Radically Open Security

Web Application Penetration Test Report

Open Technology Fund

V 1.0
Amsterdam, September 23rd, 2023
Confidential

Document Properties

Client	Open Technology Fund
Title	Web Application Penetration Test Report
Targets	Hypha web application (subdomain, particularly the project components) OTF Beta website (Wordpress)
Version	1.0
Pentester	Stefan Vink
Authors	Stefan Vink, Abhinav Mishra
Reviewed by	Abhinav Mishra
Approved by	Melanie Rieback

Version control

Version	Date	Author	Description
0.1	September 13th, 2023	Stefan Vink	Initial draft
0.2	September 14th, 2023	Stefan Vink	Ready-to-Review
1.0	September 23rd, 2023	Abhinav Mishra	Final Report

Contact

For more information about this document and its contents please contact Radically Open Security B.V.

Name	Melanie Rieback
Address	Science Park 608 1098 XH Amsterdam The Netherlands
Phone	+31 (0)20 2621 255
Email	info@radicallyopensecurity.com

Radically Open Security B.V. is registered at the trade register of the Dutch chamber of commerce under number 60628081.

Table of Contents

1 Executive Summary

1.1 Introduction

Between July 21, 2023 and September 13, 2023, Radically Open Security B.V. carried out a penetration test for Open Technology Fund

This report contains our findings as well as detailed explanations of exactly how ROS performed the penetration test.

1.2 Scope of work

The scope of the penetration test was limited to the following target:

- Hypha web application (subdomain, particularly the project components)
- OTF Beta website (Wordpress)

The scoped services are broken down as follows:

- **Total effort: 11.5 days**

1.3 Project objectives

ROS will perform a penetration test of the Hypha platform and the beta Open Tech Fund website with OTF in order to assess the security of this. To do so ROS will access the internal network and guide OTF in attempting to find vulnerabilities, exploiting any such found to try and gain further access and elevated privileges.

1.4 Timeline

The security audit took place between July 21, 2023 and September 13, 2023.

1.5 Results In A Nutshell

During this crystal-box penetration test we found 1 High, 4 Moderate and 19 Low-severity issues.

The High severity issue OTF3-022 (page 15) would allow an authenticated low privileged user or higher, such as partners, to see the comments of others. This could result in exposure of private information which when found could have a high impact on the confidentiality of the application. The other issues found in the Hypha application were Moderate and Low severity issues due to insecure session management (sessions remained valid for 14 days), improper input validation that could result in Cross site scripting, cross domain inclusion of a Google Translate script, missing

CSP header, backup 2FA tokens not properly protected, insufficient anti automation, user enumeration and several other misconfigurations.

Pentesting the new Wordpress Hypha frontend part resulted in discovery of Moderate and Low severity issues. The WP Cron feature was found to be active, config files have hardcoded SMTP credentials set, file editing is permitted through the UI, 2FA is not enabled for the admin user, lack of CSP header, ACL hardening can be improved, "Post via Email" is set up with standard server information, Wordpress version can easily be found, management interface is exposed, user enumeration can be performed via wp-json and there is public access to the development and test websites while a password is required for the beta website.

Note that Moderate and Low severity issues did not have a major immediate risk but when resolved would make it harder for adversaries to succeed to launch attacks against the application, infrastructure and users.

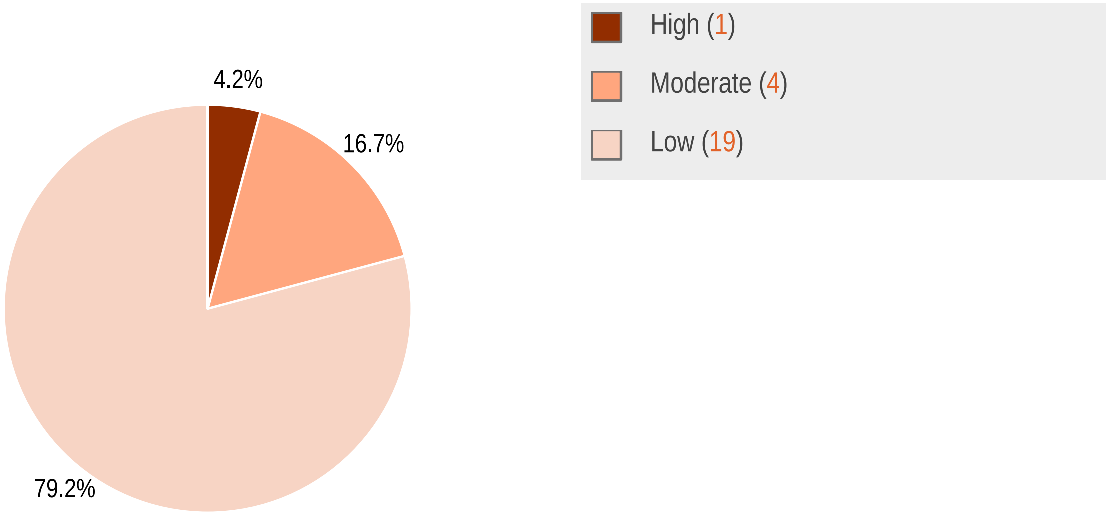
At the end of this pentest we did conduct a retest as well. It was found that the High issue and many other issues were resolved which is a good outcome. The finding status has been updated to reflect this.

1.6 Summary of Findings

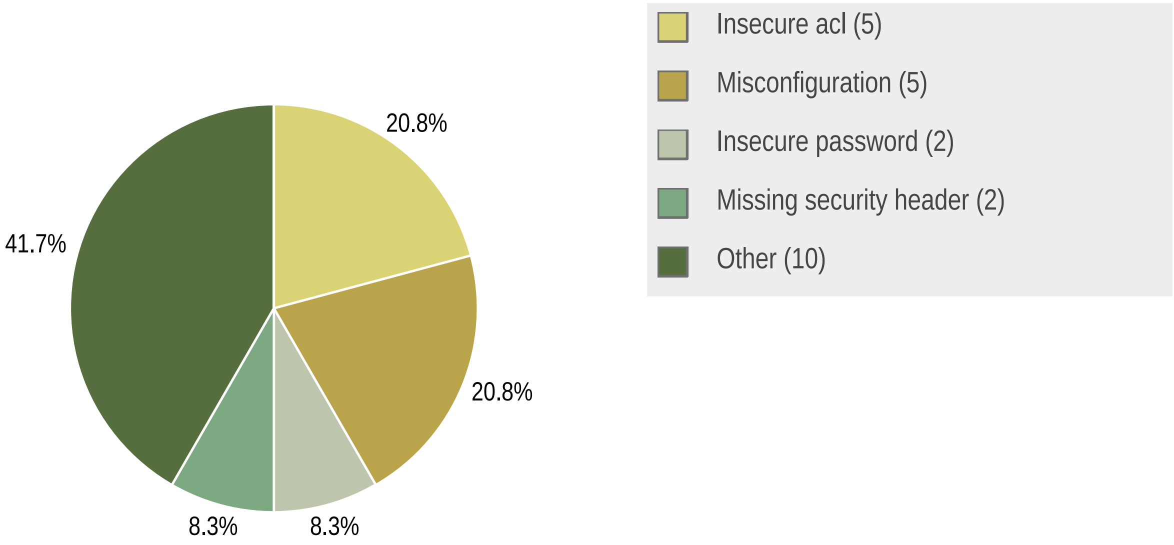
ID	Type	Description	Threat level
OTF3-022	Insecure ACL	The API Comments functionality does not have the proper ACL configured allowing other users such as applicants and partners to see the comments of others.	High
OTF3-021	Improper Session Management	Sessions remain active after a user closes the application and remain valid for 14 days.	Moderate
OTF3-016	Improper Input Validation	The application incorrectly validates input that can affect the control flow or data flow of a program.	Moderate
OTF3-008	Insecure Password	The admin account is not protected by Two-Factor-Authentication (2FA).	Moderate
OTF3-005	Misconfiguration	WordPress user details can be found through the wp-json endpoint.	Moderate
OTF3-028	XSS	Improper input validation in Wagtail CMS leads to multiple XSS vulnerabilities across the frontend and backend.	Low
OTF3-027	Cross Domain Inclusion	An external inclusion of the Google Translate script was found in the response of requests.	Low
OTF3-026	Insecure ACL	During the pentest we discovred that the Wagtail Documents can be downloaded without authentication.	Low
OTF3-024	Missing Security Header	The Hypha application does not use the Content Security Policy header.	Low
OTF3-023	Insecure Password	Backup Tokens shown during 2FA setup stay the same and can be accessed by the user even after setup is complete.	Low
OTF3-020	Misconfiguration	External images can be embedded within several forms throughout the application.	Low

OTF3-018	Missing Anti-Automation	The Apply form does not have proper Anti Automation functionality implemented.	Low
OTF3-017	Insecure ACL	Low privileged users are able to Purge CDN and Cache.	Low
OTF3-015	User Enumeration	Valid users can be found by abusing the Profile Change Email address functionality.	Low
OTF3-014	TLS Misconfiguration	The webserver allows obsolete Cipher Block Chaining (CBC) encryption.	Low
OTF3-013	Misconfiguration	The external WP-Cron appears to be enabled.	Low
OTF3-012	Hardcoded Credentials	The SMTP configuration including credentials was found in the supplied source code.	Low
OTF3-010	Misconfiguration	Wordpress backend files such as configuration and templates can be edited using the UI.	Low
OTF3-007	Misconfiguration	Default settings were found in the Post via Email functionality.	Low
OTF3-006	Information Leak	The Wordpress version is exposed.	Low
OTF3-004	Exposed Management Interface	Access to the websites Wordpress management portal can be easily guessed which would allow public access to the management interface login portal.	Low
OTF3-003	Missing Security Header	The application fails to set an appropriate Content-Security Policy header on some pages which could allow attacks such as XSS and Clickjacking.	Low
OTF3-002	Insecure ACL	The Development and Test websites lack password protection allowing unauthorized access.	Low
OTF3-001	Insecure ACL	Several directories and files are not properly hardened.	Low

1.6.1 Findings by Threat Level



1.6.2 Findings by Type



1.7 Summary of Recommendations

ID	Type	Recommendation
OTF3-022	Insecure ACL	Update the code to make sure that low privileged users such as applicants and partners cannot read other clients comments.
OTF3-021	Improper Session Management	• Reduce the default max session time to a lower value, e.g. 8 hours. Also make it possible for administrators to configure this value in the configuration file of the application. • Set a maximum amount of active sessions per user. If possible only allow one active session per user. Note that some applications require the use of multiple devices (as they can be used on other devices such as a mobile phone and laptop), which discounts allowing only one active session as a solution. In this case the user could be shown a session overview (with the time when session was created and the kind of device it was created on) in case there are multiple active sessions. The user should be able to select and invalidate any session that should be deactivated.
OTF3-016	Improper Input Validation	• Assume all input is malicious. Use an 'accept known good' input validation strategy i.e. use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. • When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. • Do not rely exclusively on looking for malicious or malformed inputs (i.e. do not rely on a blacklist). A blacklist is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. • For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then these modified values would be submitted to the server. • Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings. • When your application combines data from multiple sources, perform the validation after the sources have been combined. The individual data elements may pass the validation step but violate the intended restrictions after they have been combined. Inputs should be decoded

		and canonicalised to the application's current internal representation before being validated. • Make sure that your application does not inadvertently decode the same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked. • Consider performing repeated canonicalisation until your input does not change any more. This will avoid double-decoding and similar scenarios, but it might inadvertently modify inputs that are allowed to contain properly-encoded dangerous content.
OTF3-008	Insecure Password	Add support for 2FA authentication.
OTF3-005	Misconfiguration	The WordPress installation has the REST API `/wp-json/wp/v2/`publicly accessible. It is recommend to disallow the public access, or disable the API if it is not needed.
OTF3-028	XSS	Implement proper input validation and output encoding to prevent XSS attacks. More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting
OTF3-027	Cross Domain Inclusion	Try to avoid (where possible) using cross-domain scripts. Include and host as many scripts as possible on your own domain(s. This way, you keep control over the source code, as well as referrer information.
OTF3-026	Insecure ACL	Add authentication checks before downloading the document is allowed.
OTF3-024	Missing Security Header	Note that X-XSS-Protection and X-Frame-Options headers are already set in the response which helps to mitigate these kind of attacks. However, best practice suggests to replace these headers for a CSP Header that blocks these attacks.
OTF3-023	Insecure Password	Make sure to follow the standard practice where backup tokens are only shown to the user once, typically at the time of setting up 2FA. Implementing proper access controls and logging around the viewing and usage of backup tokens can further enhance the security and integrity of the authentication process.
OTF3-020	Misconfiguration	Disable external image uploads if possible. If images are needed, use the application's existing internal upload feature.
OTF3-018	Missing Anti-Automation	Apply an anti-automation on forms. One of the common ways to do it would be implementing a Captcha (hCAPTCHA is very effective) on those pages and only show and enforce the use of it after a certain amount of requests per IP. Note that Cloudflare does have some protection against this but still it did allow us to make more requests then should be preferred.
OTF3-017	Insecure ACL	Verify whether the current user is allowed to access the requested resource and deny access if this is not the case.
OTF3-015	User Enumeration	• Set a timeout of 1 second (so based on the respones time an attacker cannot conduct a timing attack to find out whether some processing happens or not if an account exists or not) • Mention in the message to the user that a confirmation mail is send to the email address set. If the address already exists in the system an email won't be send. If it does not exist an email would be send. And only after confirming the email the address is updated in the email field.

OTF3-014	TLS Misconfiguration	Disable the use of TLS CBC ciphers. De-prioritizing these ciphers can also help minimize successful exploitation of real-world attacks. The attacker typically cannot force the selection of a specific cipher and therefore can only execute a CBC padding oracle attack if the client/server normally negotiates a vulnerable cipher.
OTF3-013	Misconfiguration	Add the variable DISABLE_WP_CRON to true in the file wp-config.php and restrict access to the file wp-cron.php. In case the cron is used there are other ways to run a cronjob. The alternative is to create in the system a cronjob that executes the wp-cron.php script directly through PHP every minute and avoid Http requests. There are also plugins that can assist with this.
OTF3-012	Hardcoded Credentials	• Do not store plaintext credentials. • Remove hardcoded configuration files from source repositories. • Set a strong password, unique for each environment.
OTF3-010	Misconfiguration	Disable changing theme and plugin files within the GUI as an administrator in Wordpress by adding the following in wp-config.php: `define('DISALLOW_FILE_EDIT', true);`
OTF3-007	Misconfiguration	Change the email settings to that of your own domain. Even if no mailserver exist.
OTF3-006	Information Leak	Remove the information that exposes version information. Wordpress has several plugins and code snippets that can assist with this.
OTF3-004	Exposed Management Interface	Restrict access by consider implementing IP-whitelisting or using Cloudflare Access to reduce the attack vector.
OTF3-003	Missing Security Header	Include a strict CSP header in every response.
OTF3-002	Insecure ACL	Require credentials before access is allowed to Test and Development websites.
OTF3-001	Insecure ACL	• Restrict Admin access by adding restrictions who can reach the the wp-admin directory. Examples how to do this are using basic authentication or Cloudflare Access. • Disable WP-CRON (note this sometimes breaks some plugins) • Disable executing PHP files by web-users in restrictive directories. • Add additional hardening to protect against potential arbitrary file upload • Several Wordpress plugins can assist with restricting access and additionally using other best security practices.

2 Methodology

2.1 Planning

Our general approach during penetration tests is as follows:

1. **Reconnaissance**
   We attempt to gather as much information as possible about the target. Reconnaissance can take two forms: active and passive. A passive attack is always the best starting point as this would normally defeat intrusion detection systems and other forms of protection afforded to the app or network. This usually involves trying to discover publicly available information by visiting websites, newsgroups, etc. An active form would be more intrusive, could possibly show up in audit logs and might take the form of a social engineering type of attack.
2. **Enumeration**
   We use various fingerprinting tools to determine what hosts are visible on the target network and, more importantly, try to ascertain what services and operating systems they are running. Visible services are researched further to tailor subsequent tests to match.
3. **Scanning**
   Vulnerability scanners are used to scan all discovered hosts for known vulnerabilities or weaknesses. The results are analyzed to determine if there are any vulnerabilities that could be exploited to gain access or enhance privileges to target hosts.
4. **Obtaining Access**
   We use the results of the scans to assist in attempting to obtain access to target systems and services, or to escalate privileges where access has been obtained (either legitimately though provided credentials, or via vulnerabilities). This may be done surreptitiously (for example to try to evade intrusion detection systems or rate limits) or by more aggressive brute-force methods. This step also consist of manually testing the application against the latest (2017) list of OWASP Top 10 risks. The discovered vulnerabilities from scanning and manual testing are moreover used to further elevate access on the application.

2.2 Risk Classification

Throughout the report, vulnerabilities or risks are labeled and categorized according to the Penetration Testing Execution Standard (PTES). For more information, see: http://www.pentest-standard.org/index.php/Reporting

These categories are:

- **Extreme**
  Extreme risk of security controls being compromised with the possibility of catastrophic financial/reputational losses occurring as a result.

- **High**

  High risk of security controls being compromised with the potential for significant financial/reputational losses occurring as a result.
- **Elevated**

  Elevated risk of security controls being compromised with the potential for material financial/reputational losses occurring as a result.
- **Moderate**

  Moderate risk of security controls being compromised with the potential for limited financial/reputational losses occurring as a result.
- **Low**

  Low risk of security controls being compromised with measurable negative impacts as a result.

3 Reconnaissance and Fingerprinting

We were able to gain information about the software and infrastructure through the following automated scans. Any relevant scan output will be referred to in the findings.

- Burp Suite Pro - https://portswigger.net/burp
- Nmap – https://nmap.org
- Testssl.sh – https://github.com/drwetter/testssl.sh

4 Findings

We have identified the following issues:

4.1 OTF3-022 — Hypha - Insecure ACL on API Comments

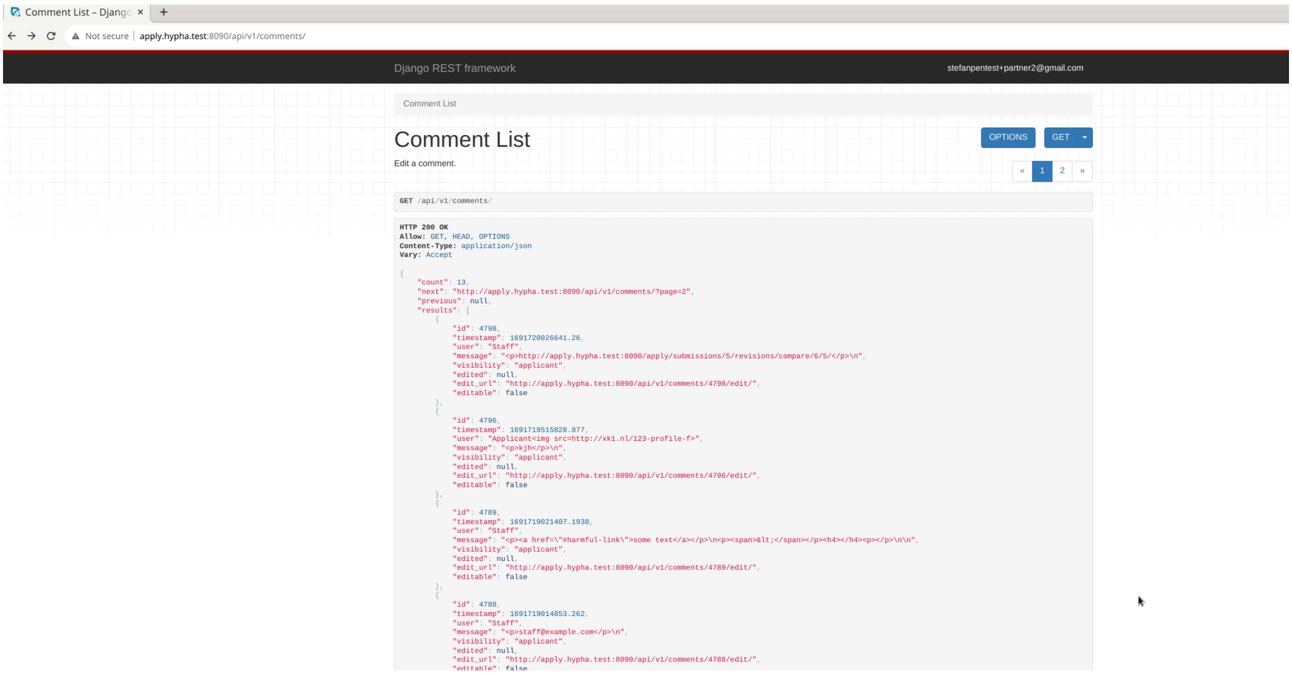
Vulnerability ID: OTF3-022	**Status:** Resolved
Vulnerability type: Insecure ACL	
Threat level: High	

Description:

The API Comments functionality does not have the proper ACL configured allowing other users such as applicants and partners to see the comments of others.

