Orange
Cyberdefense

Evermood

Web Application Retest Assessment

Project Code	**SP80363** **(Retest of SP77311)**
Date	**2024-01-29**

orange™

Contents

1 Executive Summary

1.1 Assessment Overview

The information security assessment of Evermood's application commenced on the 27th of November 2023 and concluded on the 1st of December 2023.

A subsequent retest was performed on the 22nd of January 2024, and concluded on the 23rd of January 2024.

Evermood engaged the services of Orange Cyberdefense to:

- Evaluate whether corporate security requirements and best practices were followed during the development and deployment of the in-scope system and the associated environment.
- Gauge whether the risk identified within the environment was at a level acceptable to the organisation and that such risk would not have a significant impact on the delivery of the application, expose clients to harm or loss, or other such consequences.
- Evaluate whether findings from the initial assessment were resolved.

The results provided are the output of the security assessment performed. They should be used as input into a broader risk management process.

These results are a point in time assessment of the system and environment as they were presented for testing. Any changes could yield a different set of results.

To differentiate between issues that have been resolved and current issues after the retest, resolved content will be coloured in grey.

1.1.1 Risk Summary

The overall information security risk rating following the retest was calculated as: **Medium**

This is based on the following statistics:

0	0	1	0	0	1
Critical	High	Medium	Low	Info	Total

Table 1 – Retest Risk Summary. See Appendix B.2 for the Qualitative Severity Rating Scale (QSR).

The overall information security risk rating prior to the retest was calculated as: **Medium**

This was based on the following statistics:

0	0	4	1	0	5
Critical	High	Medium	Low	Info	Total

Table 2 – Original Risk Summary. See Appendix B.2 for the Qualitative Severity Rating Scale (QSR)

1.2 Qualitative Severity Rating Summary

Critical QSR: Such attacks could have a catastrophic impact on the confidentiality, integrity and availability of the systems and the business. This could result in a significant financial loss, significant reputational damage, serious legal and compliance-related fines, and other effects on the business.

High QSR: Such attacks could have a significant impact on the confidentiality, integrity and availability of the systems and the business. This could result in a significant financial loss, significant reputational damage, serious legal and compliance-related fines, and other effects on the business.

Medium QSR: Such attacks could have a noticeable impact on the confidentiality, integrity and availability of the systems and the business, which could result in a noticeable financial loss, considerable reputational damage, legal and compliance-related fines, and other effects on the business.

Low QSR: Such attacks are unlikely to have a noticeable impact on the business. However, such issues do not exist in isolation. An attacker may use them as part of a more complicated, blended attack.

Info QSR: Such attacks have no direct impact on the business. However, such issues do not exist in isolation. An attacker may use them as part of a more complicated, blended attack.

1.3 Security Overview

What (did we assess)

Re-evaluation of the previously identified findings of Evermood’s web application.

How

Manual testing of common issues such as authentication bypass, injection attacks and data manipulation to just name a few.

Why

To verify whether the findings found during the initial assessment were resolved.

Results

A great amount of effort was made to remediate the findings, and as a result, all but one Medium-risk finding remained unresolved.

Risk

The worst-case impact to the business of an attack could be:

Reputation/customer confidence: Minimal

Financial Fraud: None

Productivity: Moderate

Safety and health: None

Fines/legal penalties: None

Next Steps

Investigate the remaining finding and define an effective rate-limiting policy, addressing this in a timely manner. It is also recommended that a CAPTCHA[1] solution be implemented as a further means of securing the chat functionality of the application.

Difficulty to Fix

The rate-limiting and CAPTCHA solution should not be too difficult to implement.

[1] https://developers.google.com/recaptcha/

1.4 Conclusion and Recommendations

Overall, the security posture of the assessed application was moderate, and generally followed industry-standard and best practices.

The consistent usage of modern security headers and session authentication tokens ensured that user requests could not be hijacked by attackers, whilst the practices put in place on the login screens prevented attackers from attacking a user's account in any meaningful way. However, certain issues were found that required attention to increase the overall security posture.

Of greatest concern was the lack of adequate data sanitisation on the edit counsellor page. It was possible to edit the about me and FAQ fields which allowed for a persistent Cross-Site Scripting (XSS) attack. This made it possible to inject custom JavaScript code to be run inside the browser of any user that happened to visit the counsellor’s page. All fields should be sufficiently sanitised to ensure that the in-place restrictions cannot be bypassed and that no malicious content may be added to the system.

A file upload function vulnerability was also found in the about me and FAQ page regarding the lack of file type and content checking. The upload function had no checks or restrictions to see if the file was the correct content or was malicious, this led to it being possible to upload malicious files including that of the EICAR anti-virus test file. It was not possible to execute any malicious payloads due to user interaction being required. Strict validation should be performed on all content being uploaded as well as restrictions should be implemented on what file types can be uploaded.

Additionally, there was no rate limitation implemented on the chat endpoint when a user modified the request slightly. This made it possible to send a high number of requests with no restrictions. This could have a potential impact on the service's performance as well as storage space related issues.

During the retest, most of the previously identified vulnerabilities had been fixed – with one exception. The data sanitisation and filtering across the platform was significantly improved from the original assessment, disallowing JavaScript logic and unexpected values from being submitted. Similarly, the file upload functionality had also been improved to appropriately check the file types and content being submitted. That said, a lack of rate limiting was still present on the chat endpoint.

While using the chat endpoint, it was still possible to send many requests in a short period of time; approximately 600 requests per minute. This could impact the usability and performance of the application and as such, it is recommended that rate limitation is put in place.

Evermood should distribute the technical results of this document to the relevant teams so a full evaluation of the issues can be conducted, and an appropriate mitigation plan defined.

The following strategic recommendations have been determined based on an interpretation of the results identified during the project:

Business Area	Strategic Recommendations
Systems Management	Defining practical security baselines for common components, and auditing against them, will help to reduce the number of common vulnerabilities present. Several such benchmarks are available at https://benchmarks.cisecurity.org/ and can be used to create company specific benchmarks. Additionally, building such benchmarks into base images can help create a "secure by default" approach to new systems.

Table 3 – Strategic Recommendations.

2 Project Summary

2.1 Assessment Scope

Orange Cyberdefense was tasked with performing a security assessment against Evermood's application. This commenced on the 27th of November 2023 and concluded on the 1st of December 2023.

A subsequent retest was performed on the 22nd of January 2024, and concluded on the 23rd of January 2024.

The assessment followed the Orange Cyberdefense methodologies, which can be viewed in Appendix C. The target's included during the assessment were:

Targets	Scope
User Platform https://orange.evermood.com/ **Manager Dashboard** https://orange.evermood.com/dashboard	A full assessment of the web application was requested, including the following key test areas: • Authentication • Session Management • Access Control • Data Validation • Authorisation • Business Logic • Data Encryption • Configuration Management The following test area was not considered to be within the scope: • Denial-of-Service (DoS) Testing

Table 4 – Assessment Scope.

2.2 Assessment Timeline

Table 5 provides a breakdown of the timeline of the assessment.

Date	Activity
2023-11-27	Basic unauthenticated familiarisation and scanning.
2023-11-28	Familiarisation, scanning and enumeration of the web application. Testing of the authentication, dashboard and chat functionality.
2023-11-29	Session management testing and further scanning.
2023-11-30	Tested the counsellor's portal functionality. Verified Cross-Site Scripting (XSS) and file upload vulnerabilities. Assessed potential avenues for exploitation arising from the identified weaknesses.
2023-12-01	Data collation and initial report generation.
2024-01-22	Retest conducted on previously identified findings.
2024-01-23	Data collation and retest report generation.

Table 5 – Assessment Timeline.

2.3 Project Contacts

Table 6 lists the Orange Cyberdefense staff that were involved with the project and their contact details.

Person	Role	Contact Details
Marius Van Der Sandt	Account Manager	marius.vandersandt@orangecyberdefense.com +27 12 460 0880
Bernice Kotze	Project Manager	bernice.kotze@orangecyberdefense.com +27 12 460 0880 Ext 206
James Hill	Security Analyst (Original)	james.hill@orangecyberdefense.com
Lauren Skinner	Security Analyst (Retest)	lauren.skinner@orangecyberdefense.com

Table 6 – Contact Persons.

The contact person at Evermood was Tobias Rohloff. The analysts would like to thank them for the assistance and insight provided during the assessment.

3 Findings Summary

In total, five security issues were identified during the original assessment, of which four were classified as a Medium risk and one was classified as a Low risk.

During the retest, four security issues were found to be resolved, of which three were classified as Medium and one as Low risk. One security issue was unresolved, which was classified as Medium risk.

Table 7 provides an overview of issues identified. The Qualitative Severity Rating method used can be viewed in Appendix B.2.

