pentest
INFORMATION SECURITY ASSURANCE
A Shearwater Group plc Company

Example Corporation Ltd

Web Application Penetration Test

Sample Report

Project Ref: 1337

Author: Paul Ritchie

Table of Contents

1 Document Revision History

Name	Date	Version	Comment
Paul Ritchie		0.1	Initial Document
Rodger Campbell		0.2	QA by Senior Consultant
Paul Ritchie		1.0	Final Draft

2 Introduction

Example Corporation (referred to as Example Corp) is an organisation that operate within the UK and US to deliver example services to the public and private sectors.

Example Corp have indicated the need for a security assessment of their internal and CMS web applications, as well as the supporting network infrastructure.

The purpose of this assessment is to identify vulnerabilities to attacks that could be launched across a computer network, and to gain independent assurance that security controls are in-line with industry best practices. Such a test will allow Example Corp to undertake remediation efforts and increase their overall security posture.

2.1 Scope & Duration

This assessment included the following phase of work:

- Phase 1 - Web application testing

The duration included 2 days effort (including reporting). Work commenced DATE and concluded on DATE and our consultants were asked to focus on two specific areas of the application, as listed below:

- New Notification system
- New Team Member profile image upload

As discussed on a kick-off call on DATE, Pentest Limited also included a vulnerability assessment of the platform to verify any recent changes had been implemented securely.

2.2 Scenarios Included

This project included a “black-box” assessment of the application and supporting infrastructure. This simulated how the target would appear to an attacker who was opportunistically searching for targets. Credentials were provided to assess the two key areas meaning that a limited authenticated assessment was conducted.

2.3 Target(s)

The following URLs were provided as the target scope:

- https://internal-staging.herokuapp.com/
- https://cms-staging.herokuapp.com/

The applications are hosted using Heroku which uses elastic IP addresses. As such, the target IP addresses could not be restricted and therefore the hostnames have been used as the target in this instance.

Targets were being assessed in their “staging” environment to remove any possibility that testing would affect live users.

2.3.1 Out of scope

No areas of the application were considered as out of scope.

3 Executive Summary

No critical or high-risk vulnerabilities were confirmed within this project. Generally, the infrastructure and application had a reasonable security posture. No trivial to exploit vulnerabilities were detected which would pose a significant risk to the integrity of the server or the confidentiality of data.

Two medium risk flaws were detected as summarised below:

Title	Summary
Updating Security Details without Re-Authentication	The user is not prompted for their current password when they update the email address associated with their account. If an attacker can gain temporary access to a victim's account, then they can convert that to complete control by first altering the email address and then completing the forgotten password process. While that impact is severe for the affected account, exploitation requires either additional vulnerabilities (not detected during testing), or physical access to the victim's unattended computer. These make exploitation difficult resulting in a medium risk.
Insecure TLS Protocol Supported	TLS version 1.0 was supported by both target applications. This protocol has been deprecated due to known security weaknesses. An attacker who can conduct a man-in-the-middle (MiTM) exploitation of a user can potentially remove the confidentiality of user data sent to the server.

The first issue is a flaw created during the design and implementation of the CMS application. While the second is an insecure configuration of the supporting infrastructure.

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 Limited 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:

- **Critical Risk Items** – Address these immediately.
- **High Risk Items** – Address these as soon as possible after any Critical Risks.
- **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 Limited Consulting believe that the target's security posture will improve, making the application more robust against threats.

3.2 Caveats

Pentest Limited 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 Example Corp 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.