Technical description:

Partner using the API can see all other comments in the system:

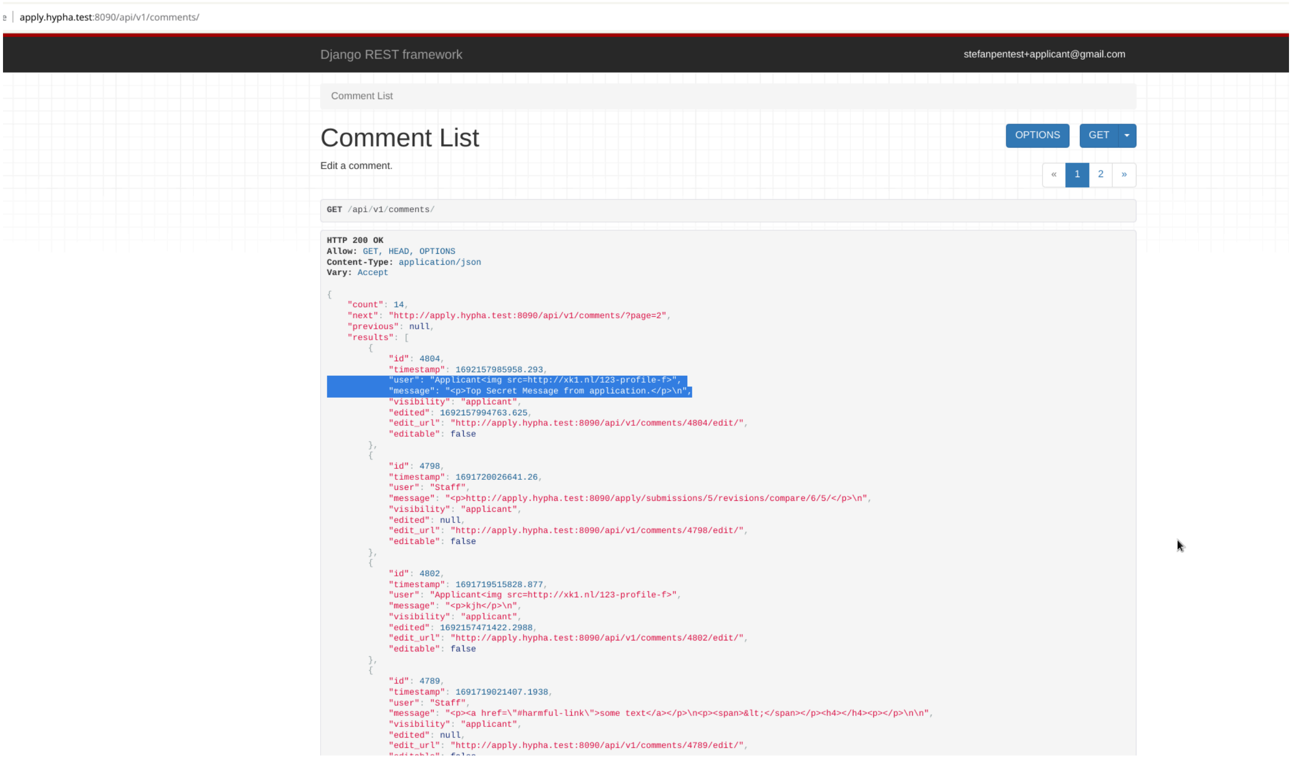

Comment List – Django ×
+
Not secure | apply.hypha.test:8090/api/v1/comments/
Django REST framework
stefanpentest+partner2@gmail.com
Comment List
OPTIONS
GET
« 1 2 »

Comment List

Edit a comment.

```
GET /api/v1/comments/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 13,
    "next": "http://apply.hypha.test:8090/api/v1/comments/?page=2",
    "previous": null,
    "results": [
        {
            "id": 4798,
            "timestamp": 1691720026641.26,
            "user": "Staff",
            "message": "<p>http://apply.hypha.test:8090/apply/submissions/5/revisions/compare/6/5/</p>\n",
            "visibility": "applicant",
            "edited": null,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4798/edit/",
            "editable": false
        },
        {
            "id": 4796,
            "timestamp": 1691719515828.877,
            "user": "Applicant<img src=http://xk1.nl/123-profile-f>",
            "message": "<p>kjh</p>\n",
            "visibility": "applicant",
            "edited": null,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4796/edit/",
            "editable": false
        },
        {
            "id": 4789,
            "timestamp": 1691719021407.1938,
            "user": "Staff",
            "message": "<p><a href=\"#harmful-link\">some text</a></p>\n<p><span>&lt;</span></p><h4></h4><p></p>\n\n",
            "visibility": "applicant",
            "edited": null,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4789/edit/",
            "editable": false
        },
        {
            "id": 4788,
            "timestamp": 1691719014853.262,
            "user": "Staff",
            "message": "<p>staff@example.com</p>\n",
            "visibility": "applicant",
            "edited": null,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4788/edit/",
            "editable": false
```

An applicant can see other applicant comments:


apply.hypha.test:8090/api/v1/comments/
Django REST framework
stefanpentest+applicant@gmail.com
Comment List
OPTIONS
GET
«
1
2
»

Comment List

Edit a comment.

```
GET /api/v1/comments/
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 14,
    "next": "http://apply.hypha.test:8090/api/v1/comments/?page=2",
    "previous": null,
    "results": [
        {
            "id": 4804,
            "timestamp": 1692157985958.293,
            "user": "Applicant<img src=http://xk1.nl/123-profile-f>",
            "message": "<p>Top Secret Message from application.</p>\n",
            "visibility": "applicant",
            "edited": 1692157994763.625,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4804/edit/",
            "editable": false
        },
        {
            "id": 4798,
            "timestamp": 1691720026641.26,
            "user": "Staff",
            "message": "<p>http://apply.hypha.test:8090/apply/submissions/5/revisions/compare/6/5/</p>\n",
            "visibility": "applicant",
            "edited": null,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4798/edit/",
            "editable": false
        },
        {
            "id": 4802,
            "timestamp": 1691719515828.877,
            "user": "Applicant<img src=http://xk1.nl/123-profile-f>",
            "message": "<p>kjh</p>\n",
            "visibility": "applicant",
            "edited": 1692157471422.2988,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4802/edit/",
            "editable": false
        },
        {
            "id": 4789,
            "timestamp": 1691719021407.1938,
            "user": "Staff",
            "message": "<p><a href=\"#harmful-link\">some text</a></p>\n<p><span>&lt;</span></p><h4></h4><p></p>\n\n",
            "visibility": "applicant",
            "edited": null,
            "edit_url": "http://apply.hypha.test:8090/api/v1/comments/4789/edit/",
            "editable": false
```

Impact:

Leaking this data to other users is a significant breach of the confidentiality and integrity of the application.

Recommendation:

Update the code to make sure that low privileged users such as applicants and partners cannot read other clients comments.

Update :

- Resolved, the API is not publicly exposed any more.

4.2 OTF3-021 — Hypha - Session Expiry is 14 days.

Vulnerability ID: OTF3-021 **Status:** Resolved

Vulnerability type: Improper Session Management

Threat level: Moderate

Description:

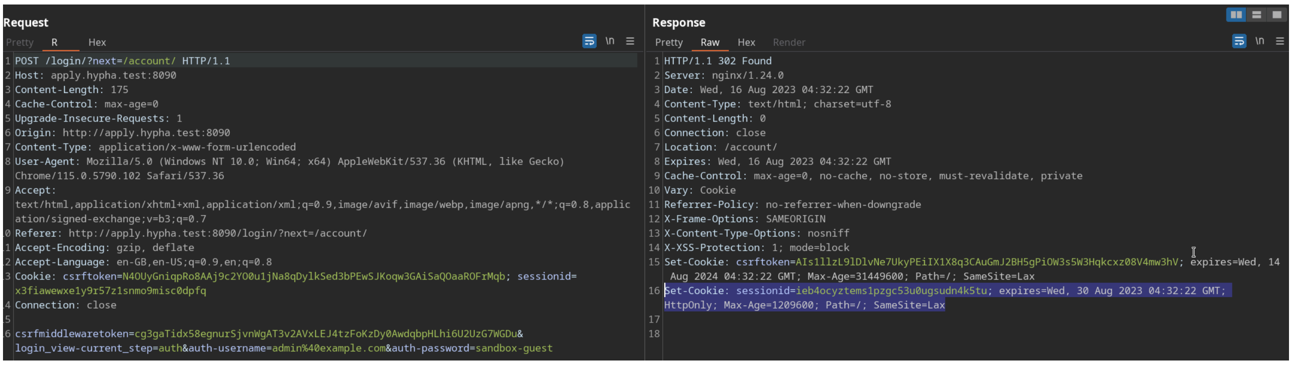
Sessions remain active after a user closes the application and remain valid for 14 days.

Technical description:

The following session management issues were found:

- Session remains active after a user closes or terminates the application.
- Session remain valid for 14 days.
- No limitation of a number of simultaneous logins with a single user account.

Session cookie is set to 14 days:



```
POST /login/?next=/account/ HTTP/1.1
Host: apply.hypha.test:8090
Content-Length: 175
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://apply.hypha.test:8090
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/login/?next=/account/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: csrftoken=N4OUyGniqpRo8AAj9c2YO0u1jNa8qDylkSed3bPEwSJKoqw3GAiSaQOaaROFrMqb; sessionid=
x3fiawewxe1y9r57z1snmo9misc0dpfq
Connection: close

csrfmiddlewaretoken=cg3gaTidx58egnurSjvnWgAT3v2AVxLEJ4tzFoKzDy0AwdqbpHLhi6U2UzG7WGDu&
login_view-current_step=auth&auth-username=admin%40example.com&auth-password=sandbox-guest
```

```
HTTP/1.1 302 Found
Server: nginx/1.24.0
Date: Wed, 16 Aug 2023 04:32:22 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 0
Connection: close
Location: /account/
Expires: Wed, 16 Aug 2023 04:32:22 GMT
Cache-Control: max-age=0, no-cache, no-store, must-revalidate, private
Vary: Cookie
Referrer-Policy: no-referrer-when-downgrade
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Set-Cookie: csrftoken=AIs1llzL9lDlvNe7UkyPEiIX1X8q3CAuGmJ2BH5gPiOW3s5W3Hqkcxz08V4mw3hV; expires=Wed, 14
 Aug 2024 04:32:22 GMT; Max-Age=31449600; Path=/; SameSite=Lax
Set-Cookie: sessionid=ieb4ocyztems1pzgc53u0ugsudn4k5tu; expires=Wed, 30 Aug 2023 04:32:22 GMT;
HttpOnly; Max-Age=1209600; Path=/; SameSite=Lax
```

Impact:

An attacker can resume a session that has not been properly invalidated and access the functionality available to the user of that session. This attack or exposure can be more damaging and practical if shared or public computers are used.

An attacker with access to application through a compromised account is able to create multiple sessions and can continue using this access with a lowered risk of detection. This weakness is simple to utilise and is by application design.

Recommendation:

- Reduce the default max session time to a lower value, e.g. 8 hours. Also make it possible for administrators to configure this value in the configuration file of the application.
- Set a maximum amount of active sessions per user. If possible only allow one active session per user. Note that some applications require the use of multiple devices (as they can be used on other devices such as a mobile phone and laptop), which discounts allowing only one active session as a solution. In this case the user could be shown a session overview (with the time when session was created and the kind of device it was created on) in case there are multiple active sessions. The user should be able to select and invalidate any session that should be deactivated.

Update :

This has been resolved. A `SESSION_COOKIE_AGE` setting has been added to the Django `base.py` configuration file.

4.3 OTF3-016 — Hypha - Improper Input Validation

Vulnerability ID: OTF3-016	**Status:** Not Retested
Vulnerability type: Improper Input Validation	
Threat level: Moderate	

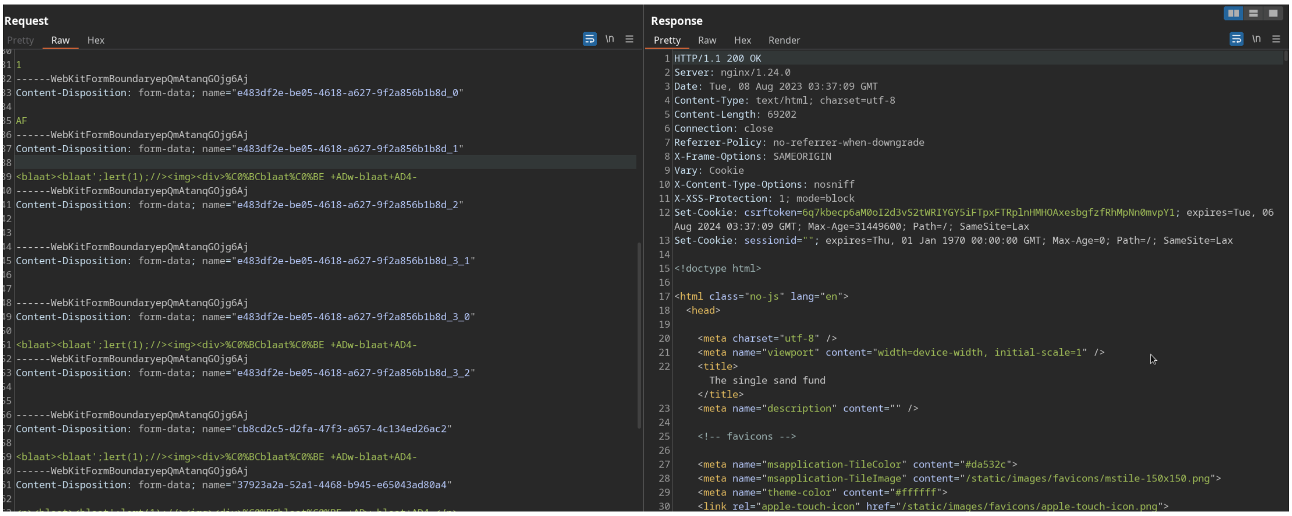
Description:

The application incorrectly validates input that can affect the control flow or data flow of a program.

Technical description:

Through the application dangerous input is accepted which could result in XSS vulnerabilities. It is important to not allow dangerous input in the first place by rejecting it. This can be done by first clientside - and secondly using server side validation.

The following form was sent containing dangerous characters and payload:



```
1
------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="e483df2e-be05-4618-a627-9f2a856b1b8d_0"

AF
------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="e483df2e-be05-4618-a627-9f2a856b1b8d_1"

<blaat><blaat';lert(1);//><img><div>%C0%BCblaat%C0%BE +ADw-blaat+AD4-
------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="e483df2e-be05-4618-a627-9f2a856b1b8d_2"


------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="e483df2e-be05-4618-a627-9f2a856b1b8d_3_1"


------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="e483df2e-be05-4618-a627-9f2a856b1b8d_3_0"

<blaat><blaat';lert(1);//><img><div>%C0%BCblaat%C0%BE +ADw-blaat+AD4-
------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="e483df2e-be05-4618-a627-9f2a856b1b8d_3_2"


------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="cb8cd2c5-d2fa-47f3-a657-4c134ed26ac2"

<blaat><blaat';lert(1);//><img><div>%C0%BCblaat%C0%BE +ADw-blaat+AD4-
------WebKitFormBoundaryepQmAtanqGOjg6Aj
Content-Disposition: form-data; name="37923a2a-52a1-4468-b945-e65043ad80a4"

```

```
HTTP/1.1 200 OK
Server: nginx/1.24.0
Date: Tue, 08 Aug 2023 03:37:09 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 69202
Connection: close
Referrer-Policy: no-referrer-when-downgrade
X-Frame-Options: SAMEORIGIN
Vary: Cookie
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Set-Cookie: csrftoken=6q7kbecp6aM0oI2d3vS2tWRIYGY5iFTpxFTRplnHMHOAxesbgfzfRhMpNn0mvpY1; expires=Tue, 06
Aug 2024 03:37:09 GMT; Max-Age=31449600; Path=/; SameSite=Lax
Set-Cookie: sessionid=""; expires=Thu, 01 Jan 1970 00:00:00 GMT; Max-Age=0; Path=/; SameSite=Lax

<!doctype html>

<html class="no-js" lang="en">
  <head>

    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>
      The single sand fund
    </title>
    <meta name="description" content="" />

    <!-- favicons -->

    <meta name="msapplication-TileColor" content="#da532c">
    <meta name="msapplication-TileImage" content="/static/images/favicons/mstile-150x150.png">
    <meta name="theme-color" content="#ffffff">
    <link rel="apple-touch-icon" href="/static/images/favicons/apple-touch-icon.png">
```

This behavior has been found in most parts of the application as well and we would recommend the developer to implement additional security to reduce the attack vector.

Impact:

Allowing dangerous input could lead to XSS and could also make it easier for zero days to succeed. While at this stage it does not result in a vulnerability allowing this does unnecessairly increase the attack vector.

Recommendation:

- Assume all input is malicious. Use an 'accept known good' input validation strategy i.e. use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does.
- When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules.
- Do not rely exclusively on looking for malicious or malformed inputs (i.e. do not rely on a blacklist). A blacklist is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.
- For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then these modified values would be submitted to the server.

- Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.
- When your application combines data from multiple sources, perform the validation after the sources have been combined. The individual data elements may pass the validation step but violate the intended restrictions after they have been combined. Inputs should be decoded and canonicalised to the application's current internal representation before being validated.
- Make sure that your application does not inadvertently decode the same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked.
- Consider performing repeated canonicalisation until your input does not change any more. This will avoid double-decoding and similar scenarios, but it might inadvertently modify inputs that are allowed to contain properly-encoded dangerous content.

Update :

OTF will leave this as it is for now. Long term we will start doing input filtering but for now we stick with filtering the output.