Issue	Risk	Title	Retest Results
R01	**Medium**	Persistent Cross-Site Scripting	**Resolved**
R02	**Medium**	Inadequate HTTP API Rate Limiting	**Unresolved**
R03	**Medium**	File Upload Vulnerability	**Resolved**
R04	**Medium**	Input Reflected as HTML in Response	**Resolved**
R05	**Low**	Missing HTTP Security Headers	**Resolved**

Table 7 – Assessment Results Summary.

4 Assessment Results

R01	Persistent Cross-Site Scripting			
Risk Rating	Medium	CVSS3	CVSS3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:L/A:N	6.3

Technical Overview

The application was vulnerable to a persistent Cross-Site Scripting (XSS) attack. The application allowed a councillor to update their account information specifically the About Manager and FAQ text without sainting the input once saved. This information was stored and would be displayed to other users of the site. The application did not perform data validation and as such scripting code, which would execute in other users' browsers, could be entered and stored in the application.

Potential Impact if Exploited

A successful attack would allow an attacker to execute malicious code in users' browsers. This could allow the attacker to potential perform malicious actions or steal session information. However, during the test this was not possible due to the secure implementation of CSRF token and session management.

Recommendations

Full and complete validation should be performed on all user supplied or otherwise tainted data, and suitable encoding applied where appropriate. These validation checks must be performed on the server side as well, where malicious users cannot tamper with the validation routines. All data should be suitably encoded before being sent to the user's browser. For further information about XSS, please see: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

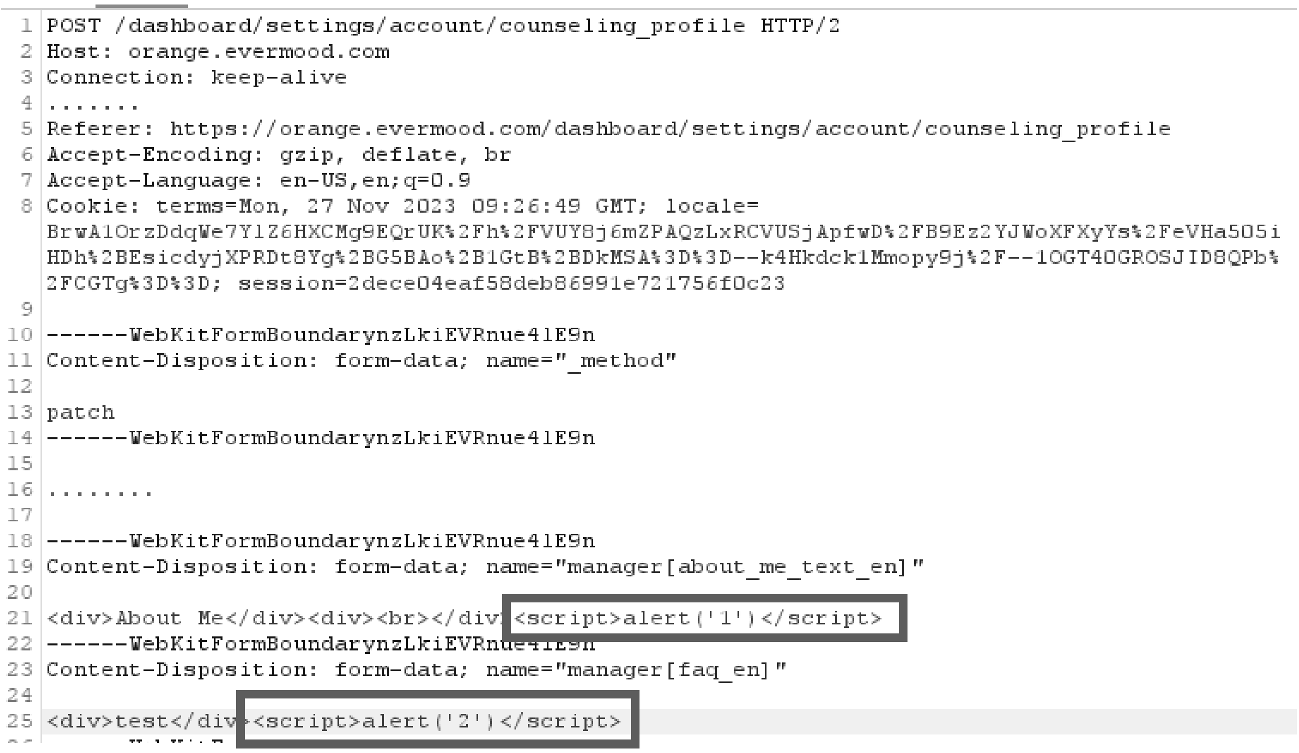
Instances

The following parameters were vulnerable when updating a counsellor’s information:

```
manager[about_me_text_en]
manager[faq_en]
manager[about_me_text_de]
manager[faq_de]
```

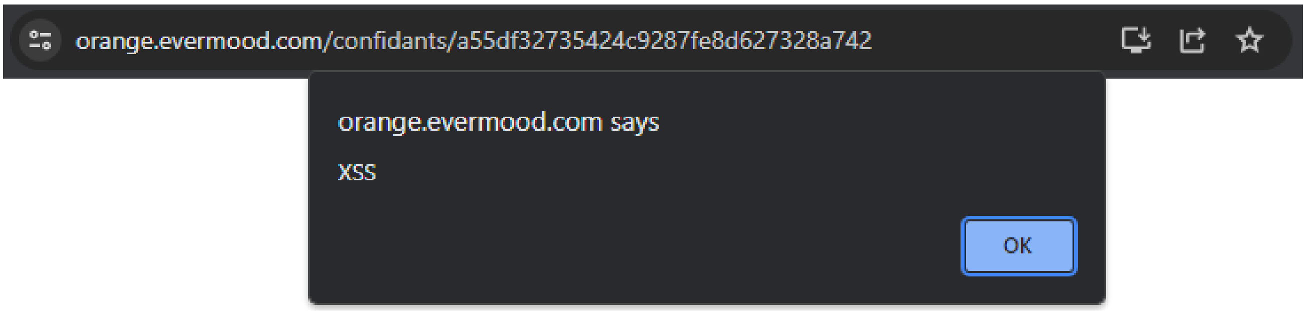
Attack Example

When entering malicious JavaScript into vulnerable fields on the browser, the application successfully sanitised the input. However, when intercepting a request using BURP[2] and changing the parameters before it was sent to the server the same sanitisation did not take place.



```
POST /dashboard/settings/account/counseling_profile HTTP/2
Host: orange.evermood.com
Connection: keep-alive
.......
Referer: https://orange.evermood.com/dashboard/settings/account/counseling_profile
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: terms=Mon, 27 Nov 2023 09:26:49 GMT; locale=
BrwA1OrzDdqWe7Y1Z6HXCMg9EQrUK%2Fh%2FVUY8j6mZPAQzLxRCVUSjApfwD%2FB9Ez2YJWoXFXyYs%2FeVHa5O5i
HDh%2BEsicdyjXPRDt8Yg%2BG5BAo%2B1GtB%2BDkMSA%3D%3D--k4Hkdck1Mmopy9j%2F--1OGT4OGROSJID8QPb%
2FCGTg%3D%3D; session=2dece04eaf58deb86991e721756f0c23

------WebKitFormBoundarynzLkiEVRnue41E9n
Content-Disposition: form-data; name="_method"

patch
------WebKitFormBoundarynzLkiEVRnue41E9n

........

------WebKitFormBoundarynzLkiEVRnue41E9n
Content-Disposition: form-data; name="manager[about_me_text_en]"

<div>About Me</div><div><br></div><script>alert('1')</script>
------WebKitFormBoundarynzLkiEVRnue41E9n
Content-Disposition: form-data; name="manager[faq_en]"

<div>test</div><script>alert('2')</script>
```

Figure 1 – Injected JavaScript code in the about_me and FAQ parameters.



orange.evermood.com/confidants/a55df32735424c9287fe8d627328a742

orange.evermood.com says

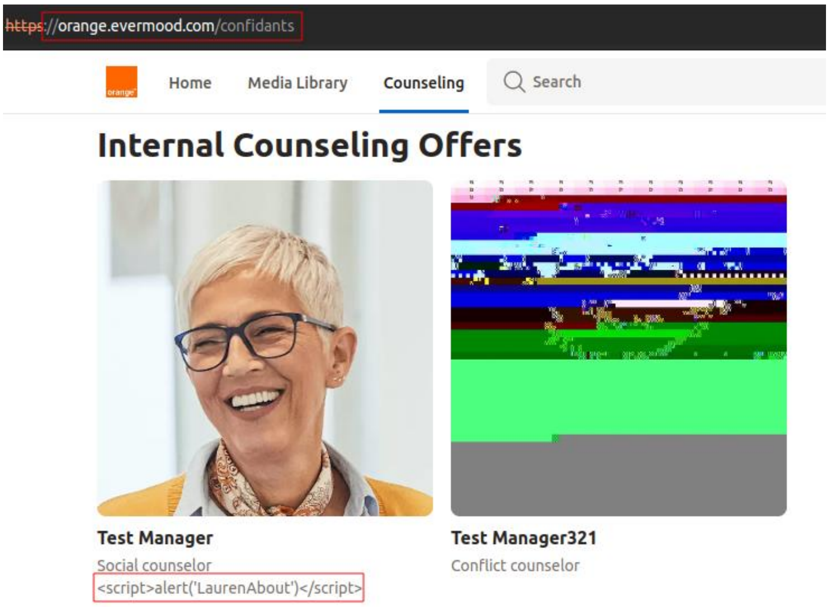
XSS

Figure 2 – Result of the injected JavaScript code in the browser upon visiting the affected counsellors page.

