Security report

SUBJECT
Security tests of the https://app.addy.io/ web application
Source code analysis

DATE
16.08.2023 – 29.08.2023

RETEST DATE
15.09.2023

LOCATION
Cracow (Poland)

AUTHOR
Dariusz Tytko

VERSION
1.1
Executive summary

This document is a summary of work conducted by Securitum. The subject of the test was the addy.io (AnonAddy) web application available at https://app.addy.io/. As part of the tests, the provided source code of the application was also analyzed.

Tests were conducted using the following roles: authenticated and unauthenticated user (visitor of the website).

During testing, no significant vulnerabilities were identified. Low-risk vulnerabilities were reported, as well as several informational points.

During testing, particular emphasis was placed on vulnerabilities that might in a negative way affect confidentiality, integrity or availability of processed data.

The security tests were carried out according to generally accepted methodologies, including: OWASP TOP10, (in a selected range) OWASP ASVS as well as internal good practices of conducting security tests developed by Securitum.

An approach based on manual tests (using the above-mentioned methodologies), supported by several automatic tools (i.a. Burp Suite Professional, SonarQube, ffuf), was used during the assessment.

The vulnerabilities are described in detail in further parts of the report.

**Status of the issues after retest (15.09.2023)**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Risk</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECURITUM-234116-001: The ability to exceed the limits set by application plans</td>
<td>LOW</td>
<td>Fixed</td>
</tr>
<tr>
<td>SECURITUM-234116-002: Username enumeration</td>
<td>LOW</td>
<td>Fixed</td>
</tr>
<tr>
<td>SECURITUM-234116-003: Changing the email address does not require re-authentication</td>
<td>INFO</td>
<td>Implemented</td>
</tr>
<tr>
<td>SECURITUM-234116-004: Enabling 2FA does not require re-authentication</td>
<td>INFO</td>
<td>Implemented</td>
</tr>
<tr>
<td>SECURITUM-234116-005: Disabling 2FA does not require using 2FA</td>
<td>INFO</td>
<td>Partially implemented</td>
</tr>
<tr>
<td>SECURITUM-234116-006: Generating a new backup code for 2FA does not require re-authentication</td>
<td>INFO</td>
<td>Implemented</td>
</tr>
<tr>
<td>SECURITUM-234116-007: The ability to recreate the state of the application generating one-time codes (TOTP)</td>
<td>INFO</td>
<td>Implemented</td>
</tr>
</tbody>
</table>
Risk classification

Vulnerabilities are classified on a five-point scale, that reflects both the probability of exploitation of the vulnerability and the business risk of its exploitation. Below, there is a short description of the meaning of each of the severity levels:

- **CRITICAL** – exploitation of the vulnerability makes it possible to compromise the server or network device, or makes it possible to access (in read and/or write mode) data with a high degree of confidentiality and significance. The exploitation is usually straightforward, i.e. an attacker does not need to gain access to the systems that are difficult to reach and does not need to perform social engineering. Vulnerabilities marked as 'CRITICAL' must be fixed without delay, mainly if they occur in the production environment.

- **HIGH** – exploitation of the vulnerability makes it possible to access sensitive data (similar to the ‘CRITICAL’ level), however the prerequisites for the attack (e.g. possession of a user account in an internal system) make it slightly less likely. Alternatively, the vulnerability is easy to exploit, but the effects are somehow limited.

- **MEDIUM** – exploitation of the vulnerability might depend on external factors (e.g. convincing the user to click on a hyperlink) or other conditions that are difficult to achieve. Furthermore, exploitation of the vulnerability usually allows access only to a limited set of data or to data of a lesser degree of significance.

- **LOW** – exploitation of the vulnerability results in minor direct impact on the security of the test subject or depends on conditions that are very difficult to achieve in practical manner (e.g. physical access to the server).