4.4 OTF3-008 — Wordpress - 2FA disabled for admin user.

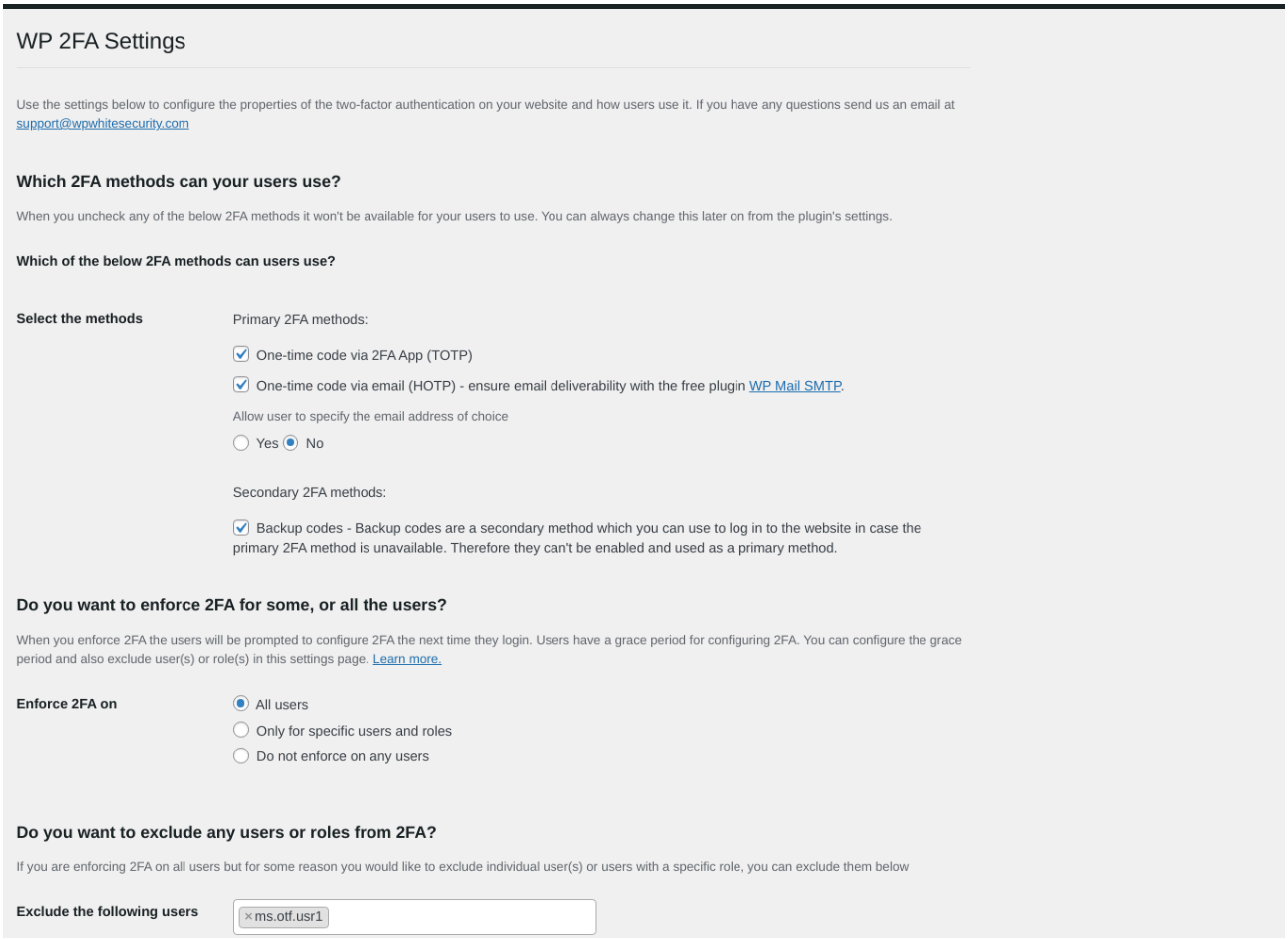
Vulnerability ID: OTF3-008	**Status:** Resolved
Vulnerability type: Insecure Password	
Threat level: Moderate	

Description:

The admin account is not protected by Two-Factor-Authentication (2FA).

Technical description:

It was found that the 2FA plugin is enabled and enforces 2FA for all users, except the admin account:



WP 2FA Settings

Use the settings below to configure the properties of the two-factor authentication on your website and how users use it. If you have any questions send us an email at support@wpwhitesecurity.com

Which 2FA methods can your users use?

When you uncheck any of the below 2FA methods it won't be available for your users to use. You can always change this later on from the plugin's settings.

Which of the below 2FA methods can users use?

Select the methods

Primary 2FA methods:

☑ One-time code via 2FA App (TOTP)

☑ One-time code via email (HOTP) - ensure email deliverability with the free plugin WP Mail SMTP.

Allow user to specify the email address of choice

○ Yes ● No

Secondary 2FA methods:

☑ Backup codes - Backup codes are a secondary method which you can use to log in to the website in case the primary 2FA method is unavailable. Therefore they can't be enabled and used as a primary method.

Do you want to enforce 2FA for some, or all the users?

When you enforce 2FA the users will be prompted to configure 2FA the next time they login. Users have a grace period for configuring 2FA. You can configure the grace period and also exclude user(s) or role(s) in this settings page. Learn more.

Enforce 2FA on

● All users

○ Only for specific users and roles

○ Do not enforce on any users

Do you want to exclude any users or roles from 2FA?

If you are enforcing 2FA on all users but for some reason you would like to exclude individual user(s) or users with a specific role, you can exclude them below

Exclude the following users × ms.otf.usr1

Impact:

Lack of 2FA authentication does not allow users to protect their account with a second factor which means that an account has to be considered breached when the password is leaked.

Recommendation:

Add support for 2FA authentication.

Update :

This has been resolved.

4.5 OTF3-005 — Wordpress - Wordpress username can be found through wp-json

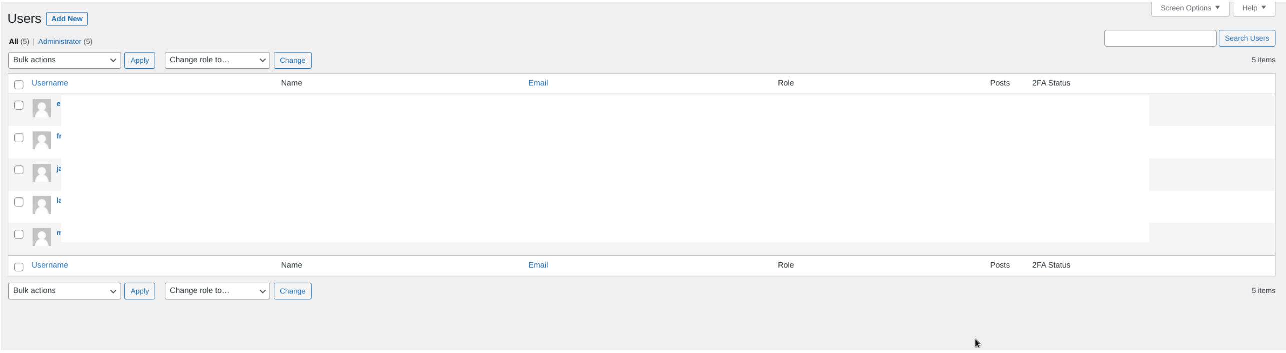
Vulnerability ID: OTF3-005	**Status:** Resolved
Vulnerability type: Misconfiguration	
Threat level: Moderate	

Description:

WordPress user details can be found through the wp-json endpoint.

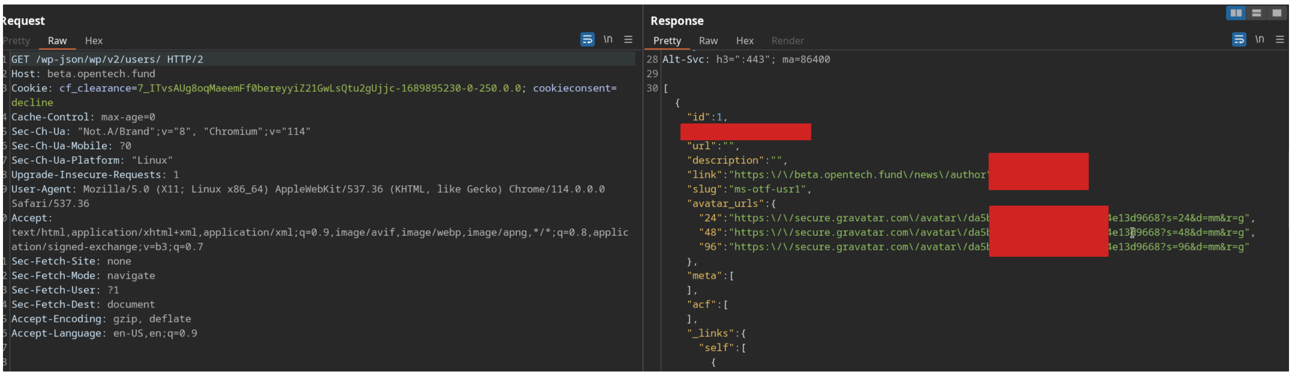
Technical description:

It was found that the default Wordpress admin user is not in use any more which is good:



□ Username	Name	Email	Role	Posts	2FA Status
□ e					
□ fr					
□ ja					
□ la					
□ m					
□ Username	Name	Email	Role	Posts	2FA Status

However the current administrator account can still be found by abusing The WP JSON functionality:



```
GET /wp-json/wp/v2/users/ HTTP/2
Host: beta.opentech.fund
Cookie: cf_clearance=7_ITvsAUg8oqMaeemFf0bereyyiZ21GwLsQtu2gUjjc-1689895230-0-250.0.0; cookieconsent=
decline
Cache-Control: max-age=0
Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="114"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
Alt-Svc: h3=":443"; ma=86400

[
  {
    "id":1,
    [redacted]
    "url":"",
    "description":"",
    "link":"https:\/\/beta.opentech.fund\/news\/author[redacted]
    "slug":"ms-otf-usr1",
    "avatar_urls":{
      "24":"https:\/\/secure.gravatar.com\/avatar\/da5b[redacted]4e13d9668?s=24&d=mm&r=g",
      "48":"https:\/\/secure.gravatar.com\/avatar\/da5b 4e13d9668?s=48&d=mm&r=g",
      "96":"https:\/\/secure.gravatar.com\/avatar\/da5b 4e13d9668?s=96&d=mm&r=g"
    },
    "meta":[
    ],
    "acf":[
    ],
    "_links":{
      "self":[
        {
```

Feedback from client:

I have used "rest_authentication_errors" filter to limit access to this end point to authenticated users.

Pantheon have an recommendation to do this I found, https://docs.pantheon.io/guides/wordpress-developer/wordpress-best-practices#disable-anonymous-access-to-wordpress-rest-api

Impact:

Anyone can directly access the affected endpoint and extract all user information. The user details can further be used to perform brute force attacks or targeted phishing.

Recommendation:

The WordPress installation has the REST API `/wp-json/wp/v2/`publicly accessible. It is recommend to disallow the public access, or disable the API if it is not needed.

Update :

Resolved: the site is now showing "Pantheon site locked" when attempting to access the wp-json functionality.

4.6 OTF3-028 — Hypha - XSS in Wagtail

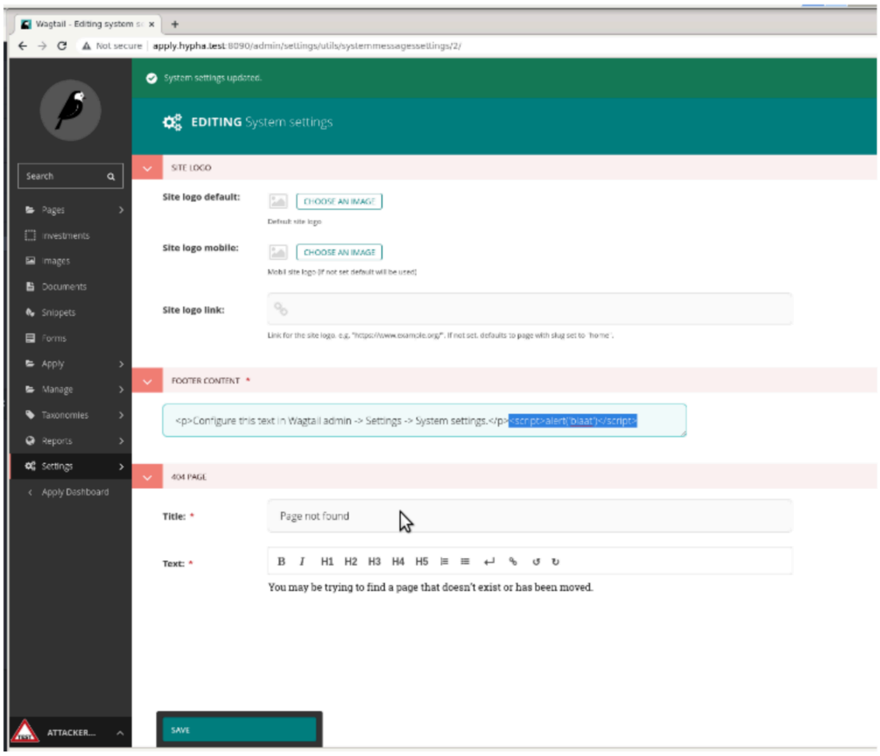
Vulnerability ID: OTF3-028	**Status:** Not Retested
Vulnerability type: XSS	
Threat level: Low	

Description:

Improper input validation in Wagtail CMS leads to multiple XSS vulnerabilities across the frontend and backend.

Technical description:

- Cross site scripting payload is allowed within the Footer resulting in XSS in the frontend. This issue was found during the previous pentest as well and requires Wagtail Admin authorizations.



apply.hypha.test:8090/admin/settings/utils/systemmessagessettings/2/

System settings updated.

EDITING System settings

SITE LOGO

Site logo default:

Default site logo

Site logo mobile:

Mobil site logo (if not set default will be used)

Site logo link:

Link for the site logo. e.g. "https://www.example.org/". If not set, defaults to page with slug set to 'home'.

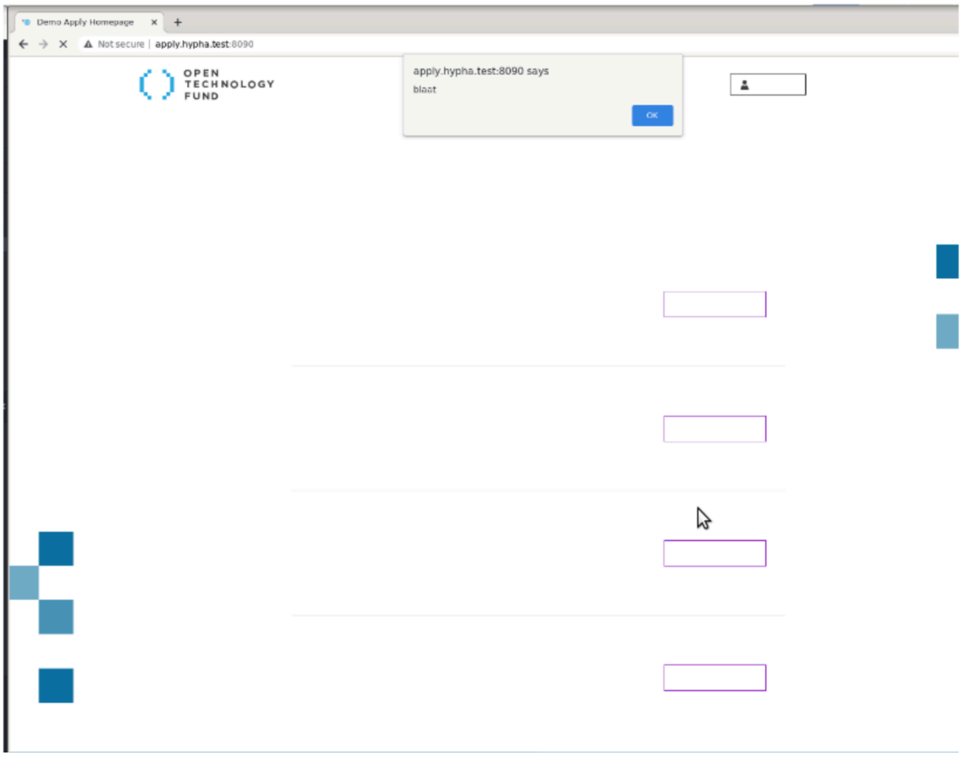
FOOTER CONTENT *

```
<p>Configure this text in Wagtail admin -> Settings -> System settings.</p><script>alert('blaat')</script>
```

404 PAGE

Title: * Page not found

Text: * You may be trying to find a page that doesn't exist or has been moved.

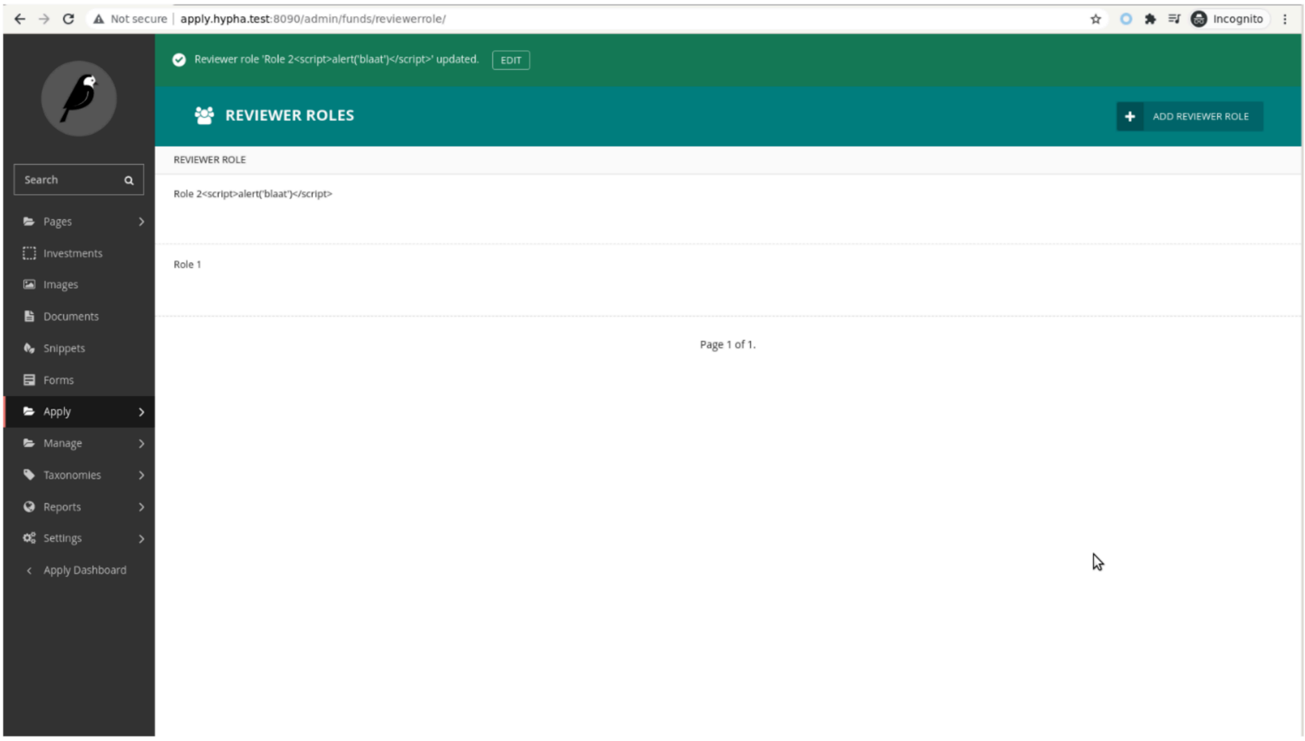


apply.hypha.test:8090

apply.hypha.test:8090 says

blaat

- XSS in Reviewer Role form. This issue was found during the previous pentest as well and requires minimally staff authorizations.

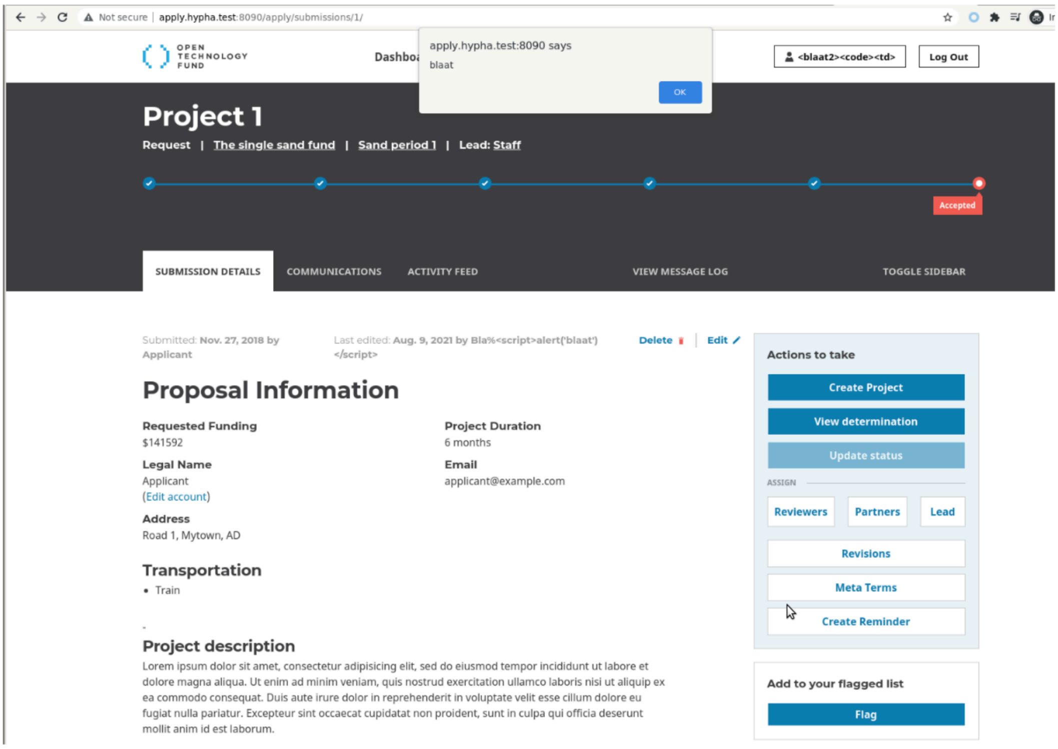


Reviewer role 'Role 2<script>alert('blaat')</script>' updated.