[2] https://portswigger.net/burp

Retest Results

During the retest, this finding was found to be resolved. When updating a Counsellor’s information section, several attempts to inject JavaScript code were not successful. The code did not trigger any client-side behaviour. Instead, the input fields were appropriately sanitised, and the script parameters removed.



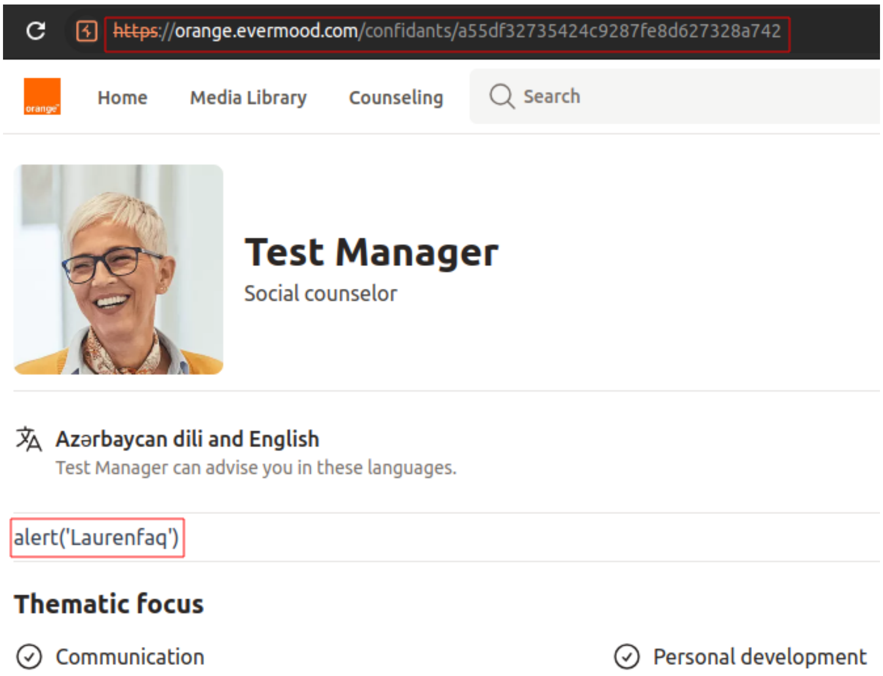
~~https~~://orange.evermood.com/confidants

Internal Counseling Offers

Test Manager
Social counselor
<script>alert('LaurenAbout')</script>

Test Manager321
Conflict counselor

Figure 3 – The injected JavaScript code was not executed after saving the changes and navigating to the Counseling homepage.



~~https~~://orange.evermood.com/confidants/a55df32735424c9287fe8d627328a742

Test Manager
Social counselor

Azərbaycan dili and English
Test Manager can advise you in these languages.

alert('Laurenfaq')

Thematic focus

Communication
Personal development

Figure 4 – The script input tag appeared to be sanitised, therefore did not trigger a popup message when navigating to the Counselor’s profile.

R02	Inadequate HTTP API Rate Limiting			
Risk Rating	Medium	CVSS3	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	5.3

Technical Overview

The API associated with the Evermood application did not implement any rate limiting, therefore allowing for an excessive number of requests to be repeated. Over 1000 messages could be sent to the chat functionality in a short period of time.

Potential Impact if Exploited

While this did not pose an immediate risk, users could clog the server with extraneous messages. As multiple threads can be made anonymously, this could lead to the dashboard being filled with extraneous threads as well. In the worst case, the messages could cause a denial of service to legitimate users.

Recommendations

The functionality should make use of adequate rate limiting, only allowing a reasonable number of messages to be sent within a period of time. Additionally, a captcha solution could be implemented prior to the chat function being enabled. It is also recommended that any successful chat request is only sent with a valid authenticity_token.

Instances

https://orange.evermood.com/chat

Attack Example

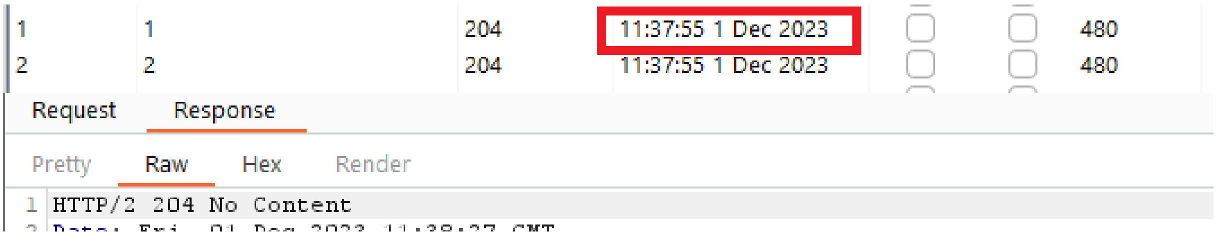
This attack was only possible by removing the authenticity_token from the request. When the access code was integrated into the request, it appeared as though a rate limiting mechanism had been implemented.

The body of modified request was changed from:

```
authenticity_token=zdClN4GcZ3khH6vom2wVfMpKtUzWhWIUxDK8T4bgXpirb44XT1BVHdamhAsuI7m_z5cEN1yTJEwn7aaVDIw3jg&chat_message%5Btext%5D=123&chat_message%5Baccess_code%5D=5IGJNA
```

To the following:

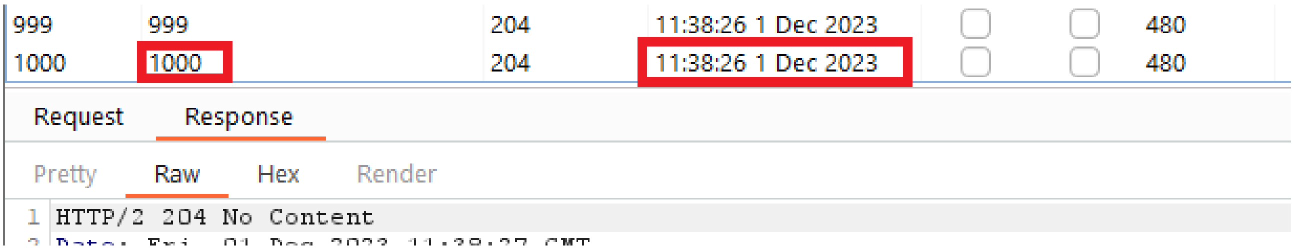
```
chat_message%5Btext%5D=123
```



1	1	204	11:37:55 1 Dec 2023	□	□	480
2	2	204	11:37:55 1 Dec 2023	□	□	480

```
HTTP/2 204 No Content
```

Figure 5 – The first request was sent at 11:37:55.



999	999	204	11:38:26 1 Dec 2023	□	□	480
1000	1000	204	11:38:26 1 Dec 2023	□	□	480

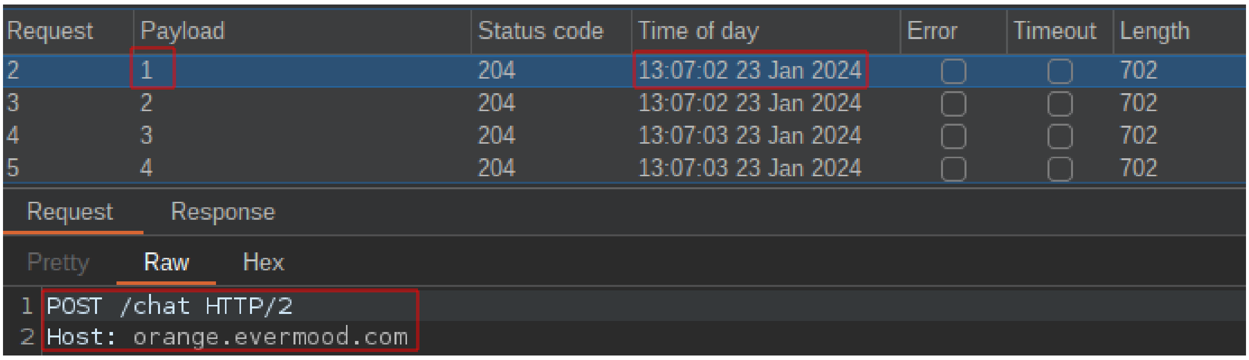
```
HTTP/2 204 No Content
Date: Fri, 01 Dec 2023 11:38:27 GMT
```

Figure 6 – The 1000th request was sent 31 seconds later at 11:38:26.

