would need to have certain settings enabled in Microsoft Excel. Despite the high potential impact, due to the complicated requirements to exploit this vulnerability, it has been raised as a medium risk.

5.1.4 Recommendation

Pentest recommends that all user input should never be trusted and should always be validated before processing or storing.

By adopting a white list approach for user input, i.e. only allowing certain known characters, this restricts what attackers can input into the application. This should also be applied to CSV files uploaded into the application.

Ensure the application validates user input prior to exporting to a native file format, such as .csv and .xls files. Ensure no cells begin with any of the following characters:

- Equals (=)
- Plus (+)
- Minus (-)

- At symbol (@)
- Pipe (|)

If the application requires these characters, each usage should be prefixed with a single quote (') for every formula. In the case of the pipe character (|), this should be prepended using the backslash character (\).

5.1.5 References

[1]	OWASP: CSV Injection
[2]	CWE-74: Improper Neutralisation of Special Elements
[3]	Dangers of CSV Injections

5.1.6 Affected Item(s)

- https://*.report-uri.com/a/t/g
  - NEL POST Request Parameter: type
  - Crash POST Request Parameters: reason, CrashID
- Export Functionality: https://report-uri.com
  - /ajax/time/nel/
  - /ajax/time/crash/

5.2 Vulnerabilities in Outdated Dependencies Detected

5.2.1 Background

Most software products are developed using APIs or libraries provided by 3rd parties. Doing so reduces development time and cost and feeds into the “why re-invent the wheel?” philosophy. Once a component has been integrated into an application it must be upgraded regularly to guard against bugs and remove publicly known vulnerabilities.

Failure to do so can mean that the application itself is at risk of exploitation due to weaknesses that exist in the supporting dependencies. This risk has been captured by the OWASP top 10 2021 project as category A06 labelled “Vulnerable and Outdated Components” defined at reference [1].

5.2.2 Details

The web application was seen to load external resources that were outdated and vulnerable. An example where these were seen to be loaded was as follows:

```
GET /account/ HTTP/2
Host: report-uri.com
[...]
                    Request                              Response
HTTP/2 200 OK
[...]
<script src="https://cdn.report-uri.com/libs/twitter-
bootstrap/3.4.4/js/bootstrap.min.js?v=1" integrity="sha256-
s7KNCsuBYruFfCQFt3kw1GjB8Y0Dw3lQd1YfXsXgS2c= sha384-
whOnYSp1fsKEUSg5E0hzzOriHBT/UeRfuMwNmTlmZ14Qz+fnf3DSqiohwCzmbgDi sha512-
SaB8xyEas3izDU3BfQvXiFdOoza+smQPXncQXP1iDRwtMMVjtTfA6oaeWR4FEn2QBshtUCB943ptPB5mtTsCxA=="
crossorigin="anonymous" nonce="XZ0qfXRf57iKbzp/8uUc+4+m"></script>
[...]
<script src="https://cdn.report-uri.com/libs/jquery-cookie/1.4.1/jquery.cookie.min.js?v=1"
integrity="sha256-1A78rJEdiWTzco6qdn3igTBv9VupN3Q1ozZNTR4WE/Y= sha384-
tSi+YsgNwyohDGfW/VhY51IK3RKAPYDcj1sNXJ16oRAyDP++K0NCzSCUW78EMFmf sha512-
3j3VU6WC5rPQB4Ld1jnLV7Kd5xr+cq9avvhwqzbH/taCRNURoeEpoPBK9pDyeukwSxwRPJ8fDgvYXd6SkaZ2TA=="
nonce="XZ0qfXRf57iKbzp/8uUc+4+m" crossorigin="anonymous"></script>
[...]
```

Details related to the vulnerabilities identified for each version and their severity and CVSS score are listed below:

Component Version	Vulnerabilities	Severity/CVSS
Bootstrap 3.4.4	CVE-2024-6485	6.4/Medium AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:L/A:L
jQuery Cookie 1.4.1	CVE-2022-23395	6.1/Medium AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

Pentest did not review the site’s JavaScript files to confirm exploitability. This is because auditing the JavaScript is time consuming and, though it may prove the site was secure at this time, would not prevent additions to the site making it vulnerable in the future. The only way to remove all residual risk is to apply the relevant updates.