REVIEWER ROLES

REVIEWER ROLE
Role 2<script>alert('blaat')</script>
Role 1

Page 1 of 1.



apply.hypha.test:8090 says
blaat

Project 1

Request | The single sand fund | Sand period 1 | Lead: Staff

Submitted: Nov. 27, 2018 by Applicant

Last edited: Aug. 9, 2021 by Bla%<script>alert('blaat')</script>

Proposal Information

Requested Funding
$141592

Project Duration
6 months

Legal Name
Applicant
(Edit account)

Email
applicant@example.com

Address
Road 1, Mytown, AD

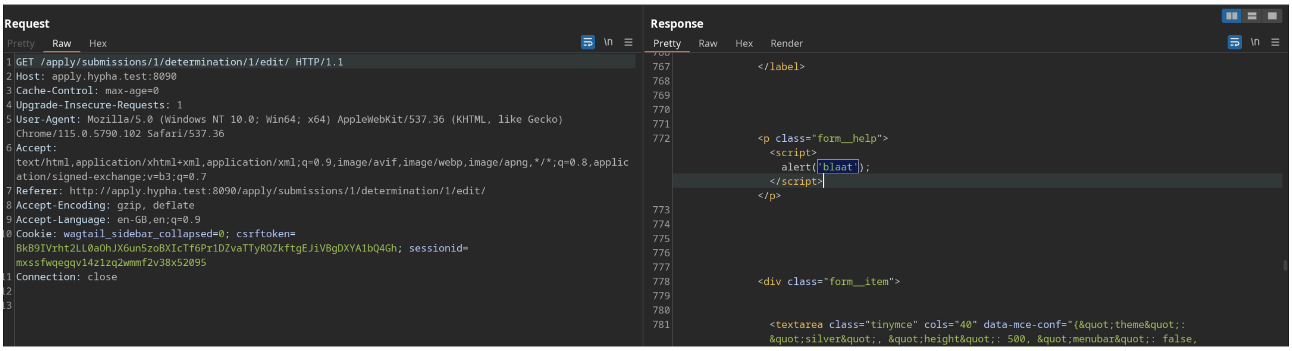
Transportation

- Train

Project description

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

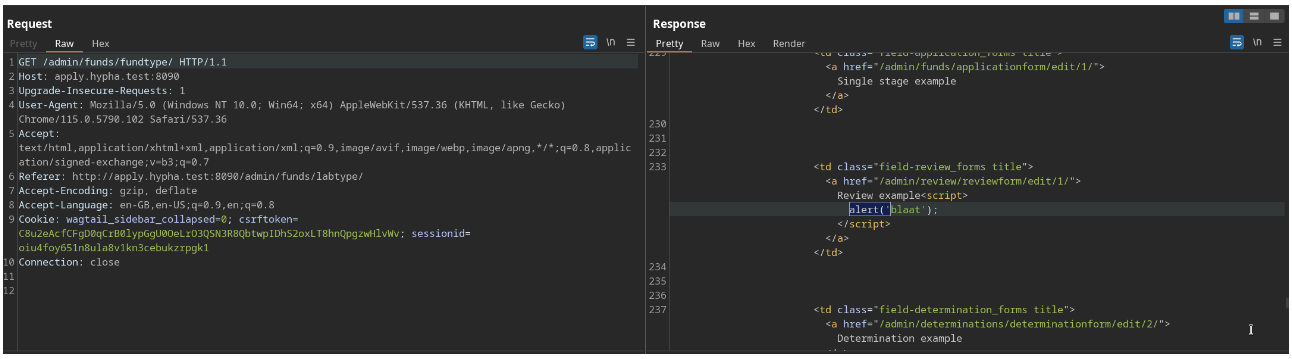
- XSS in helptext (/admin/settings/determinations/determinationformsettings/2/), requires at least Staff member authorizations and reflects the XSS in:



```
GET /apply/submissions/1/determination/1/edit/ HTTP/1.1
Host: apply.hypha.test:8090
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/apply/submissions/1/determination/1/edit/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9
Cookie: wagtail_sidebar_collapsed=0; csrftoken=
BkB9IVrht2LL0aOhJX6un5zoBXIcTf6Pr1DZvaTTyROZkftgEJiVBgDXYA1bQ4Gh; sessionid=
mxssfwqegqv14z1zq2wmmf2v38x52095
Connection: close
```

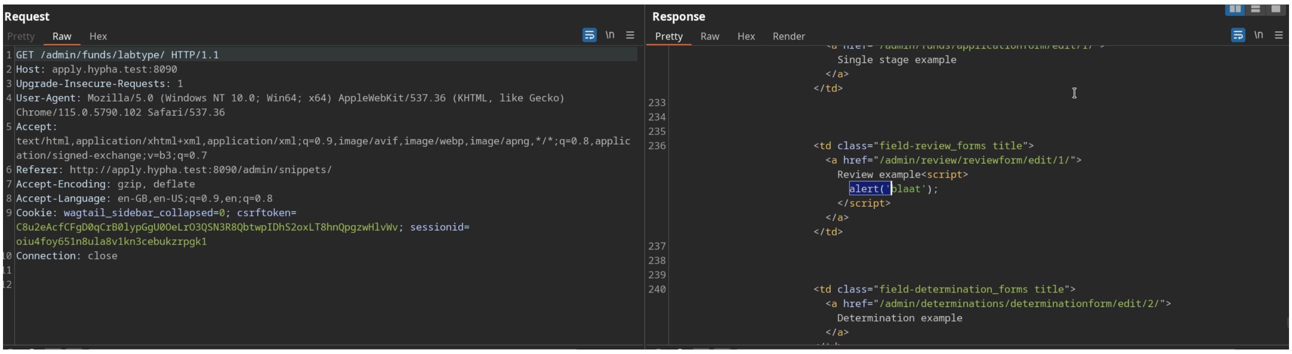
```
          </label>




          <p class="form__help">
            <script>
              alert('blaat');
            </script>
          </p>





          <div class="form__item">



            <textarea class="tinymce" cols="40" data-mce-conf="{&quot;theme&quot;:
            &quot;silver&quot;, &quot;height&quot;: 500, &quot;menubar&quot;: false,
```

- XSS in Fund form, requires at least Staff member authorizations. (Title and Type). This reflects back in:



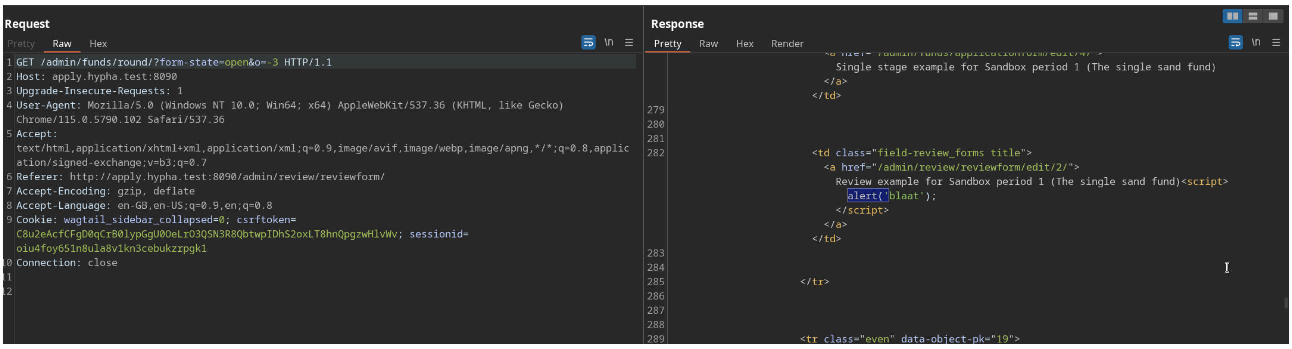
```
GET /admin/funds/fundtype/ HTTP/1.1
Host: apply.hypha.test:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/admin/funds/labtype/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: wagtail_sidebar_collapsed=0; csrftoken=
C8u2eAcfCFgD0qCrB0lypGgU0OeLrO3QSN3R8QbtwpIDhS2oxLT8hnQpgzwHlvWv; sessionid=
oiu4foy651n8ula8v1kn3cebukzrpgk1
Connection: close
```

```
              <a href="/admin/funds/applicationform/edit/1/">
                Single stage example
              </a>
            </td>



            <td class="field-review_forms title">
              <a href="/admin/review/reviewform/edit/1/">
                Review example<script>
                  alert('blaat');
                </script>
              </a>
            </td>



            <td class="field-determination_forms title">
              <a href="/admin/determinations/determinationform/edit/2/">
                Determination example
```



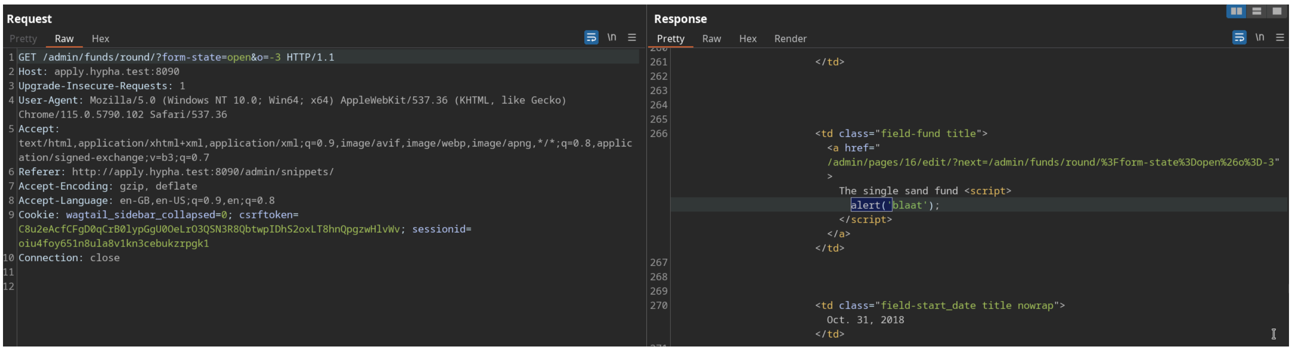
```
GET /admin/funds/labtype/ HTTP/1.1
Host: apply.hypha.test:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/admin/snippets/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: wagtail_sidebar_collapsed=0; csrftoken=
C8u2eAcfCFgD0qCrB0lypGgU0OeLrO3QSN3R8QbtwpIDhS2oxLT8hnQpgzwHlvWv; sessionid=
oiu4foy651n8ula8v1kn3cebukzrpgk1
Connection: close
```

```
                Single stage example
              </a>
            </td>



            <td class="field-review_forms title">
              <a href="/admin/review/reviewform/edit/1/">
                Review example<script>
                  alert('blaat');
                </script>
              </a>
            </td>



            <td class="field-determination_forms title">
              <a href="/admin/determinations/determinationform/edit/2/">
                Determination example
              </a>
```



```
GET /admin/funds/round/?form-state=open&o=-3 HTTP/1.1
Host: apply.hypha.test:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/admin/review/reviewform/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: wagtail_sidebar_collapsed=0; csrftoken=
C8u2eAcfCFgD0qCrB0lypGgU0OeLrO3QSN3R8QbtwpIDhS2oxLT8hnQpgzwHlvWv; sessionid=
oiu4foy651n8ula8v1kn3cebukzrpgk1
Connection: close
```

```
                Single stage example for Sandbox period 1 (The single sand fund)
              </a>
            </td>



            <td class="field-review_forms title">
              <a href="/admin/review/reviewform/edit/2/">
                Review example for Sandbox period 1 (The single sand fund)<script>
                  alert('blaat');
                </script>
              </a>
            </td>


          </tr>



          <tr class="even" data-object-pk="19">
```



```
GET /admin/funds/round/?form-state=open&o=-3 HTTP/1.1
Host: apply.hypha.test:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/admin/snippets/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: wagtail_sidebar_collapsed=0; csrftoken=
C8u2eAcfCFgD0qCrB0lypGgU0OeLrO3QSN3R8QbtwpIDhS2oxLT8hnQpgzwHlvWv; sessionid=
oiu4foy651n8ula8v1kn3cebukzrpgk1
Connection: close
```

```
                </td>




                <td class="field-fund title">
                  <a href="
                  /admin/pages/16/edit/?next=/admin/funds/round/%3Fform-state%3Dopen%26o%3D-3"
                  >
                    The single sand fund <script>
                      alert('blaat');
                    </script>
                  </a>
                </td>



                <td class="field-start_date title nowrap">
                  Oct. 31, 2018
                </td>
```

Impact:

This XSS can only be created and triggered by high privileged users (e.g staff and admin) which is the reason that this has been rated with a Low threatlevel. However it is still recommended to not allow XSS in the first place since a successful attack could lead to session hijack, credential stealing, or infecting systems with malware

Recommendation:

Implement proper input validation and output encoding to prevent XSS attacks.

More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting

Update :

This won't be resolved. OTF will leave this as it is. These forms will be removed when we remove the public part of Hypha.

4.7 OTF3-027 — Hypha - Inclusion of external Google Translate script.

Vulnerability ID: OTF3-027 **Status:** Not Retested

Vulnerability type: Cross Domain Inclusion

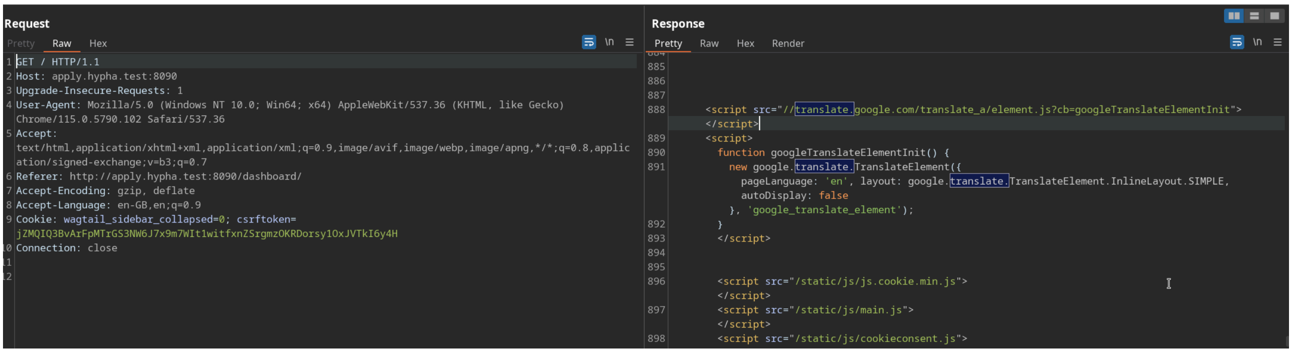
Threat level: Low

Description:

An external inclusion of the Google Translate script was found in the response of requests.

Technical description:

An external inclusion of the Google Translate script was found in the response of requests of the application:



```
GET / HTTP/1.1
Host: apply.hypha.test:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/dashboard/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9
Cookie: wagtail_sidebar_collapsed=0; csrftoken=
jZMQIQ3BvArFpMTrGS3NW6J7x9m7WIt1witfxnZSrgmzOKRDorsy1OxJVTkI6y4H
Connection: close
```

```



      <script src="//translate.google.com/translate_a/element.js?cb=googleTranslateElementInit">
      </script>
      <script>
        function googleTranslateElementInit() {
          new google.translate.TranslateElement({
            pageLanguage: 'en', layout: google.translate.TranslateElement.InlineLayout.SIMPLE,
            autoDisplay: false
          }, 'google_translate_element');
        }
        </script>


        <script src="/static/js/js.cookie.min.js">
        </script>
        <script src="/static/js/main.js">
        </script>
        <script src="/static/js/cookieconsent.js">
```

Client feedback:

Google Translate will most likely be removed when we remove the public part of Hypha in a few month time.

Impact:

A malicious third-party script can execute JavaScript and HTTP(S) requests on the included website in case an adversary is able to modify these contents.

Recommendation:

Try to avoid (where possible) using cross-domain scripts. Include and host as many scripts as possible on your own domain(s. This way, you keep control over the source code, as well as referrer information.

Update :

This was not ready to be retested.

4.8 OTF3-026 — Hypha - Wagtail Documents can be downloaded without authentication.

Vulnerability ID: OTF3-026 **Status:** Not Retested

Vulnerability type: Insecure ACL

Threat level: Low

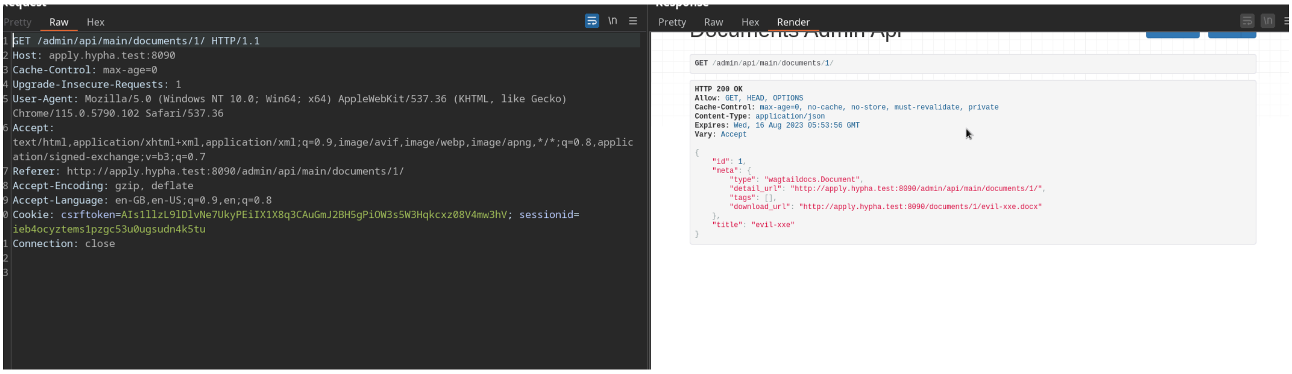
Description:

During the pentest we discovred that the Wagtail Documents can be downloaded without authentication.

Technical description:

Wagtail is the CMS used to manage the Hypha backend.