Retest Results

During the retest, this vulnerability was unresolved. It was possible to send more than 1000 requests to the application in a short period of time, specifically in 101 seconds. This resulted in a thousand messages appearing in the chat with the counsellor.

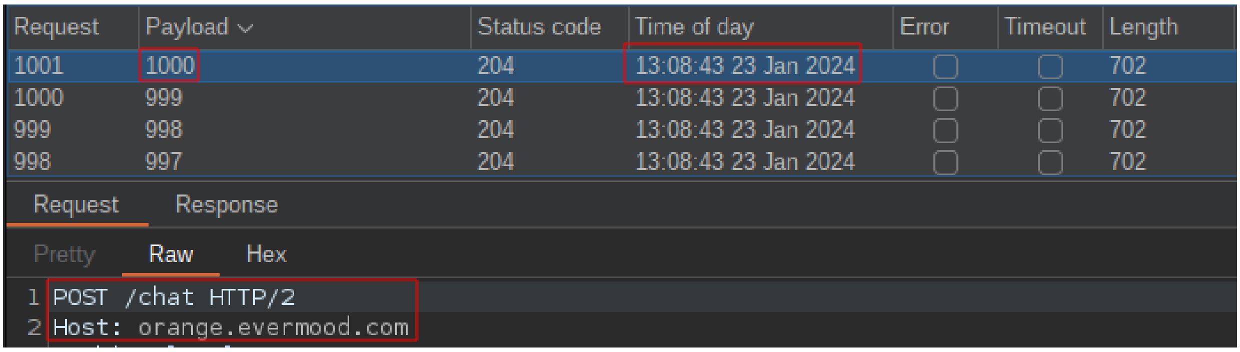
It is advised to implement rate limiting to the application to prevent an excessive number of requests and ultimately, the possibility of overwhelming the application and associated API to the point of causing a Denial-of-Service (DoS) condition.



Request	Payload	Status code	Time of day	Error	Timeout	Length
2	1	204	13:07:02 23 Jan 2024	□	□	702
3	2	204	13:07:02 23 Jan 2024	□	□	702
4	3	204	13:07:03 23 Jan 2024	□	□	702
5	4	204	13:07:03 23 Jan 2024	□	□	702

```
POST /chat HTTP/2
Host: orange.evermood.com
```

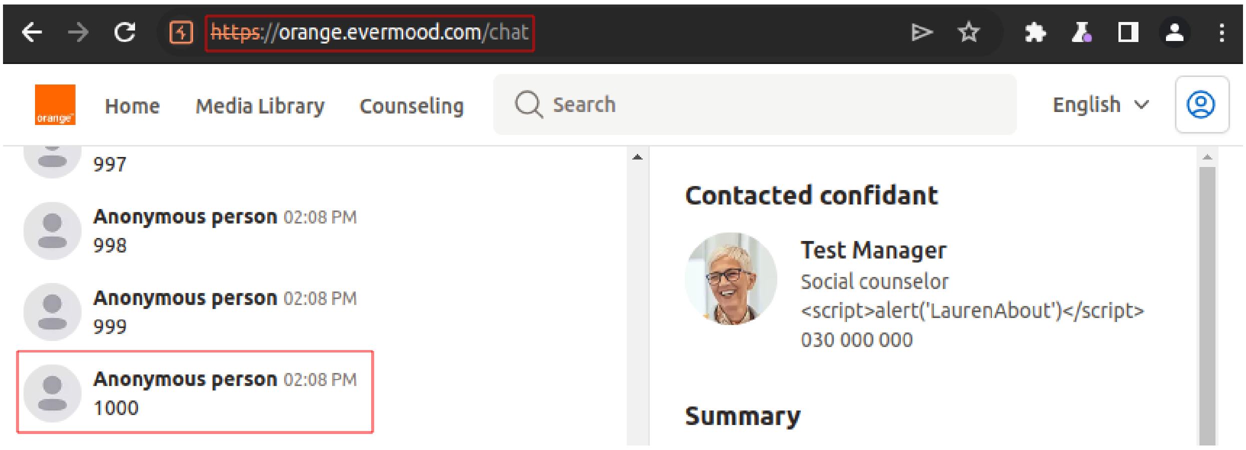
Figure 7 – The first request was sent at 13:07:02.



Request	Payload	Status code	Time of day	Error	Timeout	Length
1001	1000	204	13:08:43 23 Jan 2024	□	□	702
1000	999	204	13:08:43 23 Jan 2024	□	□	702
999	998	204	13:08:43 23 Jan 2024	□	□	702
998	997	204	13:08:43 23 Jan 2024	□	□	702

```
POST /chat HTTP/2
Host: orange.evermood.com
```

Figure 8 – The 1000th request was sent 101 seconds later at 13:08:43.



Anonymous person 02:08 PM
997

Anonymous person 02:08 PM
998

Anonymous person 02:08 PM
999

Anonymous person 02:08 PM
1000

Contacted confidant

Test Manager
Social counselor
<script>alert('LaurenAbout')</script>
030 000 000

Summary

Figure 9 - The chat showed the 1000th message sent at 02:08 PM.

R03	File Upload Vulnerability			
Risk Rating	Medium	CVSS3	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L:	4.3

Technical Overview

The application provided the ability to upload content and did not validate the file type being uploaded. As a result, any file type could be uploaded including executable content. An EICAR test file was uploaded to the Evermood AWS instance and verified, showing a user could upload this malicious content without the content being flagged by an anti-virus or other detection medium.

For more information regarding the EICAR files used during the assessment: https://www.eicar.org/download-anti-malware-testfile

Potential Impact if Exploited

Malicious files could be shared with users with the trust implicit in the application. If a user were to download a malicious file uploaded to the webserver, it could result in an attacker compromising the end user's machine.

Recommendations

Ensure strict validation is performed on all uploaded content. Content should be type checked as well as content checked, to ensure malicious uploads are not possible. Where possible, all files should be scanned by an anti-virus engine before being processed. Check that the files are not uploaded to a web-accessible area, and that the folder does not allow script execution. More detailed guidance can be found in OWASP's guide on the matter: https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

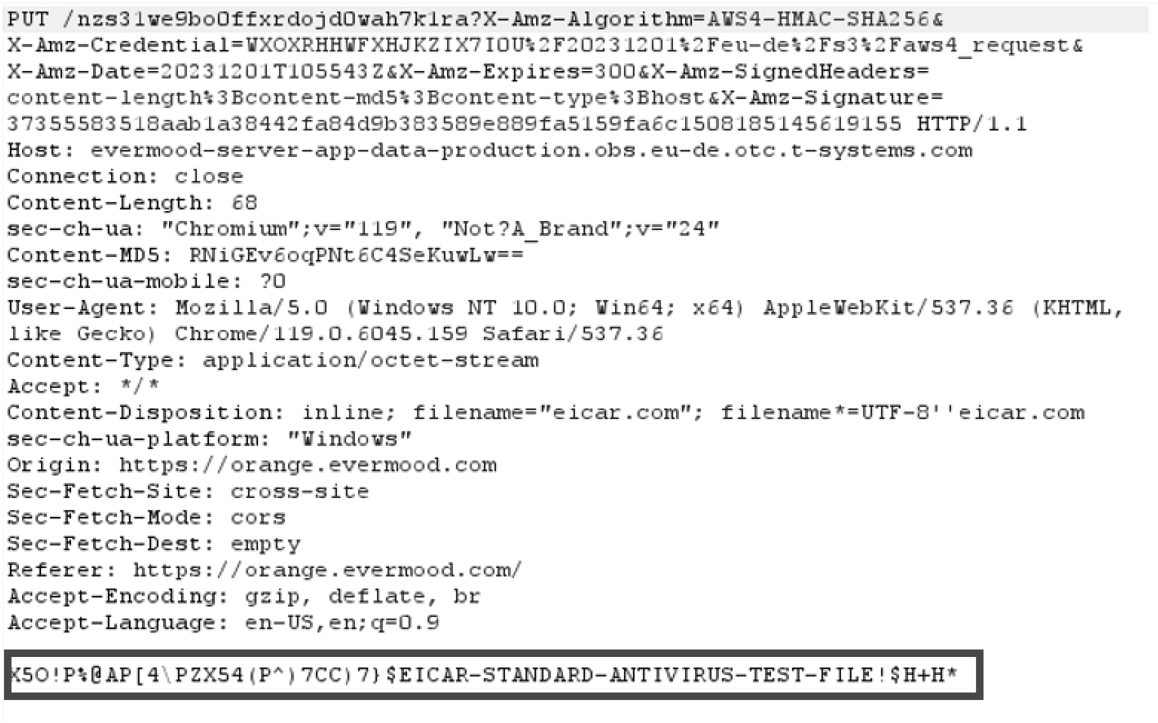
Instances

The image upload function in the following fields:

```
manager[about_me_text_en]
manager[faq_en]
manager[about_me_text_de]
manager[faq_de]
```