5.2.3 Risk Analysis

Pentest Risk Category	**Low**
CVSS	6.4/Medium AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:L/A:L
Explanation	The CVSS score for this issue aligns with the one from the table above. However, it's important to note that the consultant did not attempt to exploit these vulnerabilities due to time constraints. Therefore, the severity of the vulnerability and its actual risk may require further analysis and assessment. It's crucial to consider that the absence of successful exploitation during the engagement does not guarantee the absence of risk. Additional evaluation and follow-up actions are recommended to accurately determine the impact and prioritise mitigation efforts. As such, while the issue is categorised as low due to the absence of a demonstrable impact, it is recommended to ensure timely patching and upgrading of the affected libraries.

5.2.4 Recommendation

Download and integrate the latest supported versions of each outdated dependency.

Libraries often change their API between versions so this could be a significant undertaking requiring code to be updated. To ensure that updated components do not affect the user experience, User Acceptance Testing (UAT) should be carried out after applying the latest updates.

The advice above would triage the initial problem only and would not prevent the situation from recurring. The long-term solution is to modify the Software Development Life Cycle (SDLC) to ensure that dependencies are regularly updated. OWASP provides a free tool called "dependency-check" (see reference [2]) which can be integrated into most build processes.

5.2.5 References

[1]	OWASP Top 10: A06_2021 - Vulnerable and Outdated Components
[2]	OWASP: OWASP Dependency Check
[3]	TaringAmberini: Ready to use Java Dependencies Vulnerability Checker

5.2.6 Affected Item(s)

- https://report-uri.com/account/
  - https://cdn.report-uri.com/libs/twitter-bootstrap/3.4.4/js/bootstrap.min.js
  - https://cdn.report-uri.com/libs/jquery-cookie/1.4.1/jquery.cookie.min.js

5.3 Insufficient Session Expiration

5.3.1 Background

Sessions track users across multiple requests, enabling authentication and authorisation. However, long-lived or inactive sessions increase security risks in both remote and local attack scenarios:

Remote Attacks

- Brute-force attacks on weak session identifiers or static secrets
- Session hijacking via XSS, CSRF, or clickjacking

Local Attacks

- An attacker with physical access can exploit an unattended, unlocked device
- In shared PC environments, a session may remain active if the user assumes closing the browser ends it, allowing the next user to access it

The impact of this includes a loss of confidentiality and integrity for data the victim's session can access. The specific risk rating is dependant on how challenging it is to access the session.

5.3.2 Details

The web application set the session cookie with an expiration time of 24 hours. This could increase the potential for an attacker gaining physical access to a device with an active session on the application and accessing sensitive information or functionality.

```
POST /login/auth/ HTTP/1.1
Host: report-uri.com
[...]
csrf_token=9a16a076799acd7c63b2599a34c0f517&timezone=&email=dand-
test1%2Binvite1%40pentest.co.uk&password=[...]
                    Request                          Response
HTTP/2 302 Found
Date: Tue, 25 Nov 2025 15:14:26 GMT
[...]
Set-Cookie: __Host-report_uri_sess=[censored]; expires=Wed, 26 Nov 2025 15:14:25 GMT; [...]
[...]
```

An extended session timeout increases the risk of unauthorised access to user accounts or sensitive data because inactive but still authenticated sessions remain open for longer periods. If a user forgets to log out, leaves a device unattended, or uses a shared or public computer, someone else could hijack the session without needing to reauthenticate. This prolonged exposure also heightens vulnerability to attacks such as session hijacking or cross-site scripting, where attackers exploit active sessions to gain access. In essence, longer timeouts improve convenience but significantly weaken security by extending the window of opportunity for compromise.