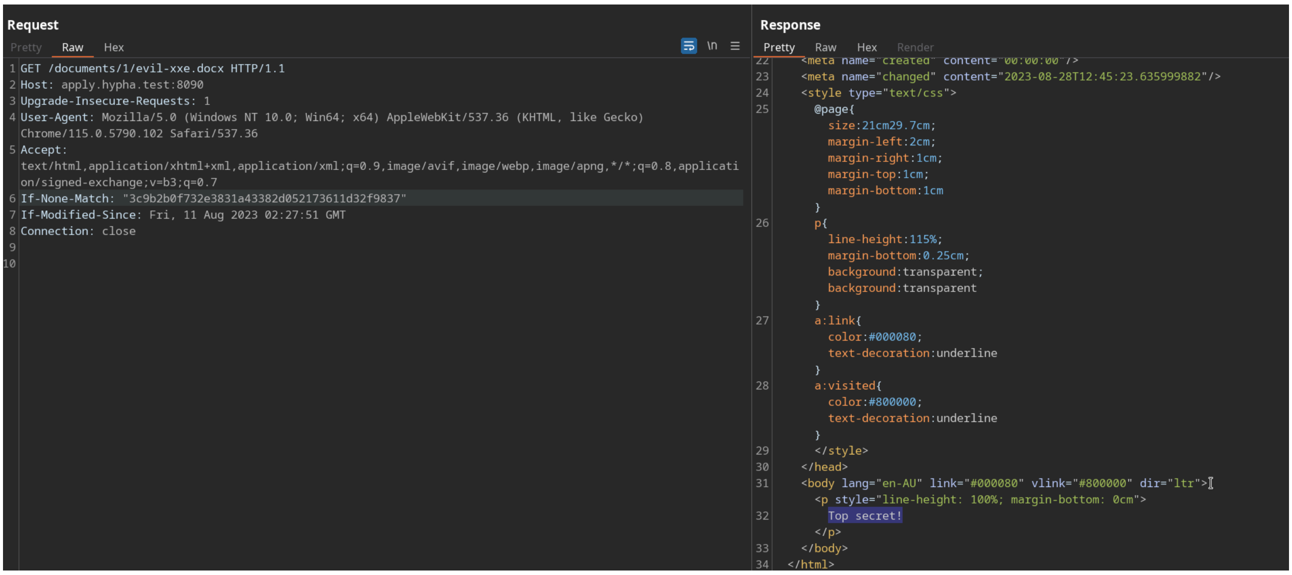
Fetching the document download URL:



```
GET /admin/api/main/documents/1/ HTTP/1.1
Host: apply.hypha.test:8090
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/admin/api/main/documents/1/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: csrftoken=AIs1llzL9lDlvNe7UkyPEiIX1X8q3CAuGmJ2BH5gPiOW3s5W3Hqkcxz08V4mw3hV; sessionid=
ieb4ocyztems1pzgc53u0ugsudn4k5tu
Connection: close
```

```
GET /admin/api/main/documents/1/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Cache-Control: max-age=0, no-cache, no-store, must-revalidate, private
Content-Type: application/json
Expires: Wed, 16 Aug 2023 05:53:56 GMT
Vary: Accept

{
    "id": 1,
    "meta": {
        "type": "wagtaildocs.Document",
        "detail_url": "http://apply.hypha.test:8090/admin/api/main/documents/1/",
        "tags": [],
        "download_url": "http://apply.hypha.test:8090/documents/1/evil-xxe.docx"
    },
    "title": "evil-xxe"
}
```

Dowloading the file without authentication:



```
GET /documents/1/evil-xxe.docx HTTP/1.1
Host: apply.hypha.test:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.7
If-None-Match: "3c9b2b0f732e3831a43382d052173611d32f9837"
If-Modified-Since: Fri, 11 Aug 2023 02:27:51 GMT
Connection: close
```

```
    <meta name="created" content="00:00:00"/>
    <meta name="changed" content="2023-08-28T12:45:23.635999882"/>
    <style type="text/css">
      @page{
        size:21cm29.7cm;
        margin-left:2cm;
        margin-right:1cm;
        margin-top:1cm;
        margin-bottom:1cm
      }
      p{
        line-height:115%;
        margin-bottom:0.25cm;
        background:transparent;
        background:transparent
      }
      a:link{
        color:#000080;
        text-decoration:underline
      }
      a:visited{
        color:#800000;
        text-decoration:underline
      }
      </style>
    </head>
    <body lang="en-AU" link="#000080" vlink="#800000" dir="ltr">
      <p style="line-height: 100%; margin-bottom: 0cm">
        Top secret!
      </p>
    </body>
  </html>
```

Impact:

Unauthenticated document access risks data exposure and potential info leaks. No current usage of this functionality was found by the Hypha application. However, future updates could change that. Hence, the lower Moderate threatlevel.

Recommendation:

Add authentication checks before downloading the document is allowed.

Update :

OTF will leave this as it is. This feature will soon be removed together with the whole public part of Hypha. Images and documents uploaded in the Wagtail CMS are only used on the public site so are ment to the public. Since we are removing the whole public part of Hypha soon these will also be removed/unused.

4.9 OTF3-024 — Hypha - Missing CSP Header

Vulnerability ID: OTF3-024 **Status:** Not Retested

Vulnerability type: Missing Security Header

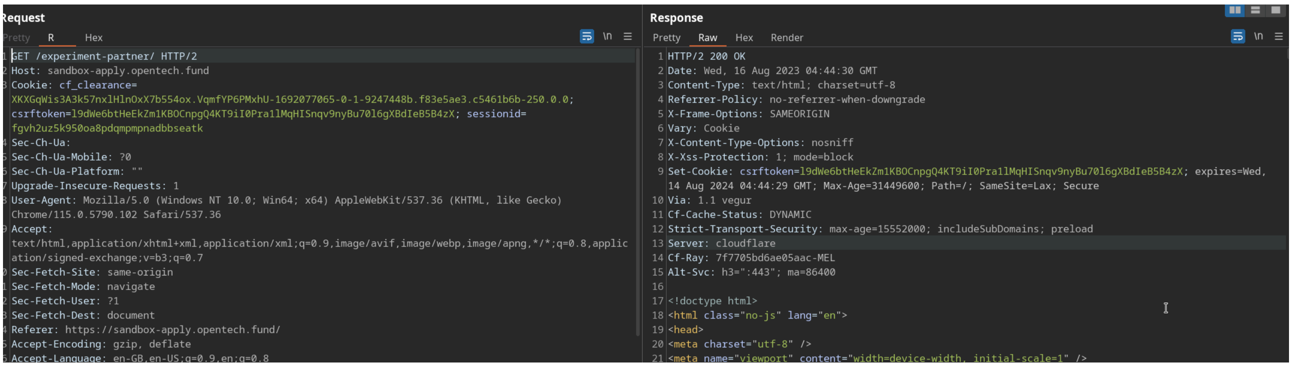
Threat level: Low

Description:

The Hypha application does not use the Content Security Policy header.

Technical description:

Notice that the CSP header is missing in the response of the following request:



```
GET /experiment-partner/ HTTP/2
Host: sandbox-apply.opentech.fund
Cookie: cf_clearance=
XKXGqWis3A3k57nxlHlnOxX7b554ox.VqmfYP6PMxhU-1692077065-0-1-9247448b.f83e5ae3.c5461b6b-250.0.0;
csrftoken=l9dWe6btHeEkZm1KBOCnpgQ4KT9iI0Pra1lMqHISnqv9nyBu70l6gXBdIeB5B4zX; sessionid=
fgvh2uz5k950oa8pdqmpmpnadbbseatk
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://sandbox-apply.opentech.fund/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
```

```
HTTP/2 200 OK
Date: Wed, 16 Aug 2023 04:44:30 GMT
Content-Type: text/html; charset=utf-8
Referrer-Policy: no-referrer-when-downgrade
X-Frame-Options: SAMEORIGIN
Vary: Cookie
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Set-Cookie: csrftoken=l9dWe6btHeEkZm1KBOCnpgQ4KT9iI0Pra1lMqHISnqv9nyBu70l6gXBdIeB5B4zX; expires=Wed,
14 Aug 2024 04:44:29 GMT; Max-Age=31449600; Path=/; SameSite=Lax; Secure
Via: 1.1 vegur
Cf-Cache-Status: DYNAMIC
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
Server: cloudflare
Cf-Ray: 7f7705bd6ae05aac-MEL
Alt-Svc: h3=":443"; ma=86400

<!doctype html>
<html class="no-js" lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

CSP is a tool which developers can use to lock down their applications in various ways, mitigating the risk of content injection vulnerabilities such as cross-site scripting, and reducing the privilege with which their applications execute.

Impact:

The impact of XSS or Clickjacking is low at this stage since the X-XSS-Protection and X-Frame-Options headers are already set.

Recommendation:

Note that X-XSS-Protection and X-Frame-Options headers are already set in the response which helps to mitigate these kind of attacks. However, best practice suggests to replace these headers for a CSP Header that blocks these attacks.

Update :

OTF will leave this as it is for now.

4.10 OTF3-023 — Hypha - Backup Tokens after confirmation can still be seen.

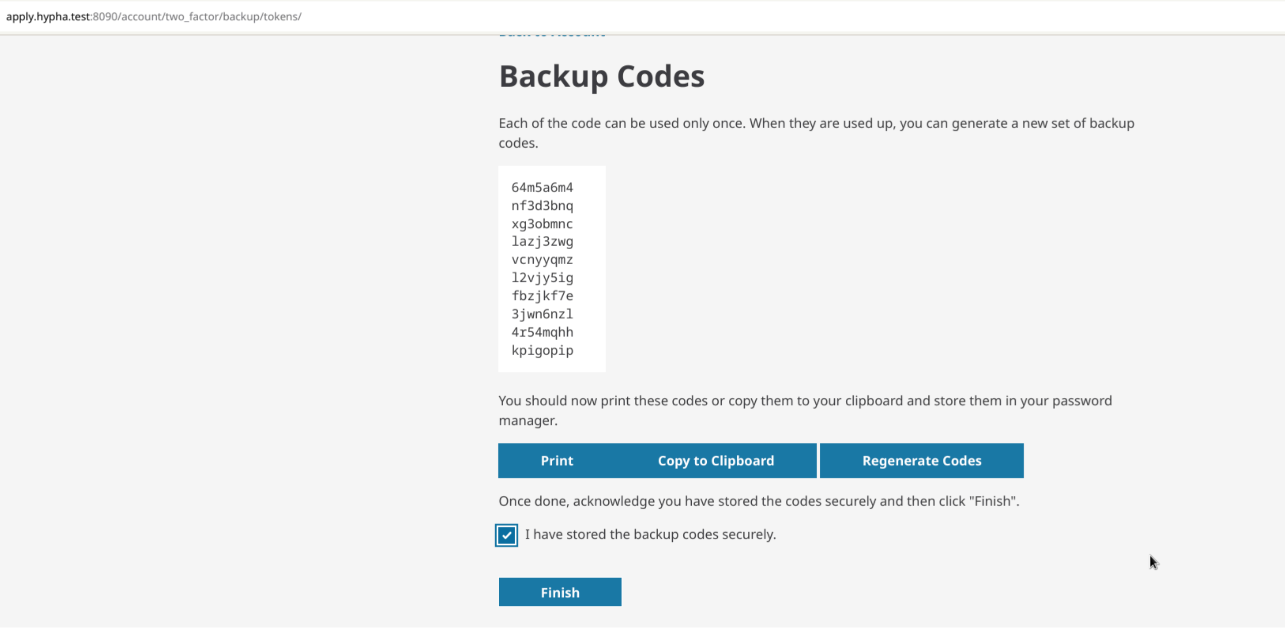
Vulnerability ID: OTF3-023	**Status:** Resolved
Vulnerability type: Insecure Password	
Threat level: Low	

Description:

Backup Tokens shown during 2FA setup stay the same and can be accessed by the user even after setup is complete.

Technical description:

During the setup of 2FA:



apply.hypha.test:8090/account/two_factor/backup/tokens/

Back to Account

Backup Codes

Each of the code can be used only once. When they are used up, you can generate a new set of backup codes.

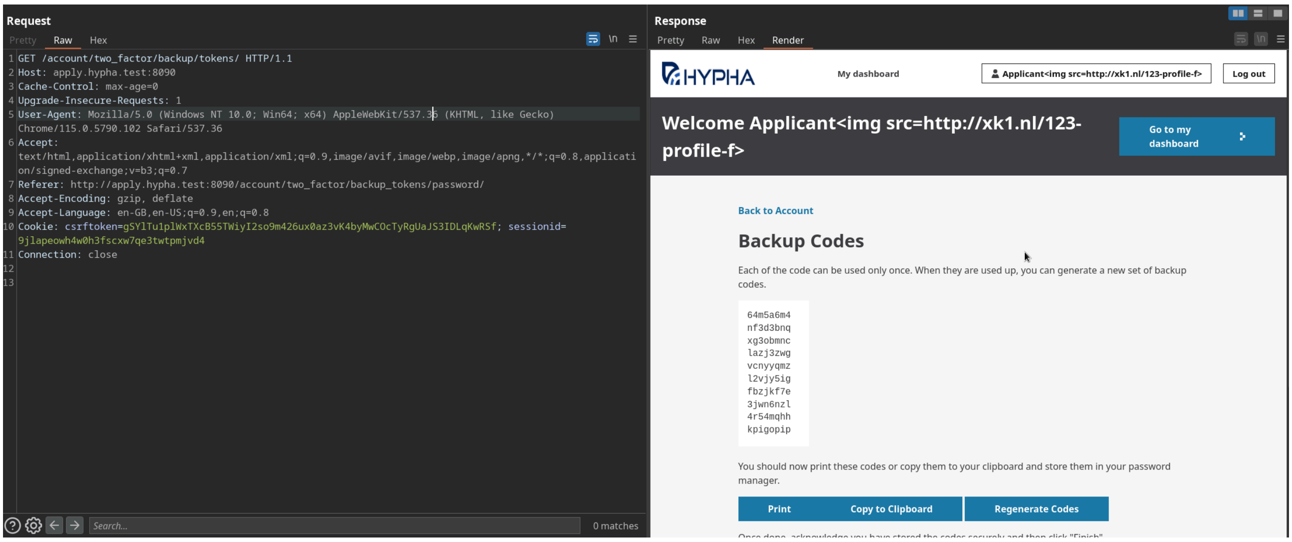
```
64m5a6m4
nf3d3bnq
xg3obmnc
lazj3zwg
vcnyyqmz
l2vjy5ig
fbzjkf7e
3jwn6nzl
4r54mqhh
kpigopip
```

You should now print these codes or copy them to your clipboard and store them in your password manager.

Once done, acknowledge you have stored the codes securely and then click "Finish".

☑ I have stored the backup codes securely.

User requesting these tokens after 2FA has been setup:



```
GET /account/two_factor/backup/tokens/ HTTP/1.1
Host: apply.hypha.test:8090
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/115.0.5790.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.7
Referer: http://apply.hypha.test:8090/account/two_factor/backup_tokens/password/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: csrftoken=gSYlTu1plWxTXcB55TWiyI2so9m426ux0az3vK4byMwCOcTyRgUaJS3IDLqKwRSf; sessionid=
9jlapeowh4w0h3fscxw7qe3twtpmjvd4
Connection: close


```

Welcome Applicant<img src=http://xk1.nl/123-profile-f>

Backup Codes

Each of the code can be used only once. When they are used up, you can generate a new set of backup codes.

```
64m5a6m4
nf3d3bnq
xg3obmnc
lazj3zwg
vcnyyqmz
l2vjy5ig
fbzjkf7e
3jwn6nzl
4r54mqhh
kpigopip
```

You should now print these codes or copy them to your clipboard and store them in your password manager.

Once done, acknowledge you have stored the codes securely and then click "Finish".

Impact:

If the backup tokens are accessible more than once, it means they can potentially be viewed by an unauthorized user if the account is ever compromised. This would provide an attacker with an alternative way to bypass 2FA.

Recommendation:

Make sure to follow the standard practice where backup tokens are only shown to the user once, typically at the time of setting up 2FA. Implementing proper access controls and logging around the viewing and usage of backup tokens can further enhance the security and integrity of the authentication process.

Update :

Resolved. The user's password is now required to access the backup tokens.

4.11 OTF3-020 — Hypha - External images can be attached

Vulnerability ID: OTF3-020	**Status:** Not Retested
Vulnerability type: Misconfiguration	
Threat level: Low	

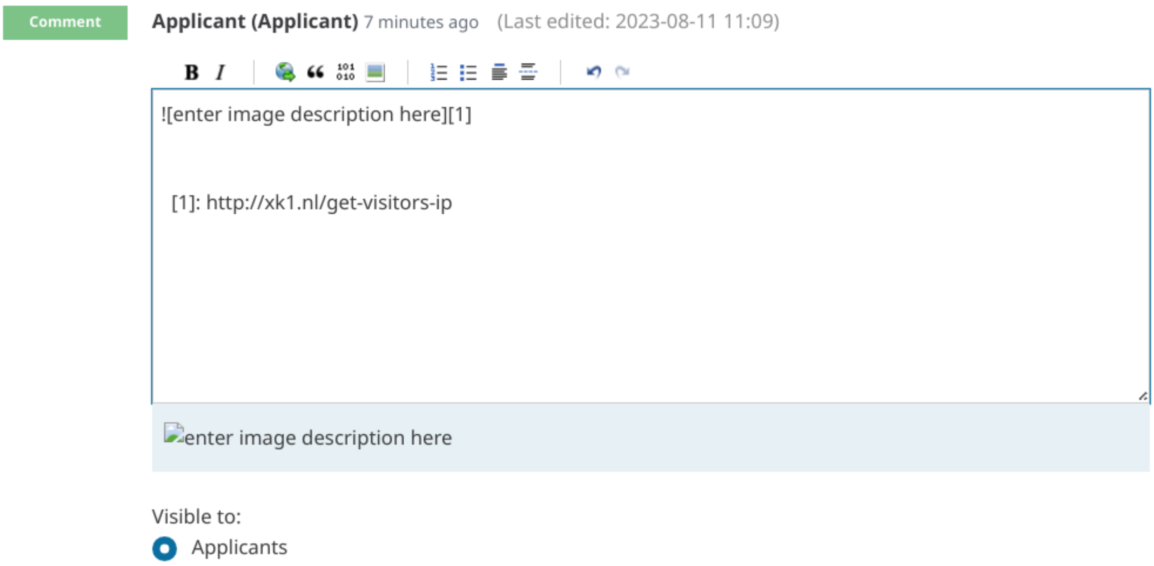
Description:

External images can be embedded within several forms throughout the application.

Technical description:

In various forms, externally hosted images can be included. When a user visits the page containing these images, the images are loaded, revealing the user's IP address.

Applicant adding an externally hosted image to a comment:



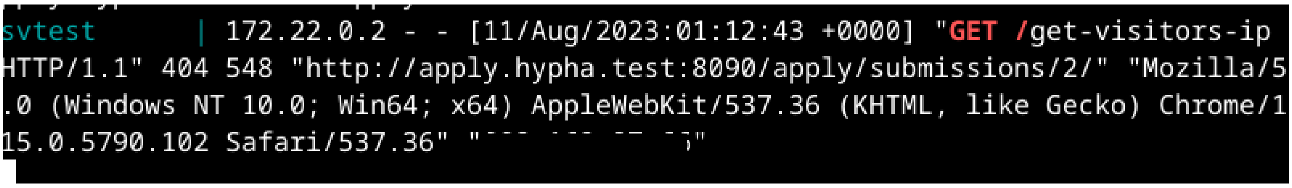
Applicant (Applicant) 7 minutes ago (Last edited: 2023-08-11 11:09)

```
![enter image description here][1]


  [1]: http://xk1.nl/get-visitors-ip
```

enter image description here

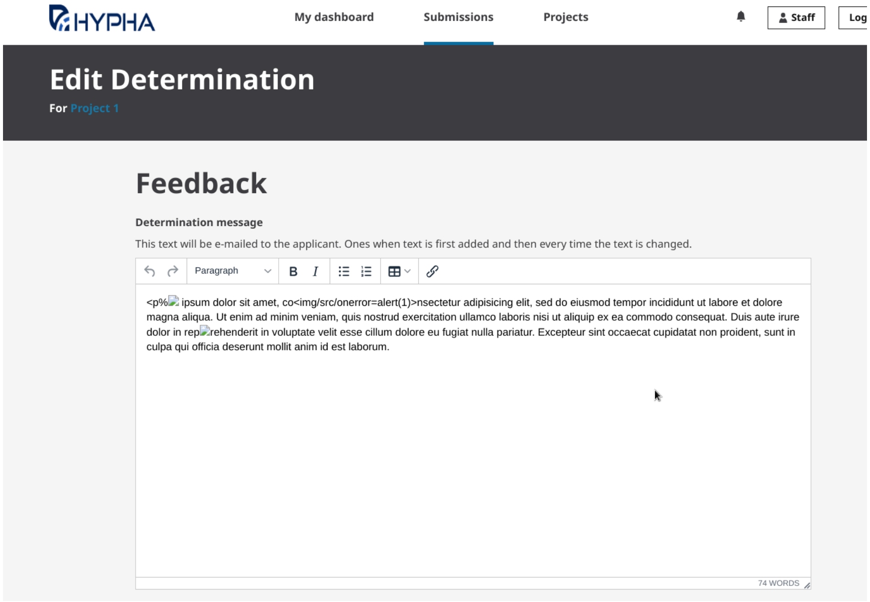
Visible to:
● Applicants

Webserver receives the IP of the visitor:



```
svtest      | 172.22.0.2 - - [11/Aug/2023:01:12:43 +0000] "GET /get-visitors-ip
HTTP/1.1" 404 548 "http://apply.hypha.test:8090/apply/submissions/2/" "Mozilla/5
.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
15.0.5790.102 Safari/537.36" "[redacted]"
```

Other forms that allow the usage of externally hosted images:



Edit Determination

For Project 1

Feedback

Determination message

This text will be e-mailed to the applicant. Ones when text is first added and then every time the text is changed.

<p% ipsum dolor sit amet, co<img/src/onerror=alert(1)>nsectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in rep rehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Impact:

Allowing adversaries to embed external images in a web application could expose IP addresses of application users to an adversary which can be used in further attacks.

Recommendation:

Disable external image uploads if possible. If images are needed, use the application's existing internal upload feature.

Update :

OTF will leave this as it is for now. Discussions on the way internally how big they feel this issue is. To completely protect ones location/ip address a VPN/Proxy service is needed.

Long term plan is maybe to replace the wysiwyg editor with a tool that converts pasted HTML in to markdown making it a lot easier to handle.

4.12 OTF3-018 — Hypha - No anti automation

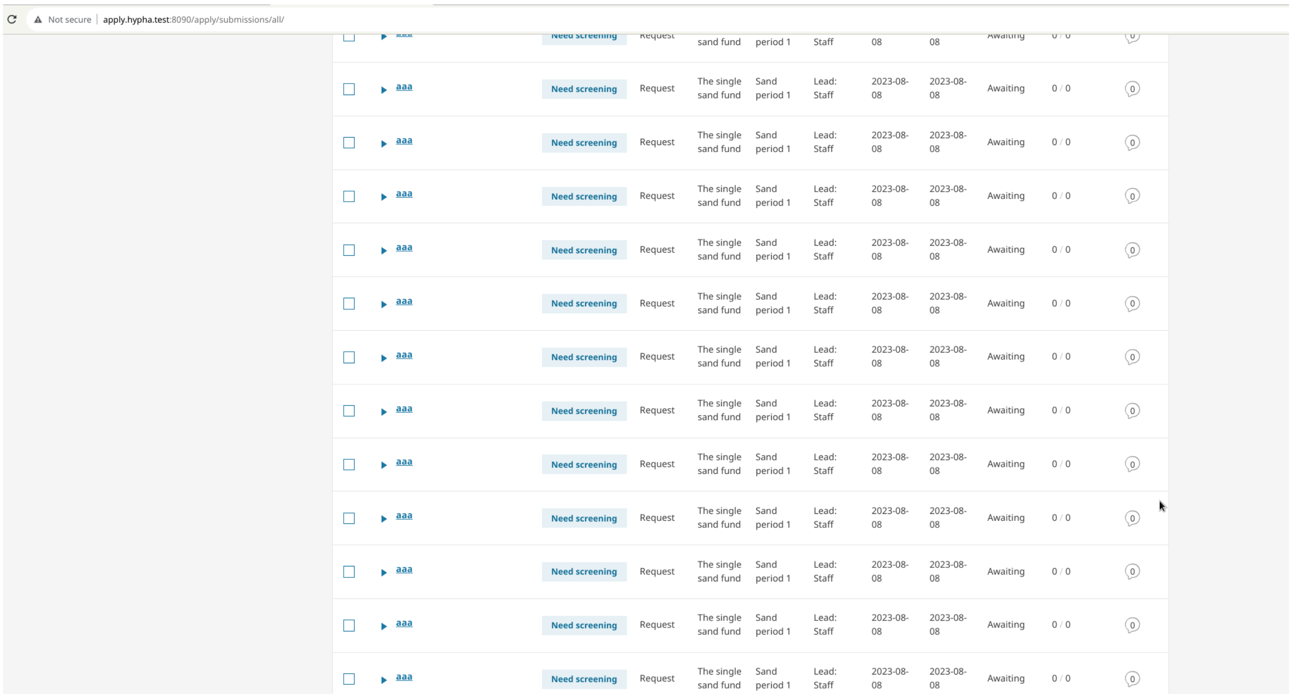
Vulnerability ID: OTF3-018	**Status:** Resolved
Vulnerability type: Missing Anti-Automation	
Threat level: Low	

Description:

The Apply form does not have proper Anti Automation functionality implemented.

Technical description:

Example of abusing the apply form:



Not secure | apply.hypha.test:8090/apply/submissions/all/

□	▸ aaa	Need screening	Request	sand fund	period 1	Staff	08	08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0
□	▸ aaa	Need screening	Request	The single sand fund	Sand period 1	Lead: Staff	2023-08-08	2023-08-08	Awaiting	0 / 0	0

Impact:

It is possible to automate the submission of this request with random data and flood the application's database with huge data. It may (technically) also lead to DOS attack on the application/database.

Recommendation:

Apply an anti-automation on forms. One of the common ways to do it would be implementing a Captcha (hCAPTCHA is very effective) on those pages and only show and enforce the use of it after a certain amount of requests per IP. Note that Cloudflare does have some protection against this but still it did allow us to make more requests then should be preferred.

Update :

Resolved. Rate limit has been implemented to the public forms such as login, password, 2FA and Mailchimp forms. Client mentions that OTF has Cloudflare WAF in front of Hypha so not an urgent concern. Long term they plan to add more internal features around this, mostly for other Hypha implementers.

4.13 OTF3-017 — Hypha - Low privileged user able to Purge CDN and Cache

Vulnerability ID: OTF3-017 **Status:** Not Retested

Vulnerability type: Insecure ACL

Threat level: Low

Description:

Low privileged users are able to Purge CDN and Cache.

Technical description:

Staff members (high privileged users), Editors and Moderators do not see the Purge CDN and Cache functionality in the User Interface but are still able to access and use the functionality by using the following URL's:

```
http://apply.hypha.test:8090/admin/cache/
http://apply.hypha.test:8090/admin/purge
```

Impact:

Impact is low since no possibility of abuse was found during testing, but new introduced functionality could make this issue more severe. In general it is recommended to prevent users accessing functionality they should not have access to.

Recommendation:

Verify whether the current user is allowed to access the requested resource and deny access if this is not the case.

Update :

OTF will leave this as it is. These features will soon be deleted when the whole public part of Hypha is removed.

4.14 OTF3-015 — Hypha - User Enumeration with Email Address Change

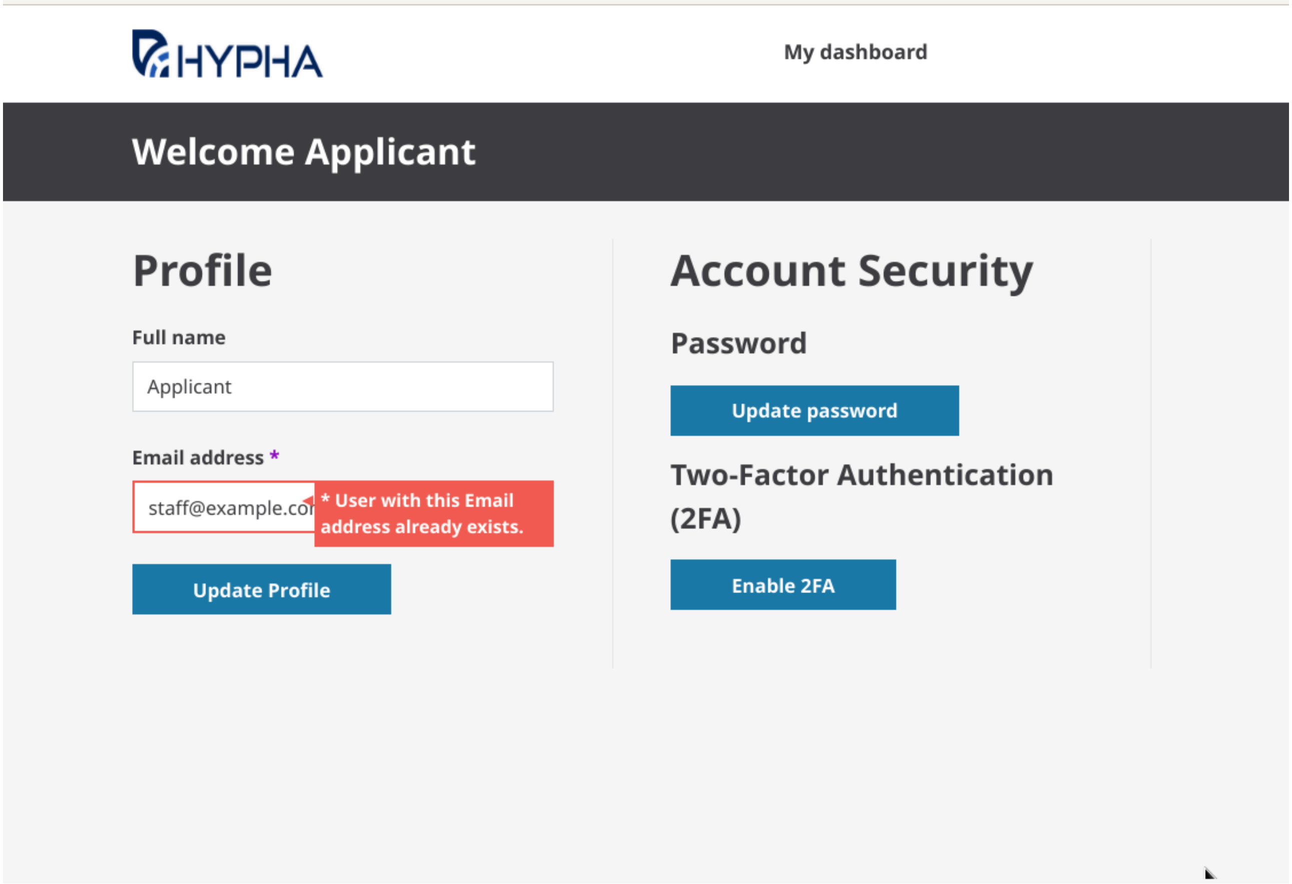
Vulnerability ID: OTF3-015	**Status:** Resolved
Vulnerability type: User Enumeration	
Threat level: Low	

Description:

Valid users can be found by abusing the Profile Change Email address functionality.

Technical description:

Changing to an existing email address shows an error that the email address already exists in the system:



Impact:

Valid usernames can be enumerated and used in further attacks.

Recommendation:

- Set a timeout of 1 second (so based on the respones time an attacker cannot conduct a timing attack to find out whether some processing happens or not if an account exists or not)
- Mention in the message to the user that a confirmation mail is send to the email address set. If the address already exists in the system an email won't be send. If it does not exist an email would be send. And only after confirming the email the address is updated in the email field.

Update :

Resolved. Email exists message not shown any more.

4.15 OTF3-014 — General - CBC Ciphers used

Vulnerability ID: OTF3-014	**Status:** Unresolved
Vulnerability type: TLS Misconfiguration	
Threat level: Low	

Description:

The webserver allows obsolete Cipher Block Chaining (CBC) encryption.

Technical description:

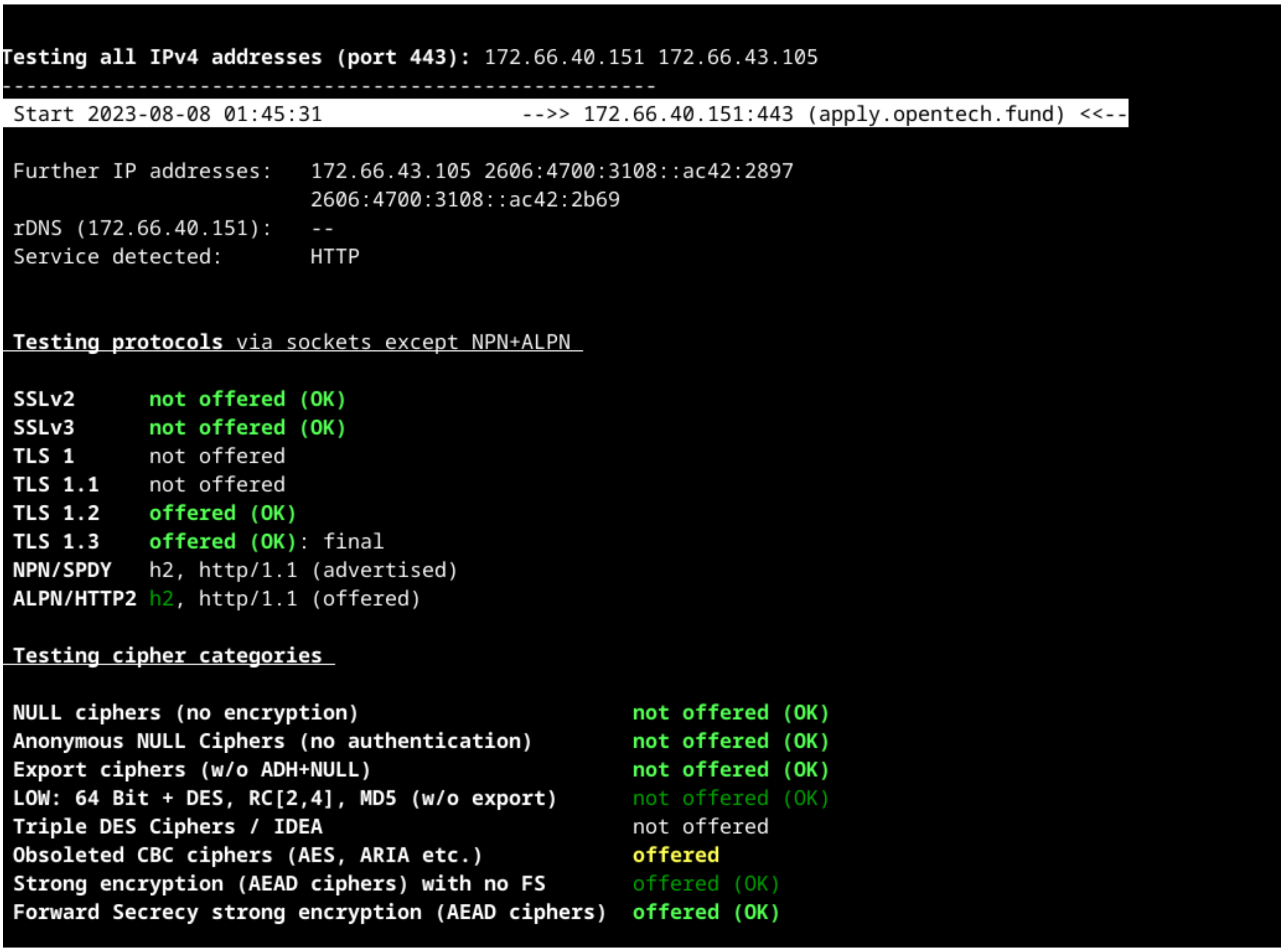
In cryptography, a padding oracle attack is an attack which uses the padding validation of a cryptographic message to decrypt the ciphertext.

Padding oracle attacks are mostly associated with CBC mode decryption used within block ciphers.

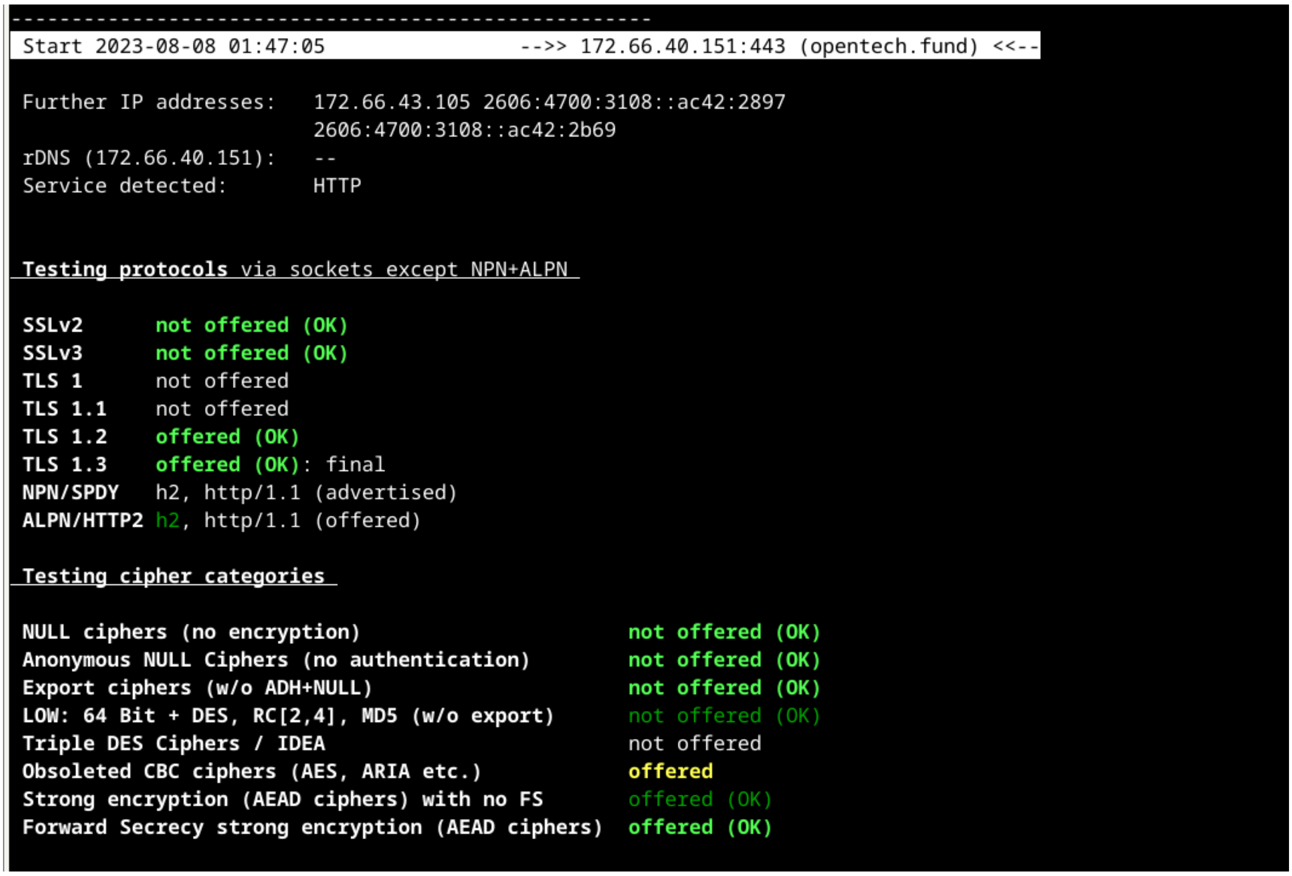
In symmetric cryptography, the padding oracle attack can be applied to the CBC mode of operation, where the "oracle" (usually a server) leaks data about whether the padding of an encrypted message is correct or not. Such data can allow attackers to decrypt (and sometimes encrypt) messages through the oracle using the oracle's key, without knowing the encryption key.

The webserver is configured to support Cipher Block Chaining (CBC) encryption on the following domains:

- apply.opentech.fund
- opentech.fund



```
Testing all IPv4 addresses (port 443): 172.66.40.151 172.66.43.105
--------------------------------------------------------------------
 Start 2023-08-08 01:45:31                -->> 172.66.40.151:443 (apply.opentech.fund) <<--

 Further IP addresses:   172.66.43.105 2606:4700:3108::ac42:2897
                         2606:4700:3108::ac42:2b69
 rDNS (172.66.40.151):   --
 Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   h2, http/1.1 (advertised)
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                      not offered (OK)
 Anonymous NULL Ciphers (no authentication)        not offered (OK)
 Export ciphers (w/o ADH+NULL)                     not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)      not offered (OK)
 Triple DES Ciphers / IDEA                         not offered
 Obsoleted CBC ciphers (AES, ARIA etc.)            offered
 Strong encryption (AEAD ciphers) with no FS       offered (OK)
 Forward Secrecy strong encryption (AEAD ciphers)  offered (OK)
```



```
-----------------------------------------------------
 Start 2023-08-08 01:47:05                --> 172.66.40.151:443 (opentech.fund) <<--