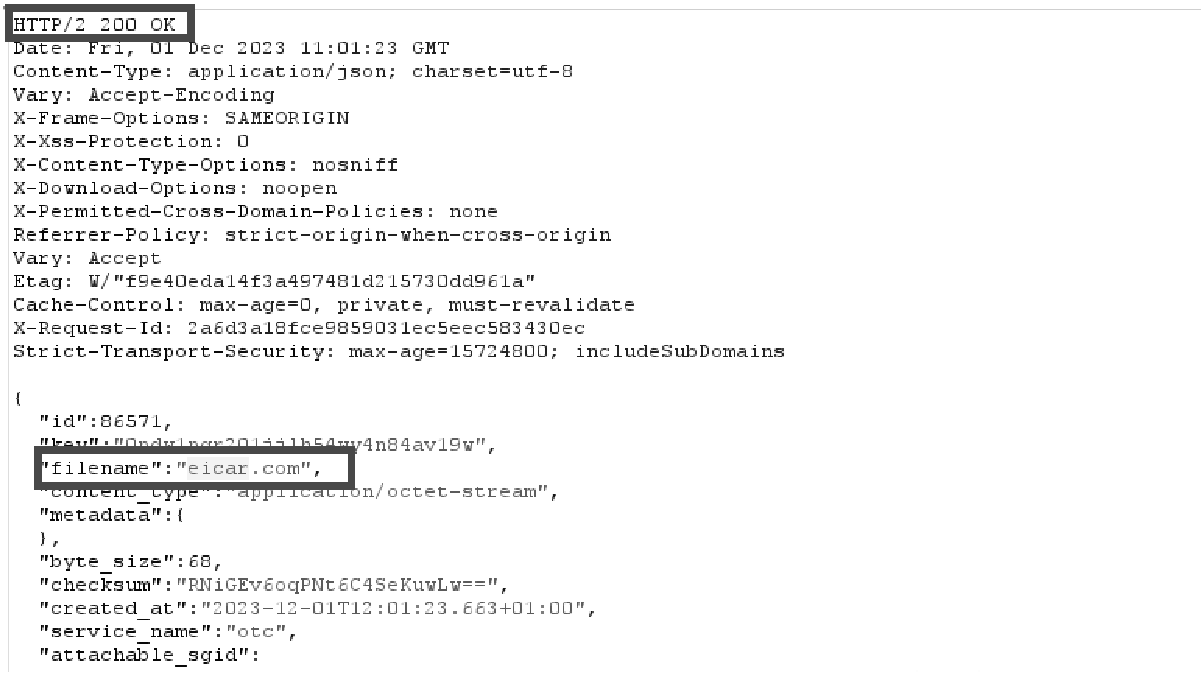
Attack Example

When using the upload image function on the About Me Text and FAQ fields on the counsellor profile page, no file type check took place allowing for any file type to be uploaded, this included a virus test file which was uploaded successfully.



```
PUT /nzs31we9bo0ffxrdojd0wah7k1ra?X-Amz-Algorithm=AWS4-HMAC-SHA256&
X-Amz-Credential=WXOXRHHWFXHJKZIX7IOU%2F20231201%2Feu-de%2Fs3%2Faws4_request&
X-Amz-Date=20231201T105543Z&X-Amz-Expires=300&X-Amz-SignedHeaders=
content-length%3Bcontent-md5%3Bcontent-type%3Bhost&X-Amz-Signature=
37355583518aab1a38442fa84d9b383589e889fa5159fa6c1508185145619155 HTTP/1.1
Host: evermood-server-app-data-production.obs.eu-de.otc.t-systems.com
Connection: close
Content-Length: 68
sec-ch-ua: "Chromium";v="119", "Not?A_Brand";v="24"
Content-MD5: RNiGEv6oqPNt6C4SeKuwLw==
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/119.0.6045.159 Safari/537.36
Content-Type: application/octet-stream
Accept: */*
Content-Disposition: inline; filename="eicar.com"; filename*=UTF-8''eicar.com
sec-ch-ua-platform: "Windows"
Origin: https://orange.evermood.com
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://orange.evermood.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

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

Figure 10 – Sample request submitting the standard EICAR file.

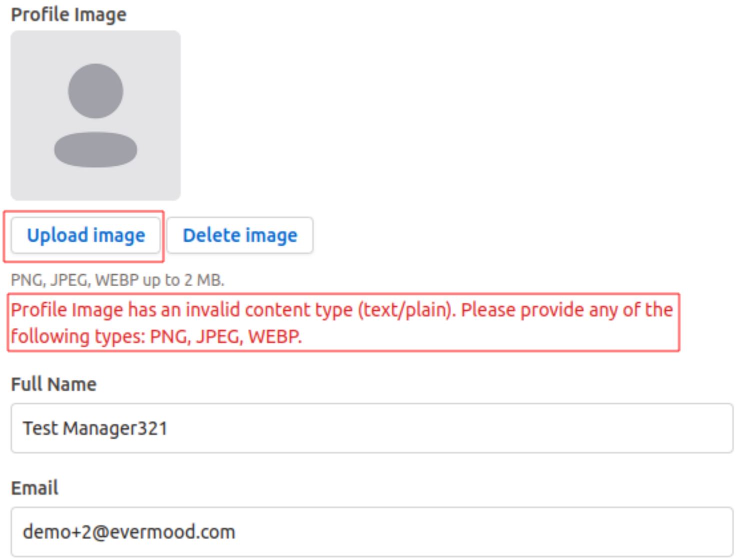


```
HTTP/2 200 OK
Date: Fri, 01 Dec 2023 11:01:23 GMT
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
Vary: Accept
Etag: W/"f9e40eda14f3a497481d215730dd961a"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 2a6d3a18fce985903 1ec5eec583430ec
Strict-Transport-Security: max-age=15724800; includeSubDomains

{
  "id":86571,
  "key":"0ndw1ngr201jj1h54wy4n84av19w",
  "filename":"eicar.com",
  "content_type":"application/octet-stream",
  "metadata":{
  },
  "byte_size":68,
  "checksum":"RNiGEv6oqPNt6C4SeKuwLw==",
  "created_at":"2023-12-01T12:01:23.663+01:00",
  "service_name":"otc",
  "attachable_sgid":
```

Figure 11 – Confirmation from the server the EICAR file uploaded successfully.

Retest Results

During the retest, this finding was found to be resolved. An attempt was made to upload a file type other than PNG, JPEG or WEBP, which was subsequently rejected by the application.



PNG, JPEG, WEBP up to 2 MB.

Profile Image has an invalid content type (text/plain). Please provide any of the following types: PNG, JPEG, WEBP.

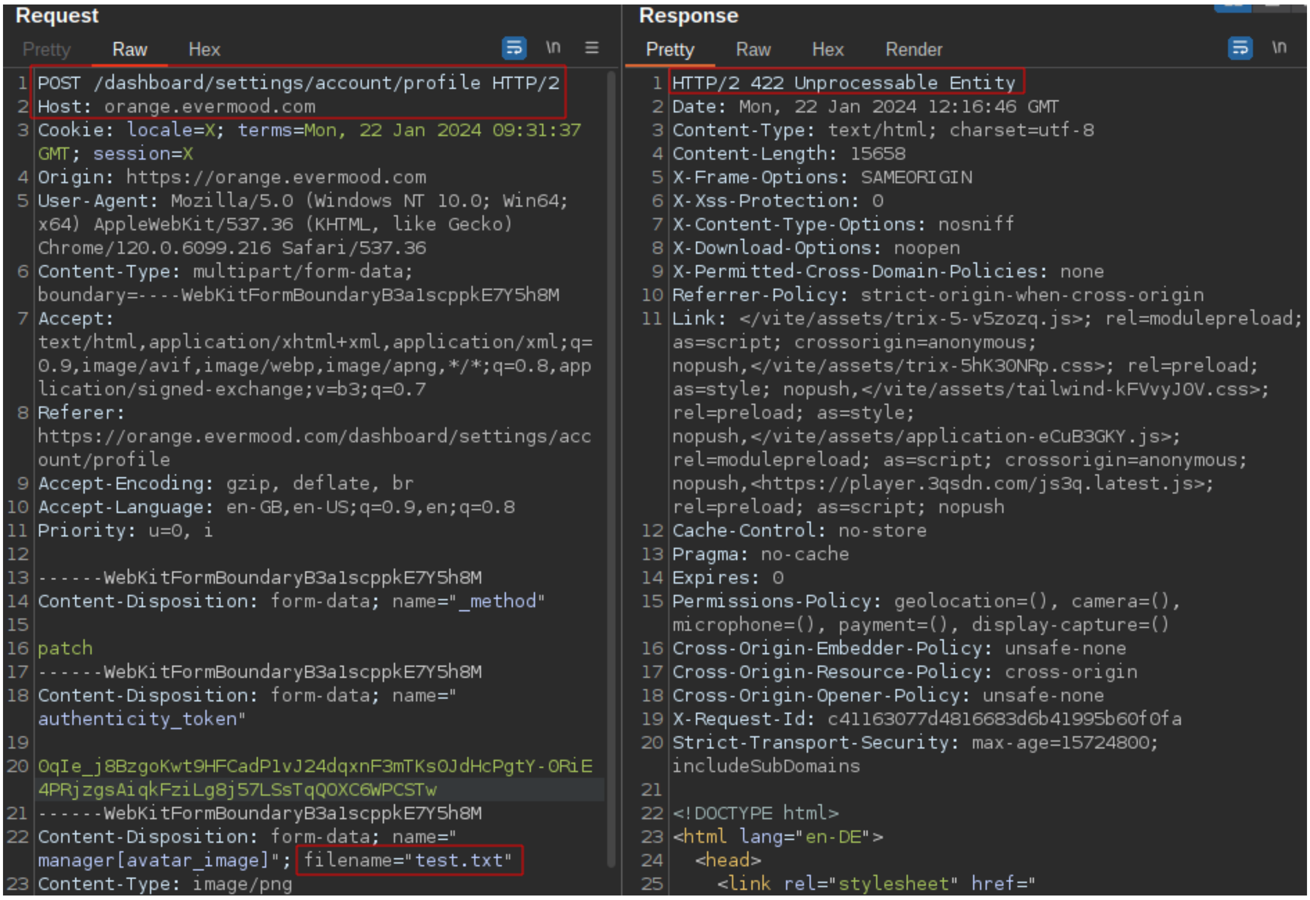
Full Name

Test Manager321

Email

demo+2@evermood.com

Figure 12 – During the retest, it was noted that client-side logic prevented file uploads other than PNG, JPEG or WEBP files to the Evermood application.