5.3.3 Risk Analysis

Pentest Risk Category	**Low**
CVSS	4.1/Medium AV:P/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L
Explanation	This could allow an attacker with physical or local access to the device, such as accessing a device left unattended in a public space or via malware, could gain access to the authenticated session of a user. Due to the pre-requisites of exploiting this vulnerability, the issue has been deemed a low risk.

5.3.4 Recommendation

Pentest recommends that after a period of inactivity, for example 20 minutes, the session is invalidated, and the user is required to re-authenticate to the application by entering their username and password.

5.3.5 References

[1]	OWASP: Insufficient Session Expiration
[2]	OWASP: Session Management Cheat Sheet
[3]	CWE-613: Insufficient Session Expiration

5.3.6 Affected Item(s)

- https://report-uri.com [Cookie: __Host-report_uri_sess]

5.4 Insecure TLS Configuration

5.4.1 Background

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols providing communications security over a computer network. TLS is often used to protect web application data from unauthorised disclosure and modification. It is used both between clients (web browsers) and application servers, and between application servers and other back-end components. Over time there have been several iterations of the SSL and TLS protocols, which have improved security at each iteration. All versions of the SSL protocol and early versions of the TLS protocol are no longer considered secure due to weaknesses identified in them. The specific weaknesses are detailed below.

Cipher suites are a set of instructions on how to achieve this secure transmission and consist of four elements:

- **Key exchange algorithm** - Specifies how the bulk encryption key is established in versions prior to TLS 1.3. In TLS 1.3 only pre-shared keys or Diffie-Hellman Ephemeral can be used, so this is not included.
- **Authentication algorithm** - Specifies how the client and server validate that they are communicating with the right endpoint in versions prior to TLS 1.3. In TLS 1.3 the signing algorithm is dependent on the certificate and no longer part of the cipher suite, so this is not included.
- **Bulk encryption algorithm** - Specifies what symmetric encryption algorithm is used to protect information in transit.
- **Message Authentication Code (MAC) algorithm** - Specifies what hashing algorithm is used to ensure that the message content has not been changed.

Several cipher suites suffer from publicly known issues rendering them cryptographically weak, these are detailed below.

Key Exchange/Authentication Algorithm Weaknesses:

- **RSA without Diffie-Hellman Ephemeral** - RSA without the use of Diffie-Hellman Ephemeral (DHE) for key exchange does not provide forward secrecy. This means that an attacker able to record communications would be able to decrypt them in the future if the client or server RSA keys were compromised.

Bulk Encryption Algorithm Weaknesses:

- **Cipher Block Chaining (CBC)** - While encryption using algorithms operating in CBC mode is not inherently insecure, it is difficult to implement it securely. There have been multiple vulnerabilities identified with implementations, most notably the POODLE, BEAST, and LUCKY 13 attacks, though others exist. The TLS 1.3 protocol removes support for encryption algorithms using CBC mode entirely due to these weaknesses.

HMAC Algorithm Weaknesses:

- **SHA-1** - This hashing function is no longer considered secure and is deprecated as of December 2021 in TLS 1.2. This may allow an attacker to modify data in transit without detection, though implementing such an attack in real-time would be difficult.

5.4.2 Details

The web server's TLS configuration was insecure, in the sense that it allowed encryption in CBC mode, weak hashing algorithms and non-ephemeral key exchange. A snippet from *sslscan* is shown below while the complete results can be found in the **SSL/TLS Assessment** section.

Protocol	Kex	Auth	Encrypt	Hash	Status
[...]					
TLSv1.2	RSA	RSA	AES 256 CBC	SHA	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	ECDHE	RSA	AES 128 CBC	SHA256	Weak
• Encryption operates in CBC mode					
TLSv1.2	RSA	RSA	AES 128 CBC	SHA256	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode					
TLSv1.2	ECDHE	RSA	AES 256 CBC	SHA384	Weak
• Encryption operates in CBC mode					
TLSv1.2	RSA	RSA	AES 256 CBC	SHA256	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode					

5.4.3 Risk Analysis

Pentest Risk Category	**Low**
CVSS	3.7/Low AV:A/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N
Explanation	An attacker could potentially intercept or manipulate sensitive communication, compromise confidentiality and integrity of data, and potentially facilitate man-in-the-middle attacks.