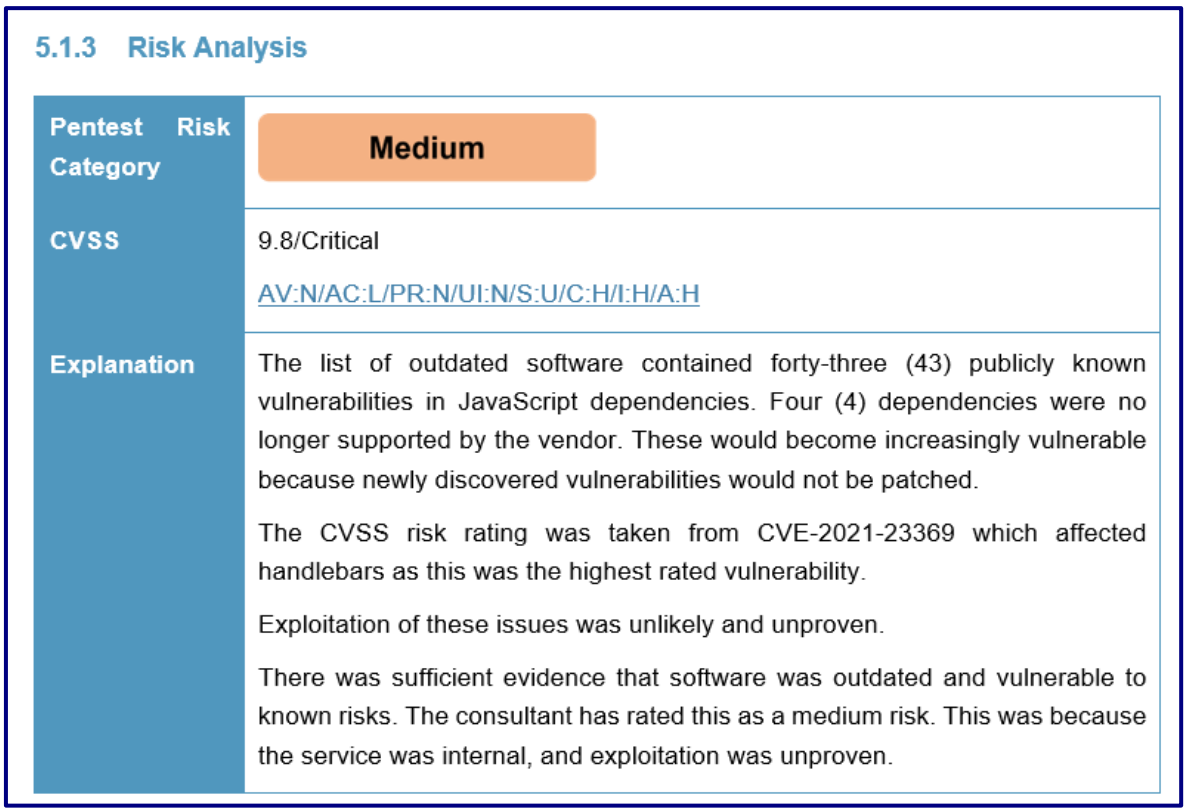
To appropriately secure assets Pentest Limited encourage a cyclical approach to assessment. Each cycle should include:

- **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 Example Corp, Pentest Limited 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.

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 & is used to prioritise remediation efforts as documented below:

Pentest Risk Category	Rationale
Critical	Poses a severe risk which may be easy to exploit. Begin remediating immediately after the issue has been presented.
High	Poses a significant risk & 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 discussion point. These are not directly exploitable but may aid attackers. Remediate to create 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.

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.

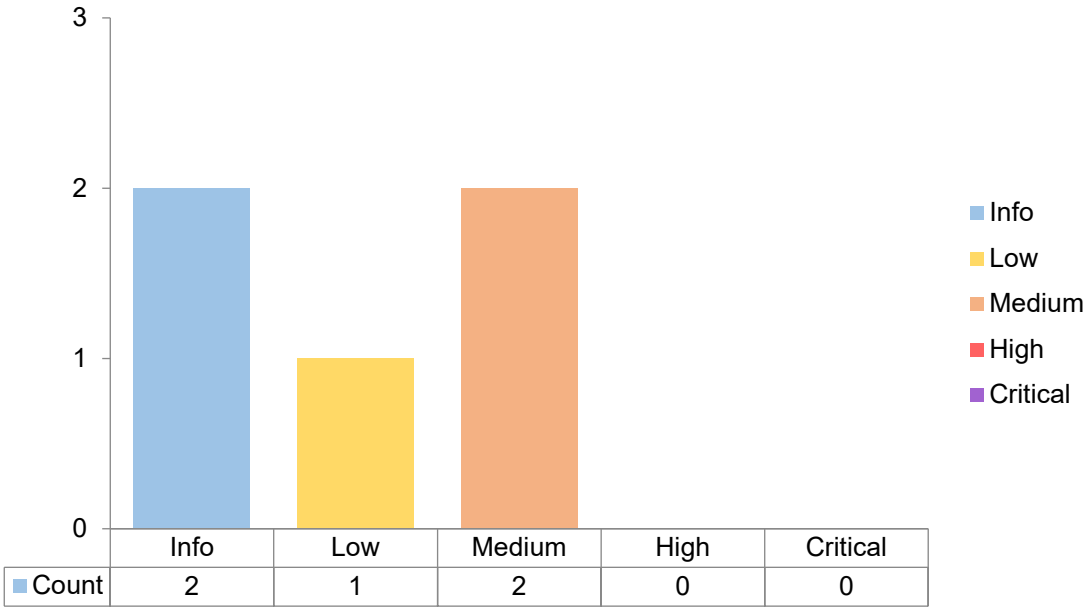
3.3.3 Rule-of-thumb Equivalency

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	Vuln Title	Recommended Action	Risk Category	CVSS
1	Updating Security Details without Re-Authentication	Require current password when altering a user's associated email address.	Medium	6.6 / Medium
2	Insecure TLS Protocol Supported	Remove support for insecure protocols – TLS 1.0.	Medium	5.3 / Medium
3	TLS Renegotiation Denial of Service	Disable TLS Renegotiation / Rate Limit TLS Handshakes.	Low	5.3 / Medium
4	Cross Domain Script Include	Consider locating the code within your application server or using a CDN for the delivery of JavaScript resources.	Info	N/A
5	Uploaded Files May Not Be Subjected to AV Scanning	Investigate the upload process and ensure that anti-virus protection can inspect files before a user can access them.	Info	N/A

5 Technical Findings

5.1 Updating Security Details without Re-Authentication

5.1.1 Background

Allowing changes to a user's security details without requiring re-authentication is a risk in either of the cases listed below:

- Attacker has temporary access to a victim's session via session hijacking, clickjacking, or XSRF.
- Attacker gains access to an unattended computer while the victim is authenticated.