 Further IP addresses:   172.66.43.105 2606:4700:3108::ac42:2897
                         2606:4700:3108::ac42:2b69
 rDNS (172.66.40.151):   --
 Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   h2, http/1.1 (advertised)
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                      not offered (OK)
 Anonymous NULL Ciphers (no authentication)        not offered (OK)
 Export ciphers (w/o ADH+NULL)                     not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)      not offered (OK)
 Triple DES Ciphers / IDEA                         not offered
 Obsoleted CBC ciphers (AES, ARIA etc.)            offered
 Strong encryption (AEAD ciphers) with no FS       offered (OK)
 Forward Secrecy strong encryption (AEAD ciphers)  offered (OK)
```

Feedback client:

This is Cloudflare controlled. OTF already have TLS 1.2 as minimum. OTF will leave this at it is.

Impact:

An attacker properly positioned between a user and the server, for example in the same network segment as the victim, may be able to obtain unencrypted network traffic between the user and the server.

Recommendation:

Disable the use of TLS CBC ciphers. De-prioritizing these ciphers can also help minimize successful exploitation of real-world attacks. The attacker typically cannot force the selection of a specific cipher and therefore can only execute a CBC padding oracle attack if the client/server normally negotiates a vulnerable cipher.

4.16 OTF3-013 — Wordpress - WP Cron is enabled.

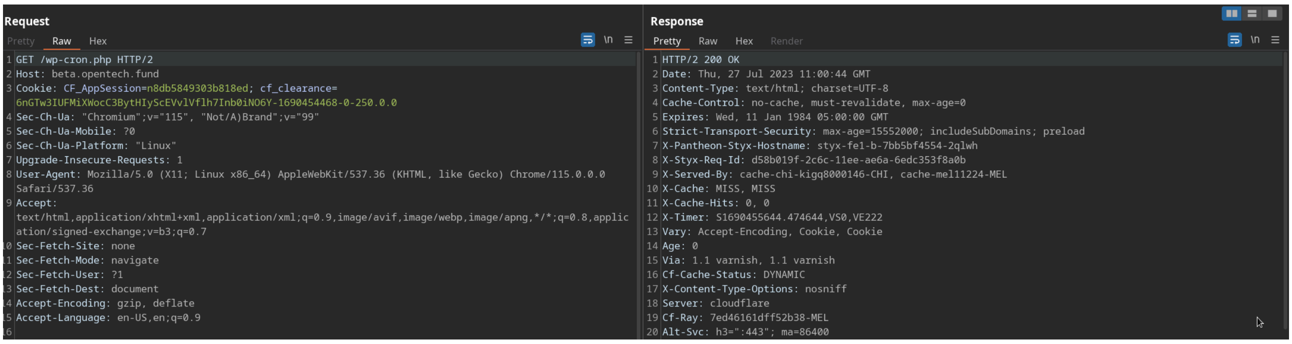
Vulnerability ID: OTF3-013	**Status:** Not Retested
Vulnerability type: Misconfiguration	
Threat level: Low	

Description:

The external WP-Cron appears to be enabled.

Technical description:

When opening the following url `https://beta.opentech.fund/wp-cron.php` it appears WP-Cron has been enabled:



```
GET /wp-cron.php HTTP/2
Host: beta.opentech.fund
Cookie: CF_AppSession=n8db5849303b818ed; cf_clearance=
6nGTw3IUFMiXWocC3BytHIyScEVvlVflh7Inb0iNO6Y-1690454468-0-250.0.0
Sec-Ch-Ua: "Chromium";v="115", "Not/A)Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

```

```
HTTP/2 200 OK
Date: Thu, 27 Jul 2023 11:00:44 GMT
Content-Type: text/html; charset=UTF-8
Cache-Control: no-cache, must-revalidate, max-age=0
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
X-Pantheon-Styx-Hostname: styx-fe1-b-7bb5bf4554-2qlwh
X-Styx-Req-Id: d58b019f-2c6c-11ee-ae6a-6edc353f8a0b
X-Served-By: cache-chi-kigq8000146-CHI, cache-mel11224-MEL
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1690455644.474644,VS0,VE222
Vary: Accept-Encoding, Cookie, Cookie
Age: 0
Via: 1.1 varnish, 1.1 varnish
Cf-Cache-Status: DYNAMIC
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 7ed46161dff52b38-MEL
Alt-Svc: h3=":443"; ma=86400
```

The wp-cron.php file is responsible for scheduled events in a WordPress website. By default, when a request is made, WordPress will generate an additional request from it to the wp-cron.php file. By generating a large number of requests to the website, it is therefore possible to make the site perform a DoS attack on itself.

Impact:

Increase of attack vector.

Recommendation:

Add the variable DISABLE_WP_CRON to true in the file wp-config.php and restrict access to the file wp-cron.php. In case the cron is used there are other ways to run a cronjob. The alternative is to create in the system a cronjob that executes the wp-cron.php script directly through PHP every minute and avoid Http requests. There are also plugins that can assist with this.

Update :

This was not ready for retesting.

4.17 OTF3-012 — Wordpress - Hardcoded SMTP credentials

Vulnerability ID: OTF3-012 **Status:** Not Retested

Vulnerability type: Hardcoded Credentials

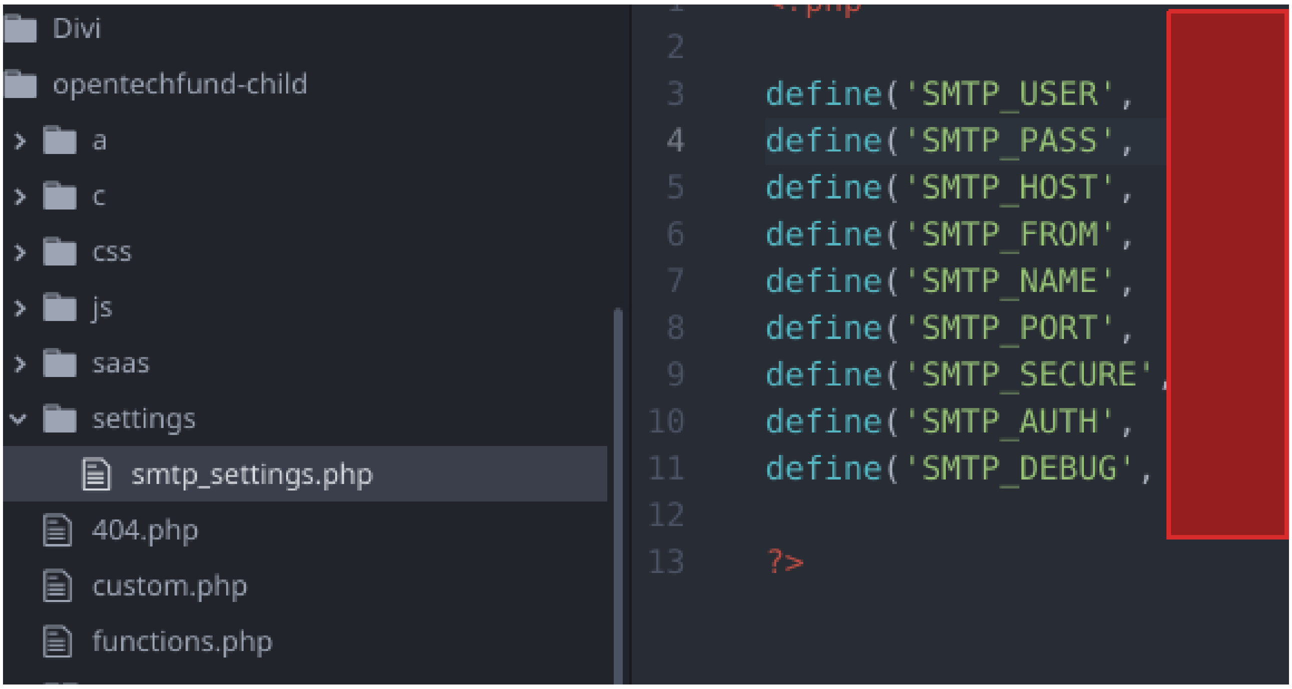
Threat level: Low

Description:

The SMTP configuration including credentials was found in the supplied source code.

Technical description:

Notice the SMTP configuration and credentials in the supplied source code:



```
<?php

define('SMTP_USER',
define('SMTP_PASS',
define('SMTP_HOST',
define('SMTP_FROM',
define('SMTP_NAME',
define('SMTP_PORT',
define('SMTP_SECURE',
define('SMTP_AUTH',
define('SMTP_DEBUG',
[redacted]

?>
```

Impact:

Leaked credentials in source repositories can be found by adversaries, for instance on a compromised development system. In the case of SMTP credentials these could be used for instance for phishing purposes to gain elevate privileges by triggering a high priviged user clicking on a link that appears to come from a domain they trust. Note that the impact is Low as the sourcecode was provided by the client and we neither did find this source code publicly exposed.

Recommendation:

- Do not store plaintext credentials.
- Remove hardcoded configuration files from source repositories.
- Set a strong password, unique for each environment.

Update :

This was not ready for retesting.

4.18 OTF3-010 — Wordpress - Disable File Edit in the UI

Vulnerability ID: OTF3-010	**Status:** Resolved
Vulnerability type: Misconfiguration	
Threat level: Low	

Description:

Wordpress backend files such as configuration and templates can be edited using the UI.

Technical description:

As a high privileged user such as an admin, configuration files can be edited by using the UI.

Impact:

This could lead to elevated privileges to the underlying system resulting in a full compromise of the server.

Recommendation:

Disable changing theme and plugin files within the GUI as an administrator in Wordpress by adding the following in wp-config.php: `define('DISALLOW_FILE_EDIT', true);`

Update :

This has been resolved.

4.19 OTF3-007 — Wordpress - Post via Email

Vulnerability ID: OTF3-007 **Status:** Resolved

Vulnerability type: Misconfiguration

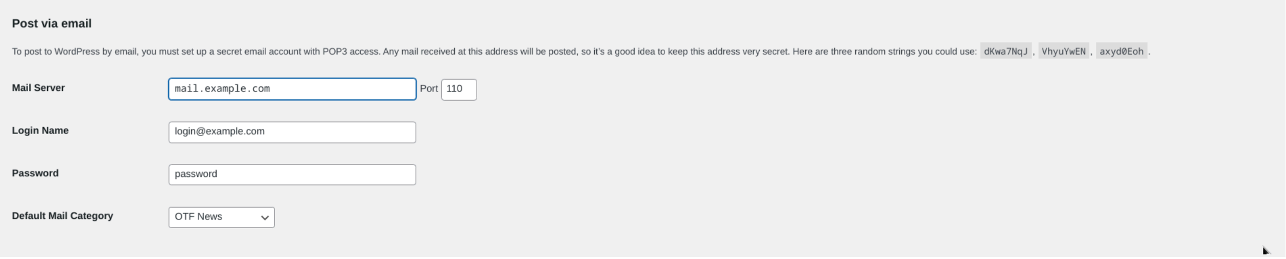
Threat level: Low

Description:

Default settings were found in the Post via Email functionality.

Technical description:

The following default parameters were found:



Post via email

To post to WordPress by email, you must set up a secret email account with POP3 access. Any mail received at this address will be posted, so it's a good idea to keep this address very secret. Here are three random strings you could use: `dKwa7NqJ` , `VhyuYwEN` , `axyd0Eoh` .

Mail Server	mail.example.com Port 110
Login Name	login@example.com
Password	password
Default Mail Category	OTF News

Feedback from client:

OTF has updated this on the beta.opentech.fund site now.

Impact:

As these are standard parameters a takeover of the example.com domain could result in defacing your website. While it is unclear that this functionality is working or not it is better to change it to prevent any potential future exploitation.

Recommendation:

Change the email settings to that of your own domain. Even if no mailserver exist.

Update :

This has been resolved.

4.20 OTF3-006 — Wordpress - Version is exposed.

Vulnerability ID: OTF3-006 **Status:** Resolved

Vulnerability type: Information Leak

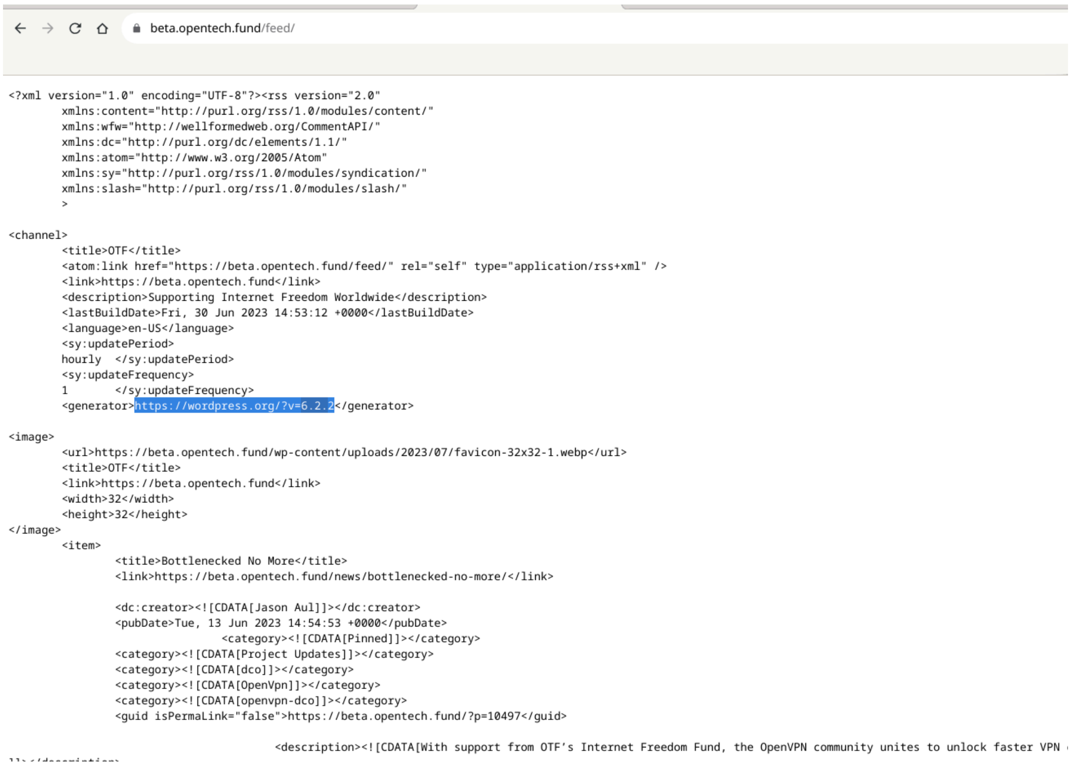
Threat level: Low

Description:

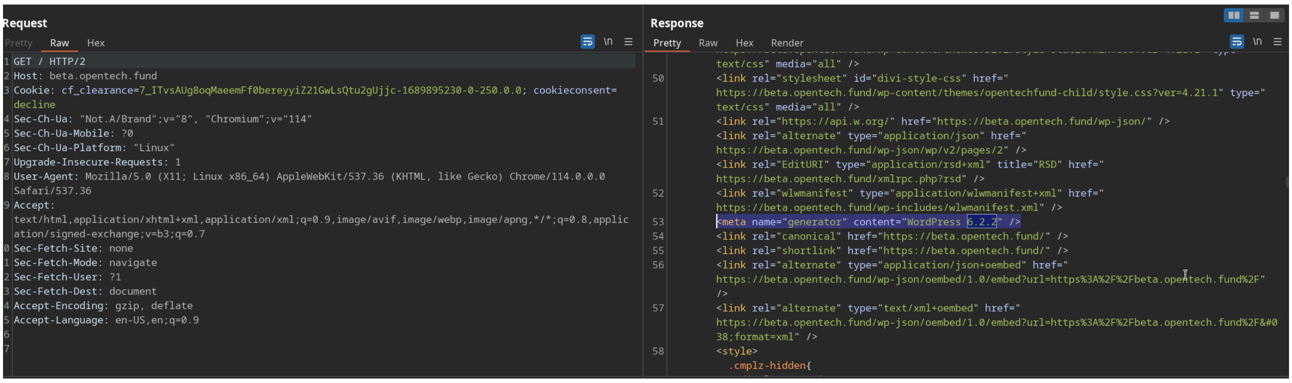
The Wordpress version is exposed.

Technical description:

Responses that reveal version information:



```
<?xml version="1.0" encoding="UTF-8"?><rss version="2.0"
        xmlns:content="http://purl.org/rss/1.0/modules/content/"
        xmlns:wfw="http://wellformedweb.org/CommentAPI/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:atom="http://www.w3.org/2005/Atom"
        xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
        xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
        >

<channel>
        <title>OTF</title>
        <atom:link href="https://beta.opentech.fund/feed/" rel="self" type="application/rss+xml" />
        <link>https://beta.opentech.fund</link>
        <description>Supporting Internet Freedom Worldwide</description>
        <lastBuildDate>Fri, 30 Jun 2023 14:53:12 +0000</lastBuildDate>
        <language>en-US</language>
        <sy:updatePeriod>
        hourly  </sy:updatePeriod>
        <sy:updateFrequency>
        1       </sy:updateFrequency>
        <generator>https://wordpress.org/?v=6.2.2</generator>

<image>
        <url>https://beta.opentech.fund/wp-content/uploads/2023/07/favicon-32x32-1.webp</url>
        <title>OTF</title>
        <link>https://beta.opentech.fund</link>
        <width>32</width>
        <height>32</height>
</image>
        <item>
                <title>Bottlenecked No More</title>
                <link>https://beta.opentech.fund/news/bottlenecked-no-more/</link>

                <dc:creator><![CDATA[Jason Aul]]></dc:creator>
                <pubDate>Tue, 13 Jun 2023 14:54:53 +0000</pubDate>
                                <category><![CDATA[Pinned]]></category>
                <category><![CDATA[Project Updates]]></category>
                <category><![CDATA[dco]]></category>
                <category><![CDATA[OpenVpn]]></category>
                <category><![CDATA[openvpn-dco]]></category>
                <guid isPermaLink="false">https://beta.opentech.fund/?p=10497</guid>

                                        <description><![CDATA[With support from OTF’s Internet Freedom Fund, the OpenVPN community unites to unlock faster VPN
]]></description>
```