- **INFO** – issues marked as ‘INFO’ are not security vulnerabilities per se. They aim to point out good practices, the implementation of which will lead to the overall increase of the system security level. Alternatively, the issues point out some solutions in the system (e.g. from an architectural perspective) that might limit the negative effects of other vulnerabilities.
Statistical overview

Statistical overview after tests:

Additionally, 5 INFO issues are reported.

Statistical overview after retest (15.09.2023):
No vulnerabilities. 1 INFO issue is reported.
Contents

Security report ................................................................................................................................................. 1

Executive summary .............................................................................................................................................. 2
  Status of the issues after retest (15.09.2023)............................................................................................... 2
  Risk classification............................................................................................................................................ 3
  Statistical overview ......................................................................................................................................... 4

Change history ................................................................................................................................................... 6

Vulnerabilities in the web application .............................................................................................................. 7
  [FIXED][LOW] SECURITUM-234116-001: The ability to exceed the limits set by application plans........... 8
  [FIXED][LOW] SECURITUM-234116-002: Username enumeration ............................................................... 10

Informational issues ......................................................................................................................................... 11
  [IMPLEMENTED][INFO] SECURITUM-234116-003: Changing the email address does not require re-
  authentication ................................................................................................................................................ 12
  [IMPLEMENTED][INFO] SECURITUM-234116-004: Enabling 2FA does not require re-authentication 13
  [PARTIALLY IMPLEMENTED][INFO] SECURITUM-234116-005: Disabling 2FA does not require using 2FA
  .................................................................................................................................................................... 14
  [IMPLEMENTED][INFO] SECURITUM-234116-006: Generating a new backup code for 2FA does not
  require re-authentication ................................................................................................................................. 15
  [IMPLEMENTED][INFO] SECURITUM-234116-007: The ability to recreate the state of the application
  generating one-time codes (TOTP)................................................................................................................ 16
## Change history

<table>
<thead>
<tr>
<th>Document date</th>
<th>Version</th>
<th>Change description</th>
</tr>
</thead>
</table>
| 15.09.2023    | 1.1     | After the retest, the following information was added:  
  • “Status of the issues after retest” section in the executive summary.  
  • Statistical overview.  
  • “Status after retest section” for all issues. |
| 29.08.2023    | 1.0     | Creation of the report. |
Vulnerabilities in the web application
**[FIXED][LOW] SECURITUM-234116-001: The ability to exceed the limits set by application plans**

**STATUS AFTER RETEST**

The vulnerability has been fixed. During the retest, it was not possible to add a number of resources exceeding the set limit.

**SUMMARY**

It has been observed that the application lacks protection against race-condition attacks. As a result, in many places within the application, it is possible to exceed the limits imposed by application plans (e.g., one can add more rules than allowed by the purchased plan). The vulnerability is generic and applies to all limits in the application. The bypass of the rule limit presented in the PoC section should be treated as an example.

More information about race-condition vulnerability:

- [https://portswigger.net/web-security/race-conditions](https://portswigger.net/web-security/race-conditions)

**PREREQUISITES FOR THE ATTACK**

An account in the application.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

The following steps were taken to confirm the existence of the vulnerability:

1. The request below was sent 19 times to add 19 rules (limit of the rules was 20):

   ```
   POST /api/v1/rules HTTP/2
   Host: app.addy.io
   Cookie: [...] 
   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/116.0
   Accept: application/json, text/plain, */*
   Accept-Language: pl,en-US;q=0.7,en;q=0.3
   Accept-Encoding: gzip, deflate
   Referer: https://app.addy.io/rules
   X-Requested-With: XMLHttpRequest
   Content-Type: application/json
   X-Xsrf-Token: [...] 
   Content-Length: 221
   Origin: https://app.addy.io

   {"name":"Test","conditions":[{"type":"sender","match":"contains","values":["test"],"currentConditionValue":""}],"actions":[{"type":"subject","value":"test"}],"operator":"AND","forwards":true,"replies":false,"sends":false}
   ```