The classic example of this issue is where an application enables users to set a new password without requiring the current password. An attacker with temporary access to the site can simply configure a new password thus gaining control over the victim's account.

The risks of this are captured in CWE620 listed at reference [1] below.

The impact can be significant but is dependent on which security details can be altered.

5.1.2 Details

The CMS application allows a user to alter the email address associated with their account without requiring the user to re-authenticate.

When authenticated to the application (as any user) this can be confirmed by first clicking on the icon as shown below:

Figure 1 - Locating Functionality

Which then enables the reader to access the “Edit Account” option which is displayed below:

Figure 2 - Edit Account Function

As can be seen, the form does not require the user to enter their current password.

If an attacker alters the associated email address, they can then use the forgotten password functionality to permanently hijack the victim’s account.

5.1.3 Risk Analysis

Pentest Risk Category	**Medium**
CVSS	6.6 / Medium AV:P/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Explanation	While the impact of hijacking a user account is significant, an attacker would require an additional vulnerability (which is unknown at this time) or physical access to the victim’s computer in order to exploit this. These requirements for exploitation mean that a low risk is appropriate. The CVSS base scores are in the opinion of the consultant higher than necessary. They reflect the severity of the impact (the complete loss of the victim’s account and the data they have access to), and while physical access has been built into the formula it has not reduced the base score enough in the consultant’s opinion.

5.1.4 Recommendation

Modify the “Edit Account” form to require the user to re-authenticate themselves.

5.1.5 References

[1]	CWE620 - Unverified Password Change

5.1.6 Affected Item(s)

The affected system is:

- cms-staging.herokuapp.com

5.2 Insecure TLS Protocol Supported

5.2.1 Background

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), both frequently referred to as "SSL", are cryptographic protocols that provide communications security over a computer network.

TLS provides protection of web application data from unauthorised disclosure and modification when it is transmitted between clients (web browsers) and the web application server, and between the web application server and back end and other non-browser-based enterprise components.

It is important to check the SSL/TLS protocol version as weaknesses have been identified with earlier SSL protocols, including SSLv2 and SSLv3. TLS version 1.0 also has a known cryptographic design flaw. Newer versions of TLS like versions 1.1, 1.2 and 1.3 guard against these flaws and should be used whenever possible.

Successfully exploiting this would remove the confidentiality of user data in-transit.

5.2.2 Details

The TLS configuration of the affected services accepted encrypted connections using TLS version 1.0. This protocol is considered outdated, having reached its End of Life (EOL) on the 30th of June 2018 [1].

Figure 3 lists the enabled protocols on both affected hosts:

```
sslscan cms-staging.herokuapp.com

[...] Removed for brevity [...]

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered
TLS 1.1    offered
TLS 1.2    offered (OK)
```

Figure 3 - TLS Version 1.0 Offered

Whilst the server is currently capable of using TLS version 1.1 and 1.2, the consultant noted that it was possible to force the server to downgrade the connection, as shown in Figure 4:

```
openssl s_client -connect cms-staging.herokuapp.com:443 -tls1

[...] Removed for brevity [...]

New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-SHA
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1
    Cipher    : ECDHE-RSA-AES128-SHA
    Session-ID: 818F99DE9FE8580CE7D44601D20821CD3FEBC0E319F8F70F757348ADE434A16E
```

Figure 4 - Downgrading TLS to use Outdated TLS Version 1.0

5.2.3 Risk Analysis

Pentest Risk Category	**Medium**
CVSS	5.3 / Medium AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Explanation	Whilst downgrade attacks are not uncommon, certain requirements would need to be met to allow such an attack to be successful. The attacker would need to be able to place themselves on the same network as their victim to be able to exploit this using man-in-the-middle (MiTM) techniques.

5.2.4 Recommendation

Weaknesses have been identified with the TLS protocol, including TLS version 1.0, and as such this protocol should no longer be used. TLS version 1.1 should be used a minimum, with the use of TLS version 1.2 recommended whenever possible. Furthermore, the latest version of TLS version 1.3 removes obsolete and insecure cipher suites and should be investigated further.

5.2.5 References

[1]	TLS Version 1.0 End of Life
[2]	SSL Labs - SSL/TLS Deployment Best Practices

5.2.6 Affected Items

- cms-staging.herokuapp.com – TCP port 443
- internal-staging.herokuapp.com – TCP port 443

5.3 TLS Renegotiation Denial of Service

5.3.1 Background

When a client requests a renegotiation of a TLS connection, the workload and processing power required by server is significantly higher than the client. If a client makes repeated renegotiation requests to the server, server resources can be exhausted which can result in a denial of service condition.