```
GET / HTTP/2
Host: beta.opentech.fund
Cookie: cf_clearance=7_ITvsAUg8oqMaeemFf0bereyyiZ21GwLsQtu2gUjjc-1689895230-0-250.0.0; cookieconsent=
decline
Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="114"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
        text/css" media="all" />
        <link rel="stylesheet" id="divi-style-css" href="
        https://beta.opentech.fund/wp-content/themes/opentechfund-child/style.css?ver=4.21.1" type="
        text/css" media="all" />
        <link rel="https://api.w.org/" href="https://beta.opentech.fund/wp-json/" />
        <link rel="alternate" type="application/json" href="
        https://beta.opentech.fund/wp-json/wp/v2/pages/2" />
        <link rel="EditURI" type="application/rsd+xml" title="RSD" href="
        https://beta.opentech.fund/xmlrpc.php?rsd" />
        <link rel="wlwmanifest" type="application/wlwmanifest+xml" href="
        https://beta.opentech.fund/wp-includes/wlwmanifest.xml" />
        <meta name="generator" content="WordPress 6.2.2" />
        <link rel="canonical" href="https://beta.opentech.fund/" />
        <link rel="shortlink" href="https://beta.opentech.fund/" />
        <link rel="alternate" type="application/json+oembed" href="
        https://beta.opentech.fund/wp-json/oembed/1.0/embed?url=https%3A%2F%2Fbeta.opentech.fund%2F"
        />
        <link rel="alternate" type="text/xml+oembed" href="
        https://beta.opentech.fund/wp-json/oembed/1.0/embed?url=https%3A%2F%2Fbeta.opentech.fund%2F&#0
        38;format=xml" />
        <style>
          .cmplz-hidden{
```

Other files that expose information:

- https://beta.opentech.fund/readme.html
- https://beta.opentech.fund/README.md
- https://beta.opentech.fund/wp-admin/install.php
- https://beta.opentech.fund/license.txt

Feedback from client:

Found a code snippet to override the function that outputs the version.

Impact:

The impact is very low due to the current version not containing (any known) vulnerabilities. However showing the version in the response is not best practice as it allows adversaries to quickly determine whether the server could be vulnerable to future introduced exploits.

Recommendation:

Remove the information that exposes version information. Wordpress has several plugins and code snippets that can assist with this.

Update :

Resolved: the version in use is not exposed any more.

4.21 OTF3-004 — Wordpress - Exposed Management Interface

Vulnerability ID: OTF3-004 **Status:** Resolved

Vulnerability type: Exposed Management Interface

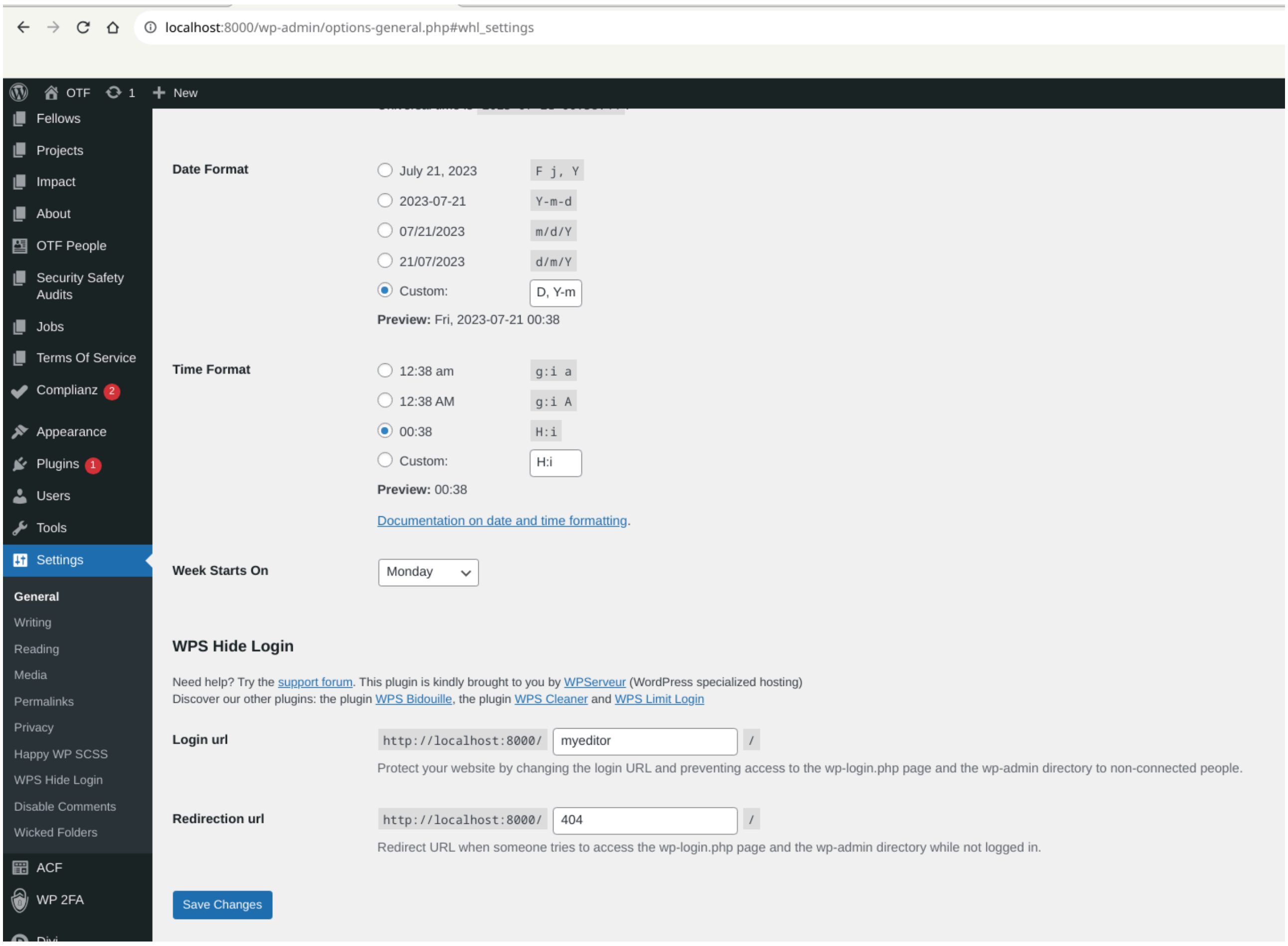
Threat level: Low

Description:

Access to the websites Wordpress management portal can be easily guessed which would allow public access to the management interface login portal.

Technical description:

The Website is using the `wps-hide-login` plugin which hides the standard `wp-admin` or `wplogin.php` page to access the management interface it is using a very simple name `myeditor` that could easily be found by bruteforcing it using a directory list.



Date Format

○ July 21, 2023 `F j, Y`
○ 2023-07-21 `Y-m-d`
○ 07/21/2023 `m/d/Y`
○ 21/07/2023 `d/m/Y`
● Custom: D, Y-m

Preview: Fri, 2023-07-21 00:38

Time Format

○ 12:38 am `g:i a`
○ 12:38 AM `g:i A`
● 00:38 `H:i`
○ Custom: H:i

Preview: 00:38

Documentation on date and time formatting.

Week Starts On Monday

WPS Hide Login

Need help? Try the support forum. This plugin is kindly brought to you by WPServeur (WordPress specialized hosting)
Discover our other plugins: the plugin WPS Bidouille, the plugin WPS Cleaner and WPS Limit Login

Login url `http://localhost:8000/` myeditor `/`

Protect your website by changing the login URL and preventing access to the wp-login.php page and the wp-admin directory to non-connected people.

Redirection url `http://localhost:8000/` 404 `/`

Redirect URL when someone tries to access the wp-login.php page and the wp-admin directory while not logged in.

Save Changes

Feedback client:

OTF will put wp-admin function behind Cloudflare access. Cloudflare access will be setup to only allow access to users with OTF Google accounts. Then they can scrap "security" features like "wps-hide-login".

Impact:

Any attacker who is able to successfully login or with access to a (0-day) exploit might be able to utilise successful attacks against the application environment to develop further attacks against other systems and users.

Recommendation:

Restrict access by consider implementing IP-whitelisting or using Cloudflare Access to reduce the attack vector.

Update :

Resolved: protected now by Cloudflare login.

4.22 OTF3-003 — Wordpress - Missing CSP Header

Vulnerability ID: OTF3-003 **Status:** Resolved

Vulnerability type: Missing Security Header

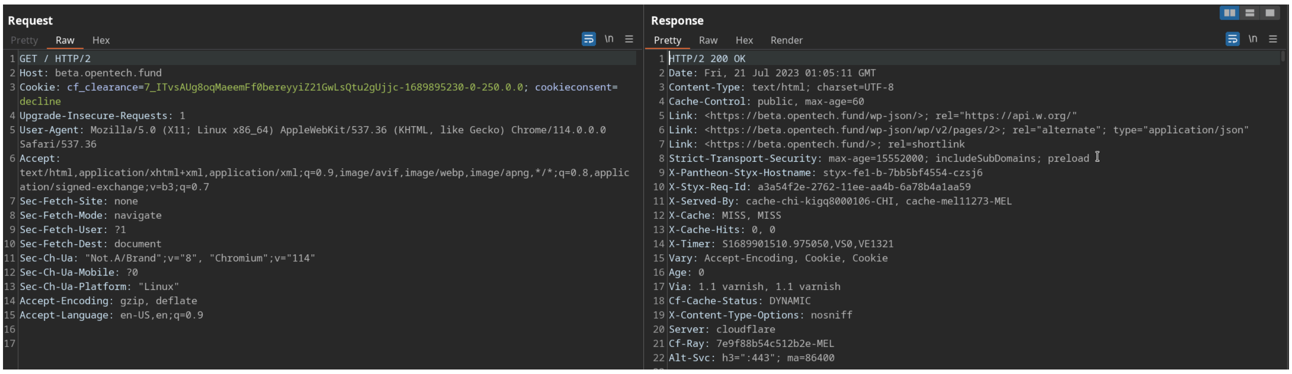
Threat level: Low

Description:

The application fails to set an appropriate Content-Security Policy header on some pages which could allow attacks such as XSS and Clickjacking.

Technical description:

This response does not contain a CSP header:



```
GET / HTTP/2
Host: beta.opentech.fund
Cookie: cf_clearance=7_ITvsAUg8oqMaeemFf0bereyyiZ21GwLsQtu2gUjjc-1689895230-0-250.0.0; cookieconsent=
decline
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="114"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
HTTP/2 200 OK
Date: Fri, 21 Jul 2023 01:05:11 GMT
Content-Type: text/html; charset=UTF-8
Cache-Control: public, max-age=60
Link: <https://beta.opentech.fund/wp-json/>; rel="https://api.w.org/"
Link: <https://beta.opentech.fund/wp-json/wp/v2/pages/2>; rel="alternate"; type="application/json"
Link: <https://beta.opentech.fund/>; rel=shortlink
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
X-Pantheon-Styx-Hostname: styx-fe1-b-7bb5bf4554-czsj6
X-Styx-Req-Id: a3a54f2e-2762-11ee-aa4b-6a78b4a1aa59
X-Served-By: cache-chi-kigq8000106-CHI, cache-mel11273-MEL
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1689901510.975050,VS0,VE1321
Vary: Accept-Encoding, Cookie, Cookie
Age: 0
Via: 1.1 varnish, 1.1 varnish
Cf-Cache-Status: DYNAMIC
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 7e9f88b54c512b2e-MEL
Alt-Svc: h3=":443"; ma=86400
```

CSP is a tool which developers can use to lock down their applications in various ways, mitigating the risk of content injection vulnerabilities such as cross-site scripting, and reducing the privilege with which their applications execute.

Impact:

This allows several attacks such as an attacker who found a XSS vulnerability and is able to bypass the browser filters to load JavaScript from external servers under his control.

Recommendation:

Include a strict CSP header in every response.

Update :

Resolved: the CSP header has been added by the client and now blocks our clickjacking payload.

4.23 OTF3-002 — Wordpress - Public access to Development and Test websites

Vulnerability ID: OTF3-002	**Status:** Resolved
Vulnerability type: Insecure ACL	
Threat level: Low	

Description:

The Development and Test websites lack password protection allowing unauthorized access.

Technical description:

The following test / staging websites do not require credentials to be accessed:

- https://test-otf-public.pantheonsite.io
- https://dev-otf-public.pantheonsite.io

Feedback from client:

Passwords are set now for all environments. Also we have added redirect for the live site to beta.opentech.fund which will be www.opentech.fund.

Impact:

Development and staging websites often undergo continuous changes, and new code might introduce temporary vulnerabilities that can be exploited by an adversary which increases the attack vector.

Recommendation:

Require credentials before access is allowed to Test and Development websites.

Update :

Resolved: now requires credentials to access the Development and Test website.

4.24 OTF3-001 — Wordpress - ACL Hardening

Vulnerability ID: OTF3-001	**Status:** Resolved
Vulnerability type: Insecure ACL	
Threat level: Low	

Description:

Several directories and files are not properly hardened.

Technical description:

The following functionality can be accessed:

- https://beta.opentech.fund/wp-admin/admin-ajax.php
- https://beta.opentech.fund/wp-cron.php

The following directories should be hardened:

- https://beta.opentech.fund/wp-includes/
- https://beta.opentech.fund/wp-content
- https://beta.opentech.fund/wp-admin
- https://beta.opentech.fund/wp-content/uploads/

Client feedback:

The dirs "wp-admin", "wp-includes" and "wp-content" are now all behind Cloudflare access. Only OTF staff have access.

Impact:

Increase of attack vector using an insecure configuration.

Recommendation:

- Restrict Admin access by adding restrictions who can reach the the wp-admin directory. Examples how to do this are using basic authentication or Cloudflare Access.
- Disable WP-CRON (note this sometimes breaks some plugins)
- Disable executing PHP files by web-users in restrictive directories.
- Add additional hardening to protect against potential arbitrary file upload
- Several Wordpress plugins can assist with restricting access and additionally using other best security practices.

Update :

Resolved: Access is now denied to these directories.

5 Non-Findings

In this section we list some of the things that were tried but turned out to be dead ends.

5.1 NF-025 — Hypha - Forms can be submitted without agreeing the terms

A form can be submitted without agreeing to the terms.

Client feedback: These forms are created by staff and if they want a checkbox to be required they mark it as such during the form creation. This is a none issue since it up to users of Hypha to create the forms they need.

5.2 NF-019 — General - Retesting findings previous report

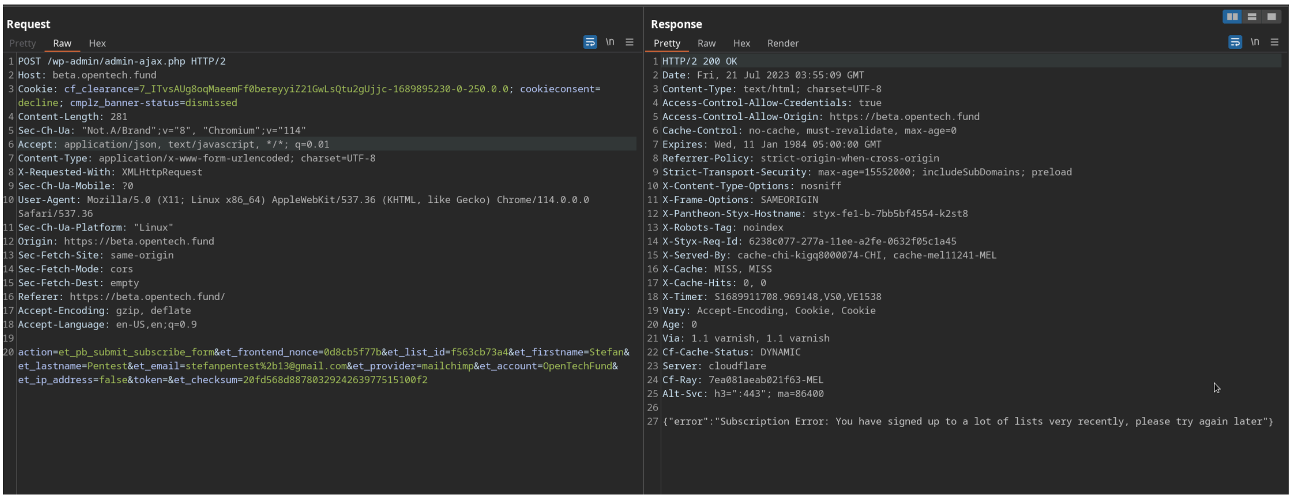
1. All findings have been retested.
2. New issues have been created for findings that remain.
3. The XSS findings related to the Hypha frontent, that were previously managed by Wagtail, have not been retested as this has been replaced by Wordpress.

5.3 NF-011 — Wordpress - Anti Automation in Newsletter Signup

Anti Automation is used to protect the Newsletter Signup form. This is done by the plugin and Cloudflare as well.

There are ways to bypass this protection, for instance by using different IP-addresses, but we found that the current implementation should be sufficient to protect against most attacks. However, there is still room for improvement for instance by adding a mandatory Captcha to the form, when correct would allow to submit the form.

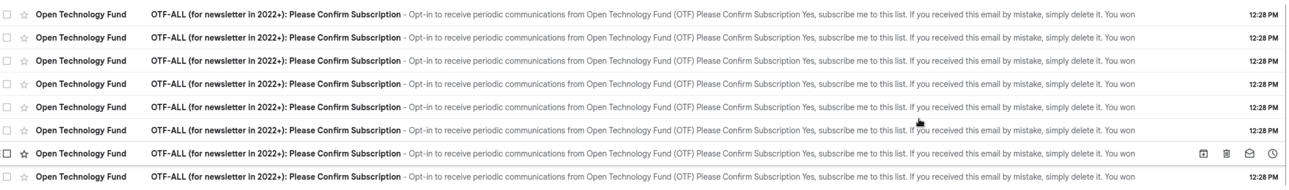
Example of some bruteforce attempts:



```
POST /wp-admin/admin-ajax.php HTTP/2
Host: beta.opentech.fund
Cookie: cf_clearance=7_ITvsAUg8oqMaeemFf0bereyyiZ21GwLsQtu2gUjjc-1689895230-0-250.0.0; cookieconsent=
decline; cmplz_banner-status=dismissed
Content-Length: 281
Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="114"
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0
Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://beta.opentech.fund
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://beta.opentech.fund/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

action=et_pb_submit_subscribe_form&et_frontend_nonce=0d8cb5f77b&et_list_id=f563cb73a4&et_firstname=Stefan&
et_lastname=Pentest&et_email=stefanpentest%2b13@gmail.com&et_provider=mailchimp&et_account=OpenTechFund&
et_ip_address=false&token=&et_checksum=20fd568d8878032924263977515100f2
```

```
HTTP/2 200 OK
Date: Fri, 21 Jul 2023 03:55:09 GMT
Content-Type: text/html; charset=UTF-8
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://beta.opentech.fund
Cache-Control: no-cache, must-revalidate, max-age=0
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Referrer-Policy: strict-origin-when-cross-origin
Strict-Transport-Security: max-age=15552000; includeSubDomains; preload
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Pantheon-Styx-Hostname: styx-fe1-b-7bb5bf4554-k2st8
X-Robots-Tag: noindex
X-Styx-Req-Id: 6238c077-277a-11ee-a2fe-0632f05c1a45
X-Served-By: cache-chi-kigq8000074-CHI, cache-mel11241-MEL
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1689911708.969148,VS0,VE1538
Vary: Accept-Encoding, Cookie, Cookie
Age: 0
Via: 1.1 varnish, 1.1 varnish
Cf-Cache-Status: DYNAMIC
Server: cloudflare
Cf-Ray: 7ea081aeab021f63-MEL
Alt-Svc: h3=":443"; ma=86400

{"error":"Subscription Error: You have signed up to a lot of lists very recently, please try again later"}
```

Note that after 8 requests we cannot conduct more anti-automation which limits this attack to 8 email addresses per IP-address:



☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won

☐ ☆ Open Technology Fund OTF-ALL (for newsletter in 2022+): Please Confirm Subscription - Opt-in to receive periodic communications from Open Technology Fund (OTF) Please Confirm Subscription Yes, subscribe me to this list. If you received this email by mistake, simply delete it. You won 12:28 PM

5.4 NF-009 — Wordpress - Xmlrpc is not enabled or accessible.

The XMLRPC functionality can be abused in several ways to target the website, for example performing a XSPA (Cross Site Port Attack) and login brute force attacks. We found that access to xmlrpc.php is not enabled or it is not accessible which is good.

6 Future Work

- **Retest of findings**

  When mitigations for the vulnerabilities described in this report have been deployed, a repeat test should be performed to ensure that they are effective and have not introduced other security problems.
- **Regular security assessments**

  Security is an ongoing process and not a product, so we advise undertaking regular security assessments and penetration tests, ideally prior to every major release or every quarter.

7 Conclusion

We discovered 1 High, 4 Moderate and 19 Low-severity issues during this penetration test.

After retesting, we are pleased to report that all significant issues have been successfully addressed. While the unresolved moderate and low-priority issues may not pose an immediate major risk, their resolution would significantly bolster our defenses against potential attacks on the application, infrastructure, and users. We strongly recommend addressing these issues as well.

After this we recommend to perform a retest in order to ensure that mitigations are effective and that no new vulnerabilities have been introduced.

Finally, we want to emphasize that security is a process – this penetration test is just a one-time snapshot. Security posture must be continuously evaluated and improved. Regular audits and ongoing improvements are essential in order to maintain control of your corporate information security. We hope that this pentest report (and the detailed explanations of our findings) will contribute meaningfully towards that end.

Please don't hesitate to let us know if you have any further questions, or need further clarification on anything in this report.

Appendix 1 Testing team

Stefan Vink	Stefan is an IT professional with a passion for IT security and automation. With 20 years hands-on experience in a diverse range of IT roles such as automation / scripting / monitoring / web development / system and network management in Windows and Linux environments. He has worked for organisations such as the Central Bank of the Netherlands (DNB), is MCITP, CCNA, LPIC, OSCP certified, and has passed the CISSP exam. He loves to travel, hike, play tennis & chess, automation, and lives with his family in Melbourne, Australia.
Melanie Rieback	Melanie Rieback is a former Asst. Prof. of Computer Science from the VU, who is also the co-founder/CEO of Radically Open Security.

Front page image by dougwoods (https://www.flickr.com/photos/deerwooduk/682390157/), "Cat on laptop", Image styling by Patricia Piolon, https://creativecommons.org/licenses/by-sa/2.0/legalcode.