```
POST /dashboard/settings/account/profile HTTP/2
Host: orange.evermood.com
Cookie: locale=X; terms=Mon, 22 Jan 2024 09:31:37
GMT; session=X
Origin: https://orange.evermood.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.216 Safari/537.36
Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryB3a1scppkE7Y5h8M
Accept:
text/html,application/xhtml+xml,application/xml;q=
0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.7
Referer:
https://orange.evermood.com/dashboard/settings/acc
ount/profile
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=0, i

------WebKitFormBoundaryB3a1scppkE7Y5h8M
Content-Disposition: form-data; name="_method"

patch
------WebKitFormBoundaryB3a1scppkE7Y5h8M
Content-Disposition: form-data; name="
authenticity_token"

0qIe_j8BzgoKwt9HFCadP1vJ24dqxnF3mTKs0JdHcPgtY-0RiE
4PRjzgsAiqkFziLg8j57LSsTqQ0XC6WPCSTw
------WebKitFormBoundaryB3a1scppkE7Y5h8M
Content-Disposition: form-data; name="
manager[avatar_image]"; filename="test.txt"
Content-Type: image/png
```

```
HTTP/2 422 Unprocessable Entity
Date: Mon, 22 Jan 2024 12:16:46 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 15658
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
Link: </vite/assets/trix-5-v5zozq.js>; rel=modulepreload;
as=script; crossorigin=anonymous;
nopush,</vite/assets/trix-5hK30NRp.css>; rel=preload;
as=style; nopush,</vite/assets/tailwind-kFVvyJ0V.css>;
rel=preload; as=style;
nopush,</vite/assets/application-eCuB3GKY.js>;
rel=modulepreload; as=script; crossorigin=anonymous;
nopush,<https://player.3qsdn.com/js3q.latest.js>;
rel=preload; as=script; nopush
Cache-Control: no-store
Pragma: no-cache
Expires: 0
Permissions-Policy: geolocation=(), camera=(),
microphone=(), payment=(), display-capture=()
Cross-Origin-Embedder-Policy: unsafe-none
Cross-Origin-Resource-Policy: cross-origin
Cross-Origin-Opener-Policy: unsafe-none
X-Request-Id: c41163077d4816683d6b41995b60f0fa
Strict-Transport-Security: max-age=15724800;
includeSubDomains

<!DOCTYPE html>
<html lang="en-DE">
  <head>
    <link rel="stylesheet" href="
```

Figure 13 – An attempt to upload a text file (.txt) was made to the Evermood application and rightfully rejected by the server component.

R04	Input Reflected as HTML in Response			
Risk Rating	Medium	CVSS3	CVSS3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N	4.3

Technical Overview

The application exhibited traces of partially unsanitised user input in its response. While this situation typically raises concerns about reflective Cross-Site Scripting, the security measures implemented by the application, such as sanitising the <> symbols at the point of reflection, effectively prevented any potential injection attack.

Potential Impact if Exploited

In the context of web applications, failure to adequately sanitise input can lead to vulnerabilities like Cross-Site Scripting (XSS) or SQL injection. XSS occurs when untrusted data from users is included in web pages without proper validation, enabling attackers to execute malicious scripts in the browsers of other users.

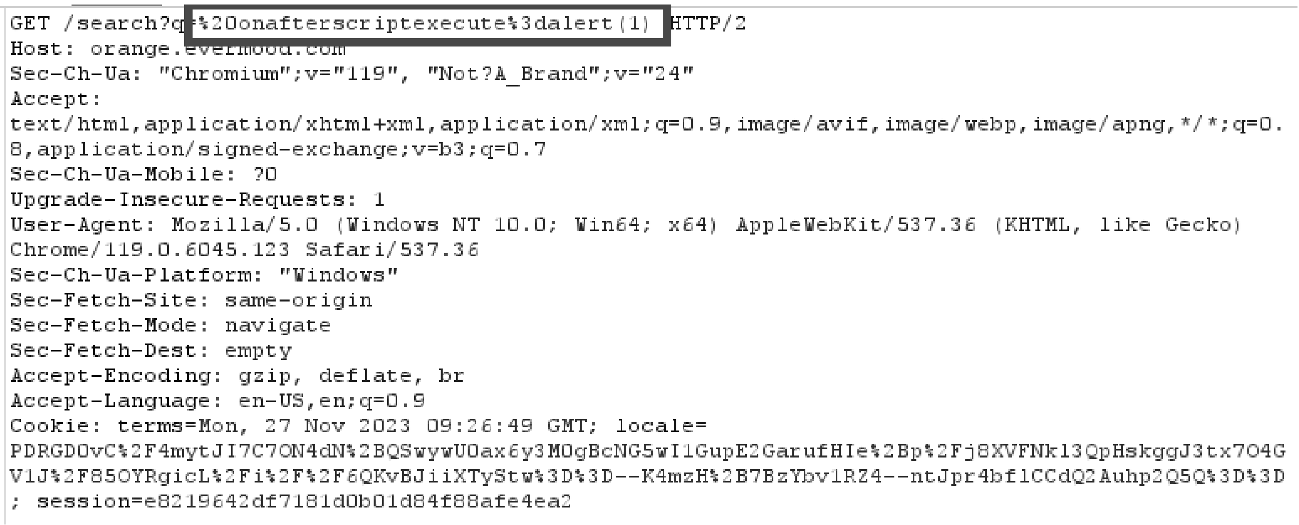
Recommendations

To prevent XSS and SQL attacks from occurring, complete input validation and output encoding should be performed on all user-supplied input. Escaping HTML is fairly easy. However, to properly protect the application from all attacks it is required to escape JavaScript, Cascading Style Sheets, and XML data.

Instances

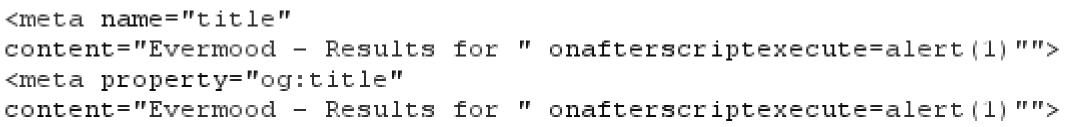
GET /search [q parameter]

Attack Example