The impact of this is a loss of service offered by the target applications.

5.3.2 Details

Both application servers were found to allow client-initiated TLS renegotiations to take place, thus making them vulnerable to Denial of Service (DoS) attacks.

Figure 5 notes the output of the tool testssl.sh (available at reference [1]):

```
testssl.sh cms-staging.herokuapp.com

[...] Removed for brevity [...]

Testing vulnerabilities:

Ticketbleed (CVE-2016-9244)               not vulnerable (OK)
Secure Client-Initiated Renegotiation     VULNERABLE (NOT ok), DoS threat
CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
```

Figure 5 - TLS Renegotiation Vulnerability

To verify and confirm the above noted vulnerability the consultant manually renegotiated the TLS handshakes using the “openssl” s_client” tool. The following snippet in Figure 6 presents the steps undertaken by the consultant to show how an attacker could create a denial of service by renegotiating the handshakes.

```
openssl s_client -connect cms-staging.herokuapp.com:443
---
R
RENEGOTIATING
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert High Assurance EV
Root CA
verify return:1
depth=1 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert SHA2 High Assurance
Server CA
verify return:1
depth=0 C = US, ST = California, L = San Francisco, O = "Heroku, Inc.", CN =
*.herokuapp.com
verify return:1
```

Figure 6 - Manually testing client to server renegotiation

Once connected to the server, the “R” command was issued multiple times to renegotiate the connection, with the consultant attempting 10 renegotiations in quick succession. Whilst this process was done manually, it was noted that exploitation tools to automate the process of repeatedly sending renegotiation requests within a short period of time exist. The use of such exploitation tools could be used to affect the application servers; thus, performance would be diminished which may affect the availability of the application.

5.3.3 Risk Analysis

Pentest Risk Category	**Low**
CVSS	5.3 / Medium AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L
Explanation	The underlying technology upon which the application has been built, the Heroku App framework, provides connection rate limiting to mitigate Denial of Service attacks. Furthermore, it is also noted that the application also utilises the services of Cloudflare which prevent sustained or high-volume DoS attacks from taking place. As such, a low-risk rating is appropriate.

5.3.4 Recommendation

Given that the application uses Heroku for hosting, which uses elastic application servers, it is unclear if this form of DoS would be effective or not. In theory the attacker could simply connect and create more renegotiation threads throughout the application server pool. This would be harder to achieve than in traditional static application server configurations. In the area of doubt Pentest Limited recommend disabling TLS renegotiation.

To prevent the abuse of client-side renegotiation, two options can be considered – disabling TLS renegotiation and rate limiting TLS handshakes.

As the vulnerability is affected by the client-side making multiple requests to the server, TLS renegotiation can be disabled. Many versions of Apache and Nginx already use such techniques and forbid client-side renegotiation taking place. Given that disabling TLS renegotiation is not always desirable, the second option would be to consider rate limiting TLS handshakes. Further information on both techniques can be found in references [3] and [4].

5.3.5 References

[1]	Download Page for Testssl.sh
[2]	TLS Renegotiation and Denial of Service Attacks
[3]	Securing SSL Renegotiation
[4]	SSL DoS Mitigation

5.3.6 Affected Items

- cms-staging.herokuapp.com – TCP port 443
- internal-staging.herokuapp.com – TCP port 443

5.4 Cross Domain Script Include

5.4.1 Background

When an application includes a script from an external domain, that script is executed by the browser within the security context of the invoking application. The script can therefore do anything that the application's own scripts can do, such as accessing application data and performing actions within the context of the current user.

If a script is included from an external domain, then that domain is being entrusted with the data and functionality of the target application. Implicitly trusting 3rd parties like this has led to several high-profile data breaches when an attacker has been able to influence the content.

When exploited, the impact of this can be severe.

5.4.2 Details

The consultant noted that the application, internal-staging.herokuapp.com is using a JavaScript file loaded from an external source. Figure 7 shows the request and response from the application, highlighting the affected file being loaded from an external source.

```
Request:
GET / HTTP/1.1
Host: internal-staging.herokuapp.com

Response:
HTTP/1.1 200 OK
Server: Cowboy
[[Snipped ]]
<script id="Cookiebot" src="https://consent.cookiebot.com/uc.js"
```

Figure 7 - JavaScript file loaded from third party source

Similarly, the cms-staging.herokuapp.com application uses a content delivery network (CDN) in order to deliver JavaScript, as shown Figure 8.

```
Request:
GET / HTTP/1.1
Host: cms-staging.herokuapp.com

Response:
HTTP/1.1 200 OK
Server: Cowboy
[[Snipped]]
<script src="//cdn.quilljs.com/1.3.6/quill.min.js"></script>
```

Figure 8 - CDN used to deliver JavaScript

Using a Content Delivery Network is considered a more secure and robust method rather than using code hosted on a 3rd party website. However, several of those high-profile data breaches occurred when an attacker was able to influence the content stored within a CDN.