5.4.4 Recommendation

To protect against the cryptographic vulnerabilities discussed above, Pentest recommends the following configuration changes be made to the TLS/SSL service. This configuration should be reviewed carefully as resolving TLS/SSL issues can be a difficult task, and incorrect configuration can introduce more problems. However, following the recommendations as laid out below will result in a minimal attack surface being presented.

Protocol Recommendations:

- Prefer the use of TLS 1.3

Key Exchange/Authentication Algorithm Recommendations:

- **RSA without Diffie-Hellman Ephemeral** - Disable any cipher suites using RSA without the use of Diffie-Hellman Ephemeral for key exchange.
- Enable and prefer cipher suites using ephemeral key exchange and elliptic curve cryptography (e.g., ECDHE_ECDSA, ECDHE_RSA).

Bulk Encryption Algorithm Recommendations:

- Disable any encryption algorithms operating in CBC mode

HMAC Algorithm Recommendations:

- Disable the following hashing algorithm: SHA1.

Mozilla provides a tool [1] for generating secure configurations for the most common web servers. Use of this tool is highly recommended to prevent implementation errors, and the ‘Intermediate’ configuration is recommended for most publicly accessible websites. The Mozilla tool doesn't cover Microsoft IIS-based servers; however, an administration tool is available in the form of IISCrypto [2]. For Cloudflare protocols and ciphers can only be managed via the “Change Minimum TLS Version setting”, “Change ciphers setting”, and “Change TLS 1.3 setting” API calls [3].

5.4.5 References

[1]	Mozilla: SSL Configuration Generator
[2]	IIS Crypto Tool
[3]	Cloudflare API v4 Documentation

5.4.6 Affected Item(s)

- https://report-uri.com

5.5 No Account Lockout or Timeout Mechanism

5.5.1 Background

Brute-force attacks are a type of password-guessing attack used by attackers to log on applications. They work by systematically trying every combination of letters, numbers and symbols until a successful value is found.

The basic form of brute-force attacks uses dictionary words where the attacker works through a dictionary of possible passwords and tries them all, making the guessing scope significantly smaller than generating all possible passwords using random characters.

A well-designed account lockout or timeout mechanism can help reduce the effectiveness of brute-force attacks by preventing additional login attempts after a set number of failed attempts for a set period of time.

5.5.2 Details

The web application did not implement an account lockout mechanism. This could allow an attacker to launch distributed password guessing attacks against legitimate users.

As an example, a brute-force attack of 100 attempts was launched against user “dand-test3@pentest.co.uk”, using Burp Suite Intruder:

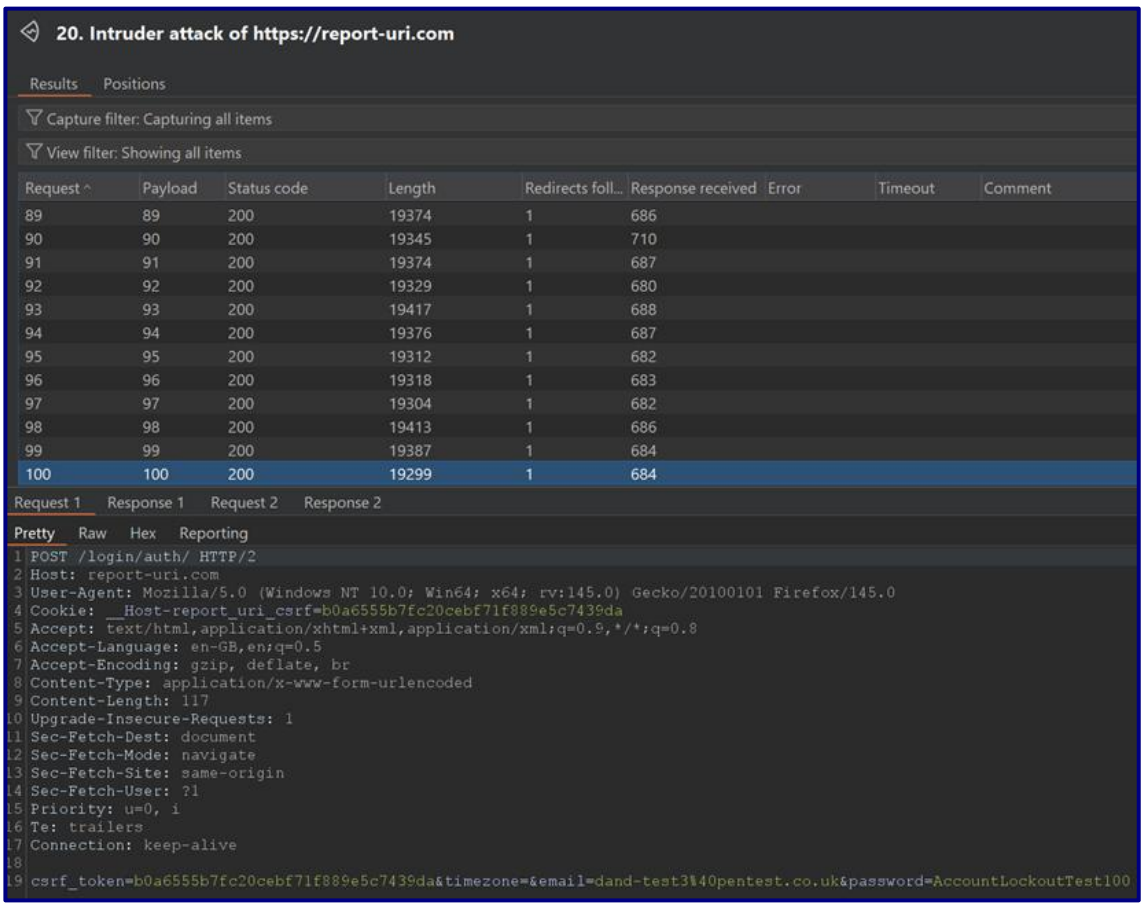


Figure 7 - Brute-force Attack - 100 Attempts

While the attack above was sent from the testing IP address with very little delay, a similar attack could be distributed across a large number of IP addresses and sent with a larger delay.

Following this, the consultant then sent an authentication attempt using the correct password for the account, and it was observed that the server returned a valid session cookie:

```
POST /login/auth/ HTTP/2
Host: report-uri.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101
Firefox/145.0
Cookie: __Host-report_uri_csrf=b0a6555b7fc20cebf71f889e5c7439da
[...]
csrf_token=b0a6555b7fc20cebf71f889e5c7439da&timezone=&email=dand-
test3%40pentest.co.uk&password=[VALID_Password]
                    Request                              Response
HTTP/2 302 Found
Date: Thu, 27 Nov 2025 12:02:17 GMT
[...]
Set-Cookie: _nss=1; path=/; secure; HttpOnly; SameSite=Strict
Set-Cookie: __Host-report_uri_csrf=d3f6418dc2c8b14aadaac342b21c2e1f; expires=Thu, 27 Nov
2025 14:02:16 GMT; Max-Age=7200; path=/; secure; HttpOnly; SameSite=Strict
Set-Cookie: __Host-report_uri_sess=bph[...]27m; expires=Fri, 28 Nov 2025 12:02:16 GMT; Max-
Age=86400; path=/; secure; HttpOnly; SameSite=Lax
[...]
<h1>Redirect</h1>

<p><a href="/account/">Please click here to continue</a>.</p><script defer src[...]
```

Please note that for the purposes of the engagement, the testing IP address was whitelisted on Cloudflare which bypassed controls that would be otherwise in place, such as the “Bot Management” controls.