```
GET /search?q=%20onafterscriptexecute%3dalert(1) HTTP/2
Host: orange.evermood.com
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.
8,application/signed-exchange;v=b3;q=0.7
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.123 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: empty
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: terms=Mon, 27 Nov 2023 09:26:49 GMT; locale=
PDRGD0vC%2F4mytJI7C7ON4dN%2BQSwywU0ax6y3MOgBcNG5wI1GupE2GarufHIe%2Bp%2Fj8XVFNkl3QpHskggJ3tx704G
V1J%2F850YRgicL%2Fi%2F%2F6QKvBJiiXTyStw%3D%3D--K4mzH%2B7BzYbv1RZ4--ntJpr4bf1CCdQ2Auhp2Q5Q%3D%3D
; session=e8219642df7181d0b01d84f88afe4ea2
```

Figure 14 – Injected JavaScript code in the q parameters.

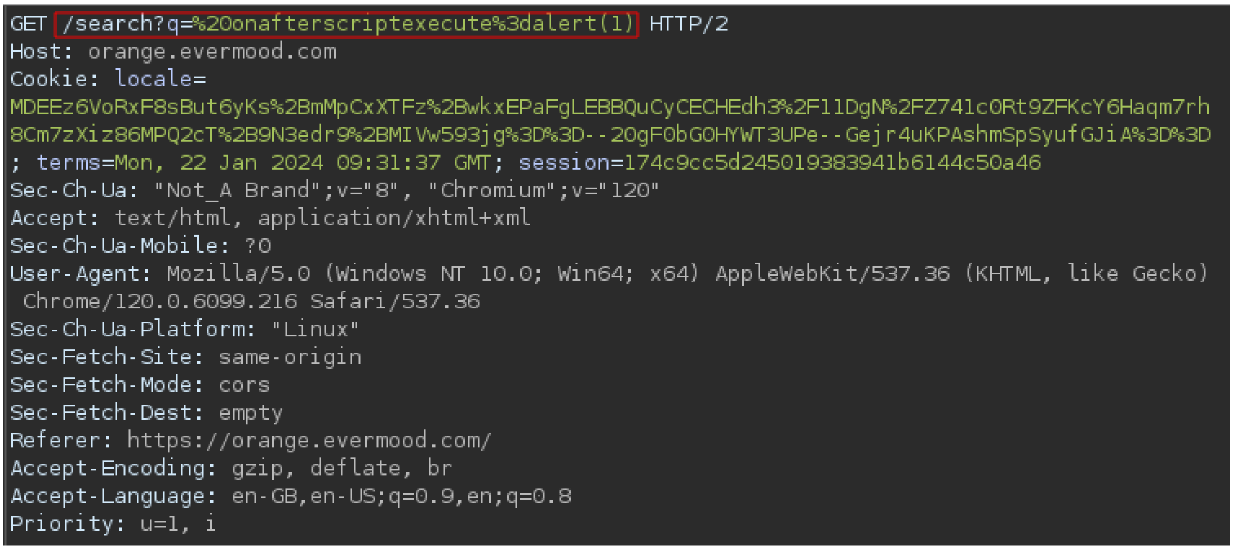


```
<meta name="title"
content="Evermood - Results for " onafterscriptexecute=alert(1)"">
<meta property="og:title"
content="Evermood - Results for " onafterscriptexecute=alert(1)"">
```

Figure 15 – The q input reflected in the response as valid HTML.

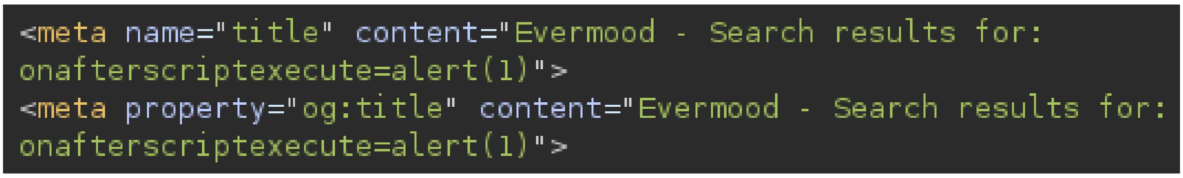
Retest Results

During the retest, this vulnerability was found to be resolved. Attempts to manipulate the string to identify new exploitation vectors by adding new attributes were not successful, as these were sanitised by the application.



```
GET /search?q=%20onafterscriptexecute%3dalert(1) HTTP/2
Host: orange.evermood.com
Cookie: locale=
MDEEz6VoRxF8sBut6yKs%2BmMpCxXTFz%2BwkxEPaFgLEBBQuCyCECHEdh3%2F11DgN%2FZ741c0Rt9ZFKcY6Haqm7rh
8Cm7zXiz86MPQ2cT%2B9N3edr9%2BMIVw593jg%3D%3D--20gF0bG0HYWT3UPe--Gejr4uKPAshmSpSyufGJiA%3D%3D
; terms=Mon, 22 Jan 2024 09:31:37 GMT; session=174c9cc5d245019383941b6144c50a46
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: text/html, application/xhtml+xml
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
 Chrome/120.0.6099.216 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://orange.evermood.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=1, i
```

Figure 16 – JavaScript logic was injected into the ‘q’ parameter.



```
<meta name="title" content="Evermood - Search results for:
onafterscriptexecute=alert(1)">
<meta property="og:title" content="Evermood - Search results for:
onafterscriptexecute=alert(1)">
```

Figure 17 – The response showed that the application was sanitising input, therefore removed any client-side logic from the output.

R05	Missing HTTP Security Headers			
Risk Rating	Low	CVSS3	CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N	3.7

Technical Overview

Some HTTP security headers were missing. HTTP security headers are a method of improving the default level of security offered by a web browser. For this reason, the lack of security headers does not constitute a vulnerability itself but is considered as a lack of hardening.

Header	Summary
Permissions-Policy	Provides a mechanism to allow and deny the use of browser features in a document or within any <iframe> elements in the document.
Cross-Origin-Embedder-Policy	Configures embedding cross-origin resources into the document.
Cross-Origin Resource Policy (CORP)	Policy set by the Cross-Origin-Resource-Policy HTTP header that lets websites and applications opt into protection against certain requests from other origins.
Cross-Origin-Opener-Policy	Ensure a top-level document does not share a browsing context group with cross-origin documents.

Potential Impact if Exploited

While missing security headers are not a vulnerability, they weaken the overall security posture of the application. Introducing them will help minimise the impact of vulnerabilities such as Cross-Site Scripting (XSS), Clickjacking and Person-in-the-Middle (PitM) attacks.

Recommendations

Consider implementing the following response headers.

Cross-Origin-Opener-Policy (COOP):
Adding COOP comes at the cost of breaking use of the Window.openerAPI, when used between origins.
Different origins unsafely share the same browsing context group. (The current default behaviour of browsers.)

```
unsafe-none
```

The origin setting the header gets its own browsing context group. (The fully secure mode.)

```
same-origin
```

Allowing pop-ups that originate from the same origin but blocks other content. (A middle ground.)

```
same-origin-allow-popups
```

Note: Adding this header might cause issues with some functions that the applications might have in place. Therefore, it must be carefully assessed in a development environment to assess the impact.

Cross-Origin-Resource-Policy (CORP):
This header allows you to manage other sites from loading your resources into their browsing context groups, giving them potential access to your private data. Other origins can embed your resources. (The current default behaviour of browsers.)

```
cross-origin
```

Resources are not accessible outside this origin. (The most secure option.)

```
same-origin
```

For site using subdomains.

```
same-site
```

Cross-Origin-Embedder-Policy (COEP):
You allow your site to embed any other sites' resources. (The current default behaviour of browsers.)

```
unsafe-none
```

Any cross-origin resources you load must explicitly allow you to load them. (The secure mode.)

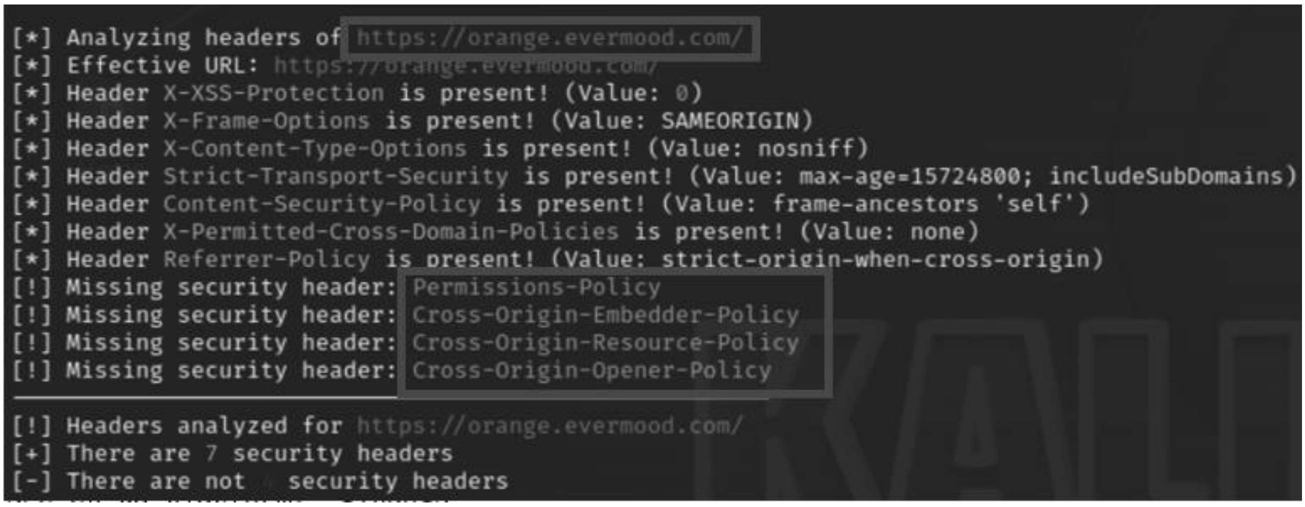
```
require-corp
```

More information about the configuration of security header can be found at the following:
https://adamj.eu/tech/2021/05/01/how-to-set-coep-coop-corp-security-headers-in-django/

Instances

https://orange.evermood.com/chat

Attack Example