5.4.3 Risk Analysis

Pentest Risk Category	**Info**
CVSS	N/A
Explanation	Given that the JavaScript content is delivered via HTTPS, the consultant does not believe that there is risk of exploitation, hence this vulnerability has been given an informational rating.

5.4.4 Recommendation

Scripts should not be included from untrusted domains. Applications that rely on third-party scripts should consider copying the contents of these scripts onto their own domain and including them from there, this will ensure that the security posture remains robust. If that is not possible (e.g. for licensing reasons) then consider reimplementing the script's functionality within application code.

Consider using SubResource Integrity (SRI) as described at reference [2] as a means of assuring the integrity of a resource. When implemented this would ensure that SRI aware web browsers would be safe against an attacker altering the content of the resource.

5.4.5 References

[1]	OWASP – Risks of using 3rd Party JavaScript Hosting
[2]	Mozilla.org describing SubResource Integrity (SRI)

5.4.6 Affected Item

- internal-staging.herokuapp.com

5.5 Uploaded Files May Not Be Subjected to AV Scanning

5.5.1 Background

Insecurely implemented file uploads can lead to exploitable vulnerabilities with significant impacts. An often-neglected defensive step is to ensure that uploaded files are subjected to anti-virus before they are accepted. Reference [1] includes generic advice for how to assess an upload mechanism which includes this recommendation.

Failure to do so may allow an attacker to store malware on the server. A user would have a degree of trust in the server to protect them. An attacker could exploit that trust by encouraging users to interact with stored malware. The impact is therefore client side and would have not server-side impact directly.

5.5.2 Details

The upload process was using an AWS service and operated via a series of HTTP requests including the “PUT” request shown below:

```
PUT /1542018934489.txt?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAJ7D6N26QYBHHEVMA%2F20181112%2Feu-west-2%2Fs3%2Faws4_request&X-Amz-
Date=20181112T103554Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=5935332f636202c0affc3fed46ccacb64dc19a0a83871add2ff8e29d3404ef8b HTTP/1.1
Host: examplecorp-staging-temp.s3.eu-west-2.amazonaws.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://cms-staging.herokuapp.com/apps/pentestapp/teams/296/edit
Content-Type: text/plain
Content-Length: 68
Origin: https://cms-staging.herokuapp.com
Connection: close

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

The above showed an attempt to upload the EICAR anti-virus test file to the server. EICAR is a safe string which has been created to enable passive assessment of anti-virus solutions. For more details on the intended use of this file please see reference [2] below.

The response code to the above request was “200 OK” and there were no indications that the file had been rejected.

The consultant was unable to locate the download URL. Due to this, the consultant was unable to confirm if the upload functionality used Anti-Virus inspection to prevent malware being uploaded. In the area of doubt, it was deemed worth raising this as an info only risk to prompt discussion.

Additionally, it is worth noting that the file upload process was pointing to an Amazon server which was separated from the app server. This segregation would likely prevent remote code execution and would certainly move the impact to the resource server rather than the application server.

5.5.3 Risk Analysis

Pentest Risk Category	**Info**
CVSS	N/A
Explanation	No obvious errors were detected; meaning the issue was not confirmed. Without it being confirmed this is an info finding only.

5.5.4 Recommendation

Pentest Limited recommend further testing to gain additional assurance, or for clarity to be sought from Amazon around the status of anti-virus.

5.5.5 References

[1]	OWASP - Test Upload of Malicious Files
[2]	EICAR intended use

5.5.6 Affected Item(s)

The affected system is:

- examplecorp-staging-temp.s3.eu-west-2.amazonaws.com

6 Key Assessment Targets

This project included two key assessment targets as listed below:

- New Notification system
- New Team Member profile image upload

The following sub-sections deal with how to use and what was tested for each.

Anything which resulted in a risk or required action has been raised under Section 5. Given the limited focus of the project the consultant decided to show additional details of testing which is not typically included.

6.1 Testing New Notification System

6.1.1 Triggering notifications

The following steps were used to ensure that a notification was triggered:

- Visit the “internal” application (https://internal-staging.herokuapp.com/).
- Register a new account (the consultant used “paul.ritchie@pentest.co.uk”.)
- Configure an account to receive email notifications for all insight categories and a few tags as shown below:

Figure 9 - Alert Configuration

- Visit the “CMS” application (https://cms-staging.herokuapp.com) and authenticate with an account that has writer privileges. For this test the “paul.ritchie@pentest.co.uk” was used.
- Note: while the usernames are the same, the applications do not share a back-end database, and these are entirely separate accounts.
- The following screenshot shows an example insight that was created:

Figure 10 - Creating a new insight

- The highlighted part corresponds to the categories on the top right corner of Figure 9. If a user has notifications for “Enterprise” selected, and if the “Send out notification” option is enabled as shown:

Figure 11 - Selected "Send out notification"

- Then the user will receive an email no more than one hour after the insight has been published.

Finally, the following screenshot demonstrates that the insight was published and was accessible via the “Internal” application:

Figure 12 - Published Insight

The above details how to configure a user and trigger a notification.

6.1.2 Checking Email Notification

The consultant received a notification email as shown below:

Figure 13 – Example Notification Email

The email included no link to the insight or anything to drive the user to the content. Figure 14 shows part of the HTML content within the email:

```
[...] Omitted for Brevity [...]