5.5.3 Risk Analysis

Pentest Risk Category	**Info**
CVSS	4.3/Medium AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:L
Explanation	An attacker could launch a distributed low-intensity brute-force attack on accounts which had not enabled multi-factor authentication. An external attacker would also need to identify email addresses of accounts on the web application through other means. Additionally, the application had implemented a strong password policy and also checks the user’s password if it is weak or not before allowing the user to updated it. Therefore, due to the pre-requisites of exploiting the vulnerability and current protection mechanisms in place, the issue was deemed an informational risk.

5.5.4 Recommendation

Implement an effective account lockout/timeout mechanism to prevent password-guessing attacks. The mechanism should allow users 5 invalid login attempts before locking out the account for a reasonable period, such as 15 minutes.

This balances the risk of a denial-of-service attack on accounts and the possibility of an attacker successfully guessing account details.

5.5.5 References

[1]	CWE-307 Improper Restriction of Excessive Authentication Attempts
[2]	OWASP - Blocking Brute Force Attacks

5.5.6 Affected Item(s)

- https://report-uri.com/login/auth

6 Additional Information

6.1 WHOIS Database

The WHOIS database stores information about the individual or organisation who owns and manages a domain or IP address range. Attackers will review WHOIS entries trying to find useful information such as names and contact details for employees.

Best practices state that generic contact details should be used such as “whois@domain.com” rather than providing the name of a member of staff.

6.1.1 Entry for Domain: report-uri.com

```
   Domain Name: REPORT-URI.COM
   Registry Domain ID: 1651365076_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.namecheap.com
   Registrar URL: http://www.namecheap.com
   Updated Date: 2025-03-18T09:08:12Z
   Creation Date: 2011-04-17T11:55:31Z
   Registry Expiry Date: 2026-04-17T11:55:31Z
   Registrar: NameCheap, Inc.
   Registrar IANA ID: 1068
   Registrar Abuse Contact Email: abuse@namecheap.com
   Registrar Abuse Contact Phone: +1.6613102107
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Name Server: CARL.NS.CLOUDFLARE.COM
   Name Server: COCO.NS.CLOUDFLARE.COM
   DNSSEC: signedDelegation
   DNSSEC DS Data: 2371 13 2
B86DC8BE786CAFA5B1D92F52AA23CD9B62AF70DBE9D907AC61A1F9469513B5F6
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-11-24T09:25:21Z <<<
```

6.1.2 Entry for IP Address Range: 104.16.0.0 - 104.31.255.255

```
NetRange:       104.16.0.0 - 104.31.255.255
CIDR:           104.16.0.0/12
NetName:        CLOUDFLARENET
NetHandle:      NET-104-16-0-0-1
Parent:         NET104 (NET-104-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Cloudflare, Inc. (CLOUD14)
RegDate:        2014-03-28
Updated:        2024-09-04
Comment:        All Cloudflare abuse reporting can be done via
https://www.cloudflare.com/abuse
Comment:        Geofeed: https://api.cloudflare.com/local-ip-ranges.csv
Ref:            https://rdap.arin.net/registry/ip/104.16.0.0



OrgName:        Cloudflare, Inc.
OrgId:          CLOUD14
Address:        101 Townsend Street
City:           San Francisco
StateProv:      CA
PostalCode:     94107
Country:        US
RegDate:        2010-07-09
Updated:        2024-11-25
Ref:            https://rdap.arin.net/registry/entity/CLOUD14
```

```
OrgRoutingHandle: CLOUD146-ARIN
OrgRoutingName:   Cloudflare-NOC
OrgRoutingPhone:  +1-650-319-8930
OrgRoutingEmail:  noc@cloudflare.com
OrgRoutingRef:    https://rdap.arin.net/registry/entity/CLOUD146-ARIN

OrgNOCHandle: CLOUD146-ARIN
OrgNOCName:   Cloudflare-NOC
OrgNOCPhone:  +1-650-319-8930
OrgNOCEmail:  noc@cloudflare.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/CLOUD146-ARIN

OrgAbuseHandle: ABUSE2916-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-650-319-8930
OrgAbuseEmail:  abuse@cloudflare.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/ABUSE2916-ARIN

OrgTechHandle: ADMIN2521-ARIN
OrgTechName:   Admin
OrgTechPhone:  +1-650-319-8930
OrgTechEmail:  rir@cloudflare.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ADMIN2521-ARIN

RAbuseHandle: ABUSE2916-ARIN
RAbuseName:   Abuse
RAbusePhone:  +1-650-319-8930
RAbuseEmail:  abuse@cloudflare.com
RAbuseRef:    https://rdap.arin.net/registry/entity/ABUSE2916-ARIN

RNOCHandle: NOC11962-ARIN
RNOCName:   NOC
RNOCPhone:  +1-650-319-8930
RNOCEmail:  noc@cloudflare.com
RNOCRef:    https://rdap.arin.net/registry/entity/NOC11962-ARIN

RTechHandle: ADMIN2521-ARIN
RTechName:   Admin
RTechPhone:  +1-650-319-8930
RTechEmail:  rir@cloudflare.com
RTechRef:    https://rdap.arin.net/registry/entity/ADMIN2521-ARIN
```