2. Then, the above request was sent an additional 5 times; the requests were sent simultaneously, using the Burp Suite Repeater tool.
3. As a result, it was possible to add three more rules (making a total of 22), thus exceeding the limit by 2 rules (the limit was 20).
The vulnerability arises from the fact that adding the resource (in this case, the rules) is not an atomic operation. First, the number of resources of a given type is checked, then the value is compared with the limit, and if the limit is not exceeded, a resource is added. If another HTTP request, adding a resource, is sent at a moment when the state (number of the resources) has not yet been updated, it is possible to add a resource even though, at the end of this operation, the limit will be exceeded.

The code responsible for adding a new rule is presented below. It can be observed that two significant steps - checking whether the limit has not been exceeded and the addition of the rule, are performed as two independent operations:

```php
public function store(StoreRuleRequest $request)
{
    // Add Limit for Rules
    if (user()->hasReachedActiveRuleLimit()) {
        return response('You\'ve reached your maximum rule limit', 403);
    }

    $conditions = collect($request->conditions)->map(function ($condition) {
        return collect($condition)->only(['type', 'match', 'values']);
    });

    $actions = collect($request->actions)->map(function ($action) {
        return collect($action)->only(['type', 'value']);
    });

    $rule = user()->rules()->create([...
```
**STATUS AFTER RETEST**

The vulnerability has been fixed. The CAPTCHA is validated first. As a result, automating the enumeration of usernames and user emails is more difficult.

**SUMMARY**

It has been noticed that the application allows one to check whether a given username and email are used in the application. The list of valid email addresses used in the application can be exploited to perform further attacks (e.g., sending phishing emails).

**PREREQUISITES FOR THE ATTACK**

None – anonymous access to the application.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

A vulnerability has been detected in the registration functionality. Entering an existing username or email address during registration returns an error indicating that the name is already taken. It is worth noting that despite the use of the CAPTCHA mechanism, the vulnerability can be exploited automatically. This is due to the fact that the correctness of the name and email is checked even if the CAPTCHA code is incorrect:

**LOCATION**

https://app.addy.io/register

**RECOMMENDATION**

It is recommended to make the automation of the enumeration attack more difficult. For this purpose, the CAPTCHA code should be verified before checking the correctness of the user’s data.
Informational issues
[IMPLEMENTED][INFO] SECURITUM-234116-003: Changing the email address does not require re-authentication

**STATUS AFTER RETEST**

The recommendation has been implemented. When changing the email address, it is necessary to provide the password.

**SUMMARY**

It has been noticed that the procedure for changing the email address does not require entering a password. As a result, if an attacker gains access to an active session, they will be able to change the user’s password by changing the email address, and then using the password reset mechanism. From a security perspective, changing the email address should be treated in a similar manner to the password changing procedure, which requires entering the current password.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

During testing, the email address was changed without the need to enter a password (Settings -> General -> Update Email).

**LOCATION**

Email change mechanism.

**RECOMMENDATION**

It is recommended that entering a password be required when changing the email address.
[IMPLEMENTED][INFO] SECURITUM-234116-004: Enabling 2FA does not require re-authentication

**STATUS AFTER RETEST**

The recommendation has been implemented. During 2FA activation, it is necessary to provide the password.

**SUMMARY**

It has been noticed that enabling 2FA does not require entering a password. As a result, if an attacker gains access to a user’s session, they can activate the 2FA mechanism, thereby blocking the user’s access to the account.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

Enabling 2FA only requires confirmation with a generated code:

![Enable Authentication App (TOTP)](image)

A similar situation also occurs for hardware keys.

**LOCATION**

2FA management (TOTP and hardware keys).

**RECOMMENDATION**

To enable 2FA, one should be required to enter a password.
[PARTIALLY IMPLEMENTED][INFO] SECURITUM-234116-005: Disabling 2FA does not require using 2FA

**STATUS AFTER RETEST**

The recommendation has been partially implemented. When disabling hardware 2FA, it is necessary to provide the password.