<img src="https://REDACTED.sendgrid.net/wf/open?upn=REDACTED" alt="" width="1" height="1"
border="0" style="height:1px !important;width:1px !important;border-width:0
!important;margin-top:0 !important;margin-bottom:0 !important;margin-right:0
!important;margin-left:0 !important;padding-top:0 !important;padding-bottom:0
!important;padding-right:0 !important;padding-left:0 !important;">
```

Figure 14 - Selected "Send out notification"

The highlighted part demonstrates that user tracking is being implemented. If a user views the email and has HTML enabled, then they will unwittingly send a connection back to "REDACTED.sendgrid.net". This helps Example Corp to gather statistics.

The sendgrid host was not in scope of this assessment and no testing was conducted.

By inference the consultant determined that the email was most likely generated using a template of the form:

Figure 15 – Infered Template Design

The likely sources of the highlighted dynamic fields are:

- insight.title – set by the CMS user.
- insight.category – categories are set by some mechanism which is likely to involve an administrator. The CMS user that was provided could not influence this.

Neither source are controlled by the subscriber account which reduces the chances of an exploitable weakness such as template injection.

However, there is the so-called "insider threat" posed by a user with CMS access privileges since they can control the title field. To assess this the consultant used automated scanning and manual techniques. This resulted in over one thousand email notifications being issued as demonstrated below:

Figure 16 - Notifications Arriving

No exploitable weakness was detected within that notification template using currently known techniques.

6.2 Testing New Team Member profile image upload

6.2.1 Uploading a new profile Image

This feature is accessible from the CMS application for user accounts with the “Create new Team Member” and/or “Manage Teams” options as highlighted below:

Figure 17 - Required Privileges

The profile picture upload function is accessed via either the new or edit team member options with the specific field highlighted below:

Figure 18 - Profile Picture Option

During this project Paul.Ritchie@pentest.co.uk accounts had these privileges.

6.2.2 Checking for vulnerabilities

A manual attempt to upload the “EICAR” test file was made. This test was inconclusive, so an info risk finding was raised in Section 5.5.

Additional manual and automated scanning techniques were used to check the upload process for weaknesses. This testing did not uncover any exploitable security weaknesses.

A potential issue of weak input validation of uploaded file types was detected. The application allowed “.txt” files (or other extensions) to be uploaded when by design only image files are required. However, no method was detected to later retrieve these files directly which prevented exploitation or further analysis.

The application used an AWS mechanism for file uploads which resulted in segregation between:

- The upload server; and
- The application server

In theory this segregation would mitigate the risks of a web shell being uploaded. Any remote code execution would be in the context of the upload server and not within the application server. This segregation would limit the risks posed in various scenarios.

7 Additional Information

7.1 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.

7.1.1 Entry for Domain: herokuapp.com

The domain is owned by Heroku and no personal information related to Example Corp was detected. This configuration did not raise any concerns.

```
whois herokuapp.com

[...] Removed for brevity[...]

The Registry database contains ONL;llY .COM, .NET, .EDU domains and
Registrars.
Domain Name: herokuapp.com
Registry Domain ID: 1616440759_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2018-07-02T21:22:25-0700
Creation Date: 2010-09-18T22:55:31-0700
Registrar Registration Expiration Date: 2023-09-18T22:55:31-0700
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited
(https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Registry Registrant ID:
Registrant Name: Domain Administrator
Registrant Organization: Salesforce.com, Inc.
Registrant Street: 1 Market Street, Suite 300
Registrant City: San Francisco
Registrant State/Province: CA
Registrant Postal Code: 94105
Registrant Country: US
Registrant Phone: +1.4159017000
Registrant Phone Ext:
Registrant Fax: +1.4159017000
Registrant Fax Ext:
Registrant Email: registrar-updates@salesforce.com
Registry Admin ID:
Admin Name: Domain Administrator
Admin Organization: Salesforce.com, Inc.
Admin Street: 1 Market Street, Suite 300
Admin City: San Francisco
Admin State/Province: CA
Admin Postal Code: 94105
Admin Country: US
Admin Phone: +1.4159017000
Admin Phone Ext:
Admin Fax: +1.4159017000
Admin Fax Ext:
Admin Email: registrar-updates@salesforce.com
Tech Organization: Heroku, Inc
Tech State/Province: CA
Tech Country: US
Name Server: ns-662.awsdns-18.net
Name Server: ns-1378.awsdns-44.org
```