6.2 DNS Reconnaissance

Domain Name Service (DNS) is used to translate human readable hostnames such as “www.pentest.co.uk” to the IP address which is hard for humans to recall. Threat actors use DNS reconnaissance to identify hosts which they can subsequently target.

6.2.1 Identifying DNS Servers for Domain: report-uri.com

The following shows the “dig” command being used to identify the name servers responsible for the target domain:

```
dig ns report-uri.com @carl.ns.cloudflare.com

; <<>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <<>> ns report-uri.com @carl.ns.cloudflare.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3426
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;report-uri.com.                        IN      NS

;; ANSWER SECTION:
report-uri.com.         86400   IN      NS      carl.ns.cloudflare.com.
report-uri.com.         86400   IN      NS      coco.ns.cloudflare.com.
```

The target used Cloudflare’s DNS service which is designed to be “always available” and has integrated support for DDoS and DNSSEC.

This was an excellent configuration and would likely ensure the availability of DNS.

6.2.2 DNS Server Configurations

The following table summarises common insecure configurations. The data was gathered by assessing each of the NS servers listed above:

Check	Outcome
Zone Transfers Disabled	TRUE
DNSSEC Enabled	TRUE
Recursive Queries Disabled	TRUE

Table 1 - DNS Server Configuration Analysis

6.3 Port Scan Results

To offer a service to other computers, a “port” is made available. Each open port creates a communication channel which can pose a security risk that an attacker can enumerate information from, or at worst exploit to compromise the target.

Best practices state that only the minimum number of open ports should be enabled to reduce the attack surface.

These open ports are the standard for installations utilising Cloudflare and do not pose additional risk.

6.3.1 Target: 104.17.214.66 - report-uri.com

Port	State	Service	Product	Version	Extra
80/tcp	open	http	cloudflare	Unknown	Unknown
443/tcp	open	https	cloudflare	Unknown	Unknown
2052/tcp	open	clearvisn	Unknown	Unknown	Unknown
2053/tcp	open	http	nginx	Unknown	Unknown
2082/tcp	open	infowave	Unknown	Unknown	Unknown
2083/tcp	open	http	nginx	Unknown	Unknown
2086/tcp	open	gnunet	Unknown	Unknown	Unknown
2087/tcp	open	http	nginx	Unknown	Unknown
2095/tcp	open	nbx-ser	Unknown	Unknown	Unknown
2096/tcp	open	http	nginx	Unknown	Unknown
8080/tcp	open	http-proxy	cloudflare	Unknown	Unknown
8443/tcp	open	https-alt	cloudflare	Unknown	Unknown
8880/tcp	open	cddbp-alt	Unknown	Unknown	Unknown