```
[*] Analyzing headers of https://orange.evermood.com/
[*] Effective URL: https://orange.evermood.com/
[*] Header X-XSS-Protection is present! (Value: 0)
[*] Header X-Frame-Options is present! (Value: SAMEORIGIN)
[*] Header X-Content-Type-Options is present! (Value: nosniff)
[*] Header Strict-Transport-Security is present! (Value: max-age=15724800; includeSubDomains)
[*] Header Content-Security-Policy is present! (Value: frame-ancestors 'self')
[*] Header X-Permitted-Cross-Domain-Policies is present! (Value: none)
[*] Header Referrer-Policy is present! (Value: strict-origin-when-cross-origin)
[!] Missing security header: Permissions-Policy
[!] Missing security header: Cross-Origin-Embedder-Policy
[!] Missing security header: Cross-Origin-Resource-Policy
[!] Missing security header: Cross-Origin-Opener-Policy
-------------------------------------------------------
[!] Headers analyzed for https://orange.evermood.com/
[+] There are 7 security headers
[-] There are not 4 security headers
```

Figure 18 – Missing Security Headers found on https://orange.evermood.com using the shcheck.py[3] tool.

[3] https://github.com/santoru/shcheck

Retest Results

During the retest, this finding was shown to be resolved. It was found that all missing security headers had been implemented, including the Permissions-Policy header.

```
➜ shcheck git:(master) ./shcheck.py https://orange.evermood.com/
> shcheck.py - santoru .............................
Simple tool to check security headers on a webserver
[*] Analyzing headers of https://orange.evermood.com/
[*] Effective URL: https://orange.evermood.com/
[*] Header X-XSS-Protection is present! (Value: 0)
[*] Header X-Frame-Options is present! (Value: SAMEORIGIN)
[*] Header X-Content-Type-Options is present! (Value: nosniff)
[*] Header Strict-Transport-Security is present! (Value: max-age=15724800;
includeSubDomains)
[*] Header Content-Security-Policy is present! (Value: frame-ancestors 'self')
[*] Header X-Permitted-Cross-Domain-Policies is present! (Value: none)
[*] Header Referrer-Policy is present! (Value: strict-origin-when-cross-origin)
[*] Header Permissions-Policy is present! (Value: geolocation=(), camera=(),
microphone=(), payment=(), display-capture=())
[*] Header Cross-Origin-Embedder-Policy is present! (Value: unsafe-none)
[*] Header Cross-Origin-Resource-Policy is present! (Value: cross-origin)
[*] Header Cross-Origin-Opener-Policy is present! (Value: unsafe-none)
-------------------------------------------------------
[!] Headers analyzed for https://orange.evermood.com/
[+] There are 11 security headers
[-] There are not 0 security headers
```

Figure 19 – The shcheck.py was used and verified that all missing security-related headers were now present.

Appendix A. Clean-Up Operations

WARNING: The changes listed below can present a significant risk to the systems reviewed and should be removed as soon as possible.

Orange Cyberdefense accepts no liability for damage if the necessary steps are not taken to remove the items listed below.

Accounts	demo+1@evermood.com demo+2@evermood.com

Table 8 – Clean-Up Operations.

Appendix B. Risk Rating System

Appendix B.1. CVSS3: An Open Standard for Vulnerability Scoring

The Common Vulnerability Scoring System (version 3) is an established method for scoring technical vulnerabilities identified in systems.

The CVSS3 is based on three metric groups:

- **Base Metric Group**: “represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.” It covers metrics relating to the complexity (proximity of attacker, authentication requirements) of the attack and its impact on the security qualities of the system (confidentiality, integrity, and availability).
- **Temporal Metric Group**: “represents the characteristics of a vulnerability that change over time but not among user environments.” It covers metrics relating to the current state of the vulnerability (exploitability and remediation options) and to the confidence of the issue at hand.
- **Environmental Metric Group**: “represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment”. These metrics allow Evermood to ensure that the controls in place are factored into the assessment of the vulnerability’s actual relationship with the environment, leading to a more accurate representation of the technical risk.

During an assessment, only the base metric group is calculated for each vulnerability. By request, and provided with additional information, the temporal and environmental metric groups can be calculated.

For further information on the CVSS3 system, see the following reference site:

http://www.first.org/cvss/user-guide

Appendix B.2. Qualitative Severity Rating Scale (QSR)

The Qualitative Severity Rating (QSR) used by Orange Cyberdefense follows the CVSS3 guidelines. It allows for a textual representation of the CVSS3 scores and provides an intuitive means of communicating an understanding of the risk to non-technical stakeholders. The model is a simple ranking of issues from Low to Critical, in descending order of severity.

The following table provides an explanation of each level.

QSR	Description
Critical	Successful attacks within this category could result in an attacker gaining access to view, modify or destroy highly confidential information; conduct or falsify large numbers of unauthorised financially sensitive operations (e.g., falsification of financial transactions, deletion of data records), or lead to a complete compromise of the target. Such attacks could have a catastrophic impact on the confidentiality, integrity and availability of the systems and the business. This could result in a significant financial loss, significant reputational damage, serious legal and compliance-related fines, and other effects on the business. An immediate remediation plan should be developed to address issues rated at this level.
High	Successful attacks within this category could result in an attacker gaining access to view, modify or destroy confidential information; conduct or falsify unauthorised financially sensitive operations (e.g., falsification of financial transactions, deletion of data records), or lead to significant compromise of the target. Such attacks could have a significant impact on the confidentiality, integrity and availability of the systems and the business. This could result in a significant financial loss, significant reputational damage, serious legal and compliance-related fines, and other effects on the business. An immediate remediation plan should be developed to address issues rated at this level.
Medium	A Medium QSR could lead to a noticeable impact on the business. Successful attacks within this category could allow an attacker to gain access to sensitive information or to private (personal) records, or could cause the system to perform unauthorised, but non-business critical, operations, or could lead to a significant outage of services. Such attacks could have a noticeable impact on the confidentiality, integrity and availability of the systems and the business, which could result in a noticeable financial loss, considerable reputational damage, legal and compliance-related fines, and other effects on the business. A timely remediation plan should be developed to address issues rated at this level. However, business requirements may dictate that other actions are more appropriate.
Low	A Low QSR is unlikely to have a noticeable impact on the business. However, such issues do not exist in isolation and may be used by an attacker as part of more complicated, blended attack, and should not be dismissed. Issues should be considered both individually and collectively. Issues identified at this level should be addressed as part of normal improvement exercises. However, business requirements may dictate that other actions are more appropriate.

Table 9 – Qualitative Severity Rating.

Appendix B.3. Mapping CVSS3 to the Qualitative Severity Rating Scale

The following table provides a mapping of CVSS3 metric scores to each QSV and follows the CVSS3 guidelines:

QSR	CVSS3 Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9

Appendix B.4. Your Risk Methodology

The QSR's and CVSS3 ratings provided in this report do not constitute a complete business risk assessment. Orange Cyberdefense analysts rarely have sufficient information to conduct a company-specific risk assessment. This would require more information than is typically available for such projects, such as knowledge of Evermood's risk appetite.

Orange Cyberdefense recommends that the information communicated via QSV's and CVSS3 metrics be used as input into the business risk methodology. However, where possible, Orange Cyberdefense analysts can assist in the assessment of the identified risks and how they should be interpreted by the business should this be required.

Appendix C. Methodologies

Orange Cyberdefense follows several methodologies when conducting security assessments. These methodologies are based on our extensive assessment experience and include a large amount of information.

To keep the length of this report to a manageable level all the current methodologies used by Orange Cyberdefense analysts can be viewed at https://sensepost.com/assessments/methodologies.

Appendix D. Document Management

Appendix D.1. Document Information

Project Title	Web Application Retest Assessment for Evermood
Project Code	SP80363 (Retest of SP77311)
Project Type	Web Application Retest Assessment
Report Date	2024-01-29
Document Name	SP80363_Orange_Cyberdefense_Security_Retest_Assessment_Report_for_Evermood
Author (Original)	James Hill
Author (Retest)	Lauren Skinner

Appendix D.2. Change Management

Ver.	Date	Author	Summary
0.1	2023-12-01	James Hill	Initial Draft.
0.2	2023-12-04	Jonathan Wagener	Technical Peer Review.
0.3	2023-12-08	Ulrich Swart	Quality Assurance.
1.0	2023-12-08	Bernice Kotze	Version 1.0 Released to Evermood.
1.1	2024-01-23	Lauren Skinner	Retest Report Draft.
1.2	2024-01-24	Justin Grobler	Technical Peer Review.
1.3	2024-01-26	Jacques Coertze	Quality Assurance.
2.0	2024-01-30	Bernice Kotze	Version 2.0 Released to Evermood.