```
Name Server: ns-1624.awsdns-11.co.uk
Name Server: ns-505.awsdns-63.com
DNSSEC: unsigned
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
>>> Last update of WHOIS database: 2018-11-08T01:47:39-0800 <<<
```

7.1.2 Entry for IP Address Range: REDACTEDIPRANGE

The IP address of the application was fluid throughout the test. The target was hosted in amazon's AWS cloud infrastructure and none of the information was related to Example Corp.

The following shows that all netblocks were owned by Amazon with one example:

```
NetRange:       REDACTEDIPRANGE
CIDR:           REDACTEDIPRANGE
NetName:        AMAZON-DUB
NetHandle:      REDACTED
Parent:         REDACTED
NetType:        Reallocated
OriginAS:       AS16509
Organization:   Amazon Data Services Ireland Limited (ADSIL-1)
RegDate:        2016-11-30
Updated:        2016-11-30
Ref:            https://rdap.arin.net/registry/ip/REDACTED


OrgName:        Amazon Data Services Ireland Limited
OrgId:          ADSIL-1
Address:        Unit 4033, Citywest Avenue Citywest Business Park
City:           Dublin
StateProv:      D24
PostalCode:
Country:        IE
RegDate:        2014-07-18
Updated:        2014-07-18
Ref:            https://rdap.arin.net/registry/entity/ADSIL-1


OrgTechHandle: ANO24-ARIN
OrgTechName:   Amazon EC2 Network Operations
OrgTechPhone:  +1-206-266-4064
OrgTechEmail:  amzn-noc-contact@amazon.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ANO24-ARIN

OrgNOCHandle: AANO1-ARIN
OrgNOCName:   Amazon AWS Network Operations
OrgNOCPhone:  +1-206-266-4064
OrgNOCEmail:  amzn-noc-contact@amazon.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/AANO1-ARIN

OrgAbuseHandle: AEA8-ARIN
OrgAbuseName:   Amazon EC2 Abuse
OrgAbusePhone:  +1-206-266-4064
OrgAbuseEmail:  abuse@amazonaws.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/AEA8-ARIN

# end


# start

NetRange:       REDACTED
CIDR:           REDACTED
NetName:        AT-88-Z
NetHandle:      REDACTED
Parent:         NET34 (NET-34-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Amazon Technologies Inc. (AT-88-Z)
RegDate:        2016-09-12
Updated:        2016-09-12
```

```
Ref:            https://rdap.arin.net/registry/ip/REDACTED


OrgName:        Amazon Technologies Inc.
OrgId:          AT-88-Z
Address:        410 Terry Ave N.
City:           Seattle
StateProv:      WA
PostalCode:     98109
Country:        US
RegDate:        2011-12-08
Updated:        2017-01-28
Comment:        All abuse reports MUST include:
Comment:        * src IP
Comment:        * dest IP (your IP)
Comment:        * dest port
Comment:        * Accurate date/timestamp and timezone of activity
Comment:        * Intensity/frequency (short log extracts)
Comment:        * Your contact details (phone and email) Without these we will be unable to
identify the correct owner of the IP address at that point in time.
Ref:            https://rdap.arin.net/registry/entity/AT-88-Z


OrgTechHandle: ANO24-ARIN
OrgTechName:   Amazon EC2 Network Operations
OrgTechPhone:  +1-206-266-4064
OrgTechEmail:  amzn-noc-contact@amazon.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ANO24-ARIN

OrgAbuseHandle: AEA8-ARIN
OrgAbuseName:   Amazon EC2 Abuse
OrgAbusePhone:  +1-206-266-4064
OrgAbuseEmail:  abuse@amazonaws.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/AEA8-ARIN

OrgNOCHandle: AANO1-ARIN
OrgNOCName:   Amazon AWS Network Operations
OrgNOCPhone:  +1-206-266-4064
OrgNOCEmail:  amzn-noc-contact@amazon.com
OrgNOCRef:    https://rdap.arin.net/registry/entity/AANO1-ARIN
```

7.2 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.

7.2.1 Target: internal-staging.herokuapp.com

The following shows only the expected HTTP and HTTPS port open on the target, as is the recommended configuration:

```
nmap -sS -P0 --reason -p 1-65535 -sV -A internal-staging.herokuapp.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-08 14:19 GMT
Nmap scan report for internal-staging.herokuapp.com

[...] Omitted for brevity [...]

Not shown: 65533 filtered ports
Reason: 65533 no-responses
PORT    STATE SERVICE  REASON          VERSION
80/tcp  open  http     syn-ack ttl 235 Obscured httpd
|_http-server-header: Obscured
|_http-title: REDACTED
443/tcp open  ssl/http syn-ack ttl 235 Obscured httpd
|_http-server-header: Obscured
|_http-title: REDACTED
```

The following shows that no common UDP services were confirmed as open:

```
nmap -sU -P0 --reason --top-ports 5000 internal-staging.herokuapp.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-08 14:36 GMT
Nmap scan report for internal-staging.herokuapp.com

[...] Omitted for brevity [...]