6.3.2 Target: 104.17.215.66 - report-uri.com

Port	State	Service	Product	Version	Extra
80/tcp	open	http	cloudflare	Unknown	Unknown
443/tcp	open	https	cloudflare	Unknown	Unknown
2052/tcp	open	clearvisn	Unknown	Unknown	Unknown
2053/tcp	open	http	nginx	Unknown	Unknown
2082/tcp	open	infowave	Unknown	Unknown	Unknown
2083/tcp	open	http	nginx	Unknown	Unknown
2086/tcp	open	gnunet	Unknown	Unknown	Unknown
2087/tcp	open	http	nginx	Unknown	Unknown
2095/tcp	open	nbx-ser	Unknown	Unknown	Unknown
2096/tcp	open	http	nginx	Unknown	Unknown
8080/tcp	open	http-proxy	cloudflare	Unknown	Unknown
8443/tcp	open	https-alt	cloudflare	Unknown	Unknown
8880/tcp	open	cddbp-alt	Unknown	Unknown	Unknown

6.4 SSL/TLS Assessment

Transport Layer Security (TLS) is used to ensure the confidentiality and integrity of traffic as it transits a network. It is also used to give certainty of the identity of the client, server, or both. Insecure configurations are common. The following sub-sections show information gathered using SSLScan.

6.4.1 SSLScan Results for: report-uri.com:443

Protocol	Kex	Auth	Encrypt	Hash	Status
TLSv1.3	-	-	AES 128 GCM	SHA256	Recommended
TLSv1.3	-	-	AES 256 GCM	SHA384	Recommended
TLSv1.3	-	-	CHACHA20 POLY1305	SHA256	Recommended
TLSv1.2	ECDHE	ECDSA	CHACHA20 POLY1305	SHA256	Recommended
TLSv1.2	ECDHE	ECDSA	AES 128 GCM	SHA256	Recommended
TLSv1.2	ECDHE	ECDSA	AES 128 CBC	SHA	Weak
• Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	ECDHE	ECDSA	AES 256 GCM	SHA384	Recommended
TLSv1.2	ECDHE	ECDSA	AES 256 CBC	SHA	Weak
• Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	ECDHE	ECDSA	AES 128 CBC	SHA256	Weak
• Encryption operates in CBC mode					
TLSv1.2	ECDHE	ECDSA	AES 256 CBC	SHA384	Weak
• Encryption operates in CBC mode					
TLSv1.2	ECDHE	RSA	CHACHA20 POLY1305	SHA256	Secure
TLSv1.2	ECDHE	RSA	AES 128 GCM	SHA256	Secure
TLSv1.2	ECDHE	RSA	AES 128 CBC	SHA	Weak

Protocol	Kex	Auth	Encrypt	Hash	Status
• Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	RSA	RSA	AES 128 GCM	SHA256	Weak
• Key exchange is non-ephemeral					
TLSv1.2	RSA	RSA	AES 128 CBC	SHA	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	ECDHE	RSA	AES 256 GCM	SHA384	Secure
TLSv1.2	ECDHE	RSA	AES 256 CBC	SHA	Weak
• Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	RSA	RSA	AES 256 GCM	SHA384	Weak
• Key exchange is non-ephemeral					
TLSv1.2	RSA	RSA	AES 256 CBC	SHA	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode • Hash uses weak algorithm					
TLSv1.2	ECDHE	RSA	AES 128 CBC	SHA256	Weak
• Encryption operates in CBC mode					
TLSv1.2	RSA	RSA	AES 128 CBC	SHA256	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode					
TLSv1.2	ECDHE	RSA	AES 256 CBC	SHA384	Weak
• Encryption operates in CBC mode					

Protocol	Kex	Auth	Encrypt	Hash	Status
TLSv1.2	RSA	RSA	AES 256 CBC	SHA256	Weak
• Key exchange is non-ephemeral • Encryption operates in CBC mode					

22 Great James Street
Holborn
London
WC1N 3ES