**SUMMARY**

It has been noticed that when disabling 2FA (TOTP), only a password is required. From a security improvement perspective, it is recommended that a user disabling 2FA should prove that they have access to 2FA. For this purpose, in addition to entering a password, it should be required to use a 2FA or a backup code.

It should be noted that when turning off the hardware key, there is no need to enter even a password.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

Disabling 2FA requires only entering a password:

![Disable Authentication App (TOTP)
To disable two-factor authentication enter your password below. You can always enable it again later if you wish.

<table>
<thead>
<tr>
<th>Current Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>·················</td>
</tr>
</tbody>
</table>

![Disable Two-Factor Authentication]

Disabling hardware key does not require entering even a password:

![Remove Hardware Key
Once this key is removed, Two-Factor Authentication will be disabled on your account.

Remove Close]

**LOCATION**

2FA management (TOTP and hardware keys).

**RECOMMENDATION**

To disable 2FA, one should be required to enter a password and use either a 2FA or a backup code.
[IMPLEMENTED][INFO] SECURITUM-234116-006: Generating a new backup code for 2FA does not require re-authentication

**STATUS AFTER RETEST**

The recommendation has been implemented. When generating a new backup code, it is necessary to provide the password.

**SUMMARY**

It has been noticed that generating a backup code for 2FA does not require entering a password or using 2FA. From the perspective of enhancing the security of this solution, it is recommended that generating a new code should require re-authentication using a password and optionally 2FA.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

Generating a backup code does not require re-authentication:

---

**Generate New Backup Code**

The backup code can be used in a situation where you have lost your 2FA device to allow you to access your account. If you’ve forgotten or lost your backup code then you can generate a new one by clicking the button below. This code will only be displayed once so make sure you store it in a secure place. If you have an old backup code saved you must update it with this one.

---

**LOCATION**

2FA backup code.

**RECOMMENDATION**

It is recommended that generating a new backup code should require re-authentication using a password and optionally 2FA.
**STATUS AFTER RETEST**

The recommendation has been implemented. Information that allowed the recreation of the application’s state, which generates one-time codes, is no longer revealed.

**SUMMARY**

It has been noticed that after enabling the 2FA mechanism (TOTP), anyone who gains access to an active session can recreate on their device the state of the application generating one-time codes, thereby gaining the ability to bypass the 2FA mechanism during future logins. The entire process is undetectable to the account owner – after the attack, there will be two applications generating the same codes.

**TECHNICAL DETAILS (PROOF OF CONCEPT)**

If 2FA is enabled, the application’s interface does not allow registering a new 2FA device. Only disabling 2FA is possible:

Two-Factor Authentication

Two-factor authentication, also known as 2FA or multi-factor, adds an extra layer of protection to your account beyond your username and password. There are two options for authentication: (e.g., Google Authenticator or another, Aegis, and OTP) or a key (e.g., YubiKey, SoloKey, Nitrokey).

When you login with 2FA enabled, you will be prompted to use a security key (time passcode) depending on which method you choose below. You can only enable 2nd factor authentication enabled at once.

Generate New Backup Code

The backup code can be used in a situation where you have lost your 2FA device or need to access your account. If you’ve forgotten or lost your backup code then you can get a new one by clicking the button below. This code will only be displayed once so store it in a secure place. If you have an old backup code saved you must update it.

![](Generate New Backup Code)

Disable Authentication App (TOTP)

To disable two-factor authentication enter your password below. You can always disable it later if you wish.

Current Password

![Disable Two-Factor Authentication]

However, it has been noted that in response to a request to `GET /settings/security`, data is returned that allows to recreate the state of the application generating one-time codes (authentication secret and QR code):
LOCATION

https://app.addy.io/settings/security

RECOMMENDATION

It is recommended that after enabling 2FA, the application should not return data that allows for the recreation of the state of the application generating one-time codes.