All 5000 scanned ports on internal-staging.herokuapp.com (REDACTEDIP1) are open|filtered
because of 5000 no-responses
```

7.2.2 Target: cms-staging.herokuapp.com

The following shows only the expected HTTP and HTTPS port open on the target, as is the recommended configuration:

```
nmap -sS -P0 --reason -p 1-65535 -sV -A cms-staging.herokuapp.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-08 14:27 GMT
Nmap scan report for cms-staging.herokuapp.com

[...] Omitted for brevity [...]

Not shown: 65533 filtered ports
Reason: 65533 no-responses
PORT    STATE SERVICE  REASON          VERSION
80/tcp  open  http     syn-ack ttl 235 Obscured httpd
|_http-server-header: Obscured
|_http-title: REDACTED
443/tcp open  ssl/http syn-ack ttl 235 Obscured httpd
|_http-server-header: Obscured
|_http-title: REDACTED
```

The following shows that no common UDP services were confirmed as open:

```
nmap -sU -P0 --reason --top-ports 5000 cms-staging.herokuapp.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-08 14:39 GMT
Nmap scan report for cms-staging.herokuapp.com

[...] Omitted for brevity [...]

All 5000 scanned ports on cms-staging.herokuapp.com (REDACTEDIP2) are open|filtered because
of 5000 no-responses
```

7.3 SSL/TLS Assessment

7.3.1 SSLScan Results for: cms-staging.herokuapp.com

```
Testing SSL server cms-staging.herokuapp.com on port 443 using SNI name cms-
staging.herokuapp.com

TLS Fallback SCSV:
Server supports TLS Fallback SCSV

TLS renegotiation:
Secure session renegotiation supported

TLS Compression:
Compression disabled

Heartbleed:
TLS 1.2 not vulnerable to heartbleed
TLS 1.1 not vulnerable to heartbleed
TLS 1.0 not vulnerable to heartbleed

 Supported Server Cipher(s):
Preferred TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.2  128 bits  AES128-GCM-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA
Accepted  TLSv1.2  256 bits  AES256-GCM-SHA384
Accepted  TLSv1.2  256 bits  AES256-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA
Preferred TLSv1.1  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.1  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.1  128 bits  AES128-SHA
Accepted  TLSv1.1  256 bits  AES256-SHA
Preferred TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.0  128 bits  AES128-SHA
Accepted  TLSv1.0  256 bits  AES256-SHA

SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.herokuapp.com
Altnames: DNS:*.herokuapp.com, DNS:herokuapp.com
Issuer:   DigiCert SHA2 High Assurance Server CA

Not valid before: Apr 19 00:00:00 2017 GMT
Not valid after:  Jun 22 12:00:00 2020 GMT
```

The highlighted part confirmed that TLS v 1.0 was enabled which is against best practices.

7.3.2 SSLScan Results for: pentest-internal.kerokuapp.com

```
Testing SSL server internal-staging.herokuapp.com on port 443 using SNI name internal-
staging.herokuapp.com

TLS Fallback SCSV:
Server supports TLS Fallback SCSV

TLS renegotiation:
Secure session renegotiation supported

TLS Compression:
Compression disabled

Heartbleed:
TLS 1.2 not vulnerable to heartbleed
TLS 1.1 not vulnerable to heartbleed
TLS 1.0 not vulnerable to heartbleed

Supported Server Cipher(s):
Preferred TLSv1.2  128 bits  ECDHE-RSA-AES128-GCM-SHA256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA256
Accepted  TLSv1.2  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA384
Accepted  TLSv1.2  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.2  128 bits  AES128-GCM-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA256
Accepted  TLSv1.2  128 bits  AES128-SHA
Accepted  TLSv1.2  256 bits  AES256-GCM-SHA384
Accepted  TLSv1.2  256 bits  AES256-SHA256
Accepted  TLSv1.2  256 bits  AES256-SHA
Preferred TLSv1.1  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.1  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.1  128 bits  AES128-SHA
Accepted  TLSv1.1  256 bits  AES256-SHA
Preferred TLSv1.0  128 bits  ECDHE-RSA-AES128-SHA
Accepted  TLSv1.0  256 bits  ECDHE-RSA-AES256-SHA
Accepted  TLSv1.0  128 bits  AES128-SHA
Accepted  TLSv1.0  256 bits  AES256-SHA

  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.herokuapp.com
Altnames: DNS:*.herokuapp.com, DNS:herokuapp.com
Issuer:   DigiCert SHA2 High Assurance Server CA

Not valid before: Apr 19 00:00:00 2017 GMT
Not valid after:  Jun 22 12:00:00 2020 GMT
```

pentest

INFORMATION SECURITY ASSURANCE

A Shearwater Group plc Company

22 Great James Street
Holborn
London
WC1N 3ES

Telephone: +44 (0)1506 888 843
Email: Contact@pentest.co.uk