

URnetwork Penetration Test Report - May 2025

Report generated on May 12 2025

Prepared for BringYour, Inc.. For informational purposes only and may not be relied upon for any other purpose. Report is intended for internal use by BringYour, Inc. only. Cobalt disclaims all liability to any third-party arising from this report. Usage of this report by shall be subject to Cobalt's terms, available at <https://cobalt.io/terms/>.

Targets

https://ur.io

2001:470:173::51

2001:470:173::52

2001:470:173:52:e643:4bff:fe23:a341

2001:470:173::54

2001:470:99::56

2001:470:99::57

2001:470:99:57:e643:4bff:fe23:a343

2001:470:99::58

65.19.157.32/27

2001:470:173::/48

65.49.70.64/27

2001:470:99::/48

185.217.1.200/29

2a0b:c041:8::1/48

Test period**Status**

Apr 25, 2025 May 5, 2025

Final

Test performed by

Sebastian Mihalache

Coordinator

Contents

Executive Summary4

Approach..... 5

 Risk Factors 6

 Severity Definitions 6

Post-Test Remediation.....8

Terms.....9

Appendix A - Finding Details..... 10

Executive Summary

Cobalt conducted a pentest of the URnetwork application and API to assess its risk posture and identify security issues that could negatively affect BringYour, Inc.'s data, systems, or reputation. The scope of the assessment covered URnetwork and included credentials for various levels of privilege within the scope. A Cobalt pentest team of 1 conducted this engagement between Apr 25, 2025 and May 5, 2025.

The web application pentest was a manual assessment of the security of the application's functionality, business logic, and vulnerabilities, such as those cataloged in the [Open Web Application Security Project \(OWASP\) Top 10](#). The assessment also included a review of security controls and requirements listed in the [OWASP Application Security Verification Standard \(ASVS\)](#).

The pentesters leveraged tools to facilitate their work. However, the majority of the assessment involved manual analysis.

Approach

The engagement was done according to industry best practices. The following outlines the process from start to finish.

Pre Engagement

- Scoping
- Customer documentation
- Access

Engagement

- Reconnaissance
- Tool assisted assessment
- Manual assessment
- Vulnerability identification and/or exploitation
- Risk analysis
- Reporting

Post Engagement

- Prioritized remediation
- Recommendations
- Retesting (if applicable)

Risk Factors

Each finding is assigned two factors to measure its risk. Factors are measured on a scale of 1 (very low) through 5 (very high).

Impact

This indicates the finding's effect on technical and business operations. It covers aspects such as the confidentiality, integrity, and availability of data or systems; and financial or reputational loss.

Likelihood

This indicates the finding's potential for exploitation. It takes into account aspects such as skill level required of an attacker and relative ease of exploitation.

Severity Definitions

When our pentesters find vulnerabilities, they use the standard [OWASP Risk Rating Methodology](#), and then classify them into one of the following risk levels, based on their business impact and likelihood: $\text{risk} = \text{impact} * \text{likelihood}$

● CRITICAL

Includes vulnerabilities that require immediate attention. Risk score of 25.

● HIGH

Impacts the security of your application/platform/hardware, including supported systems. Includes high probability vulnerabilities with a high business impact. Risk score range: 16 through 24.

● MEDIUM

Includes vulnerabilities that are: medium risk, medium impact; low risk, high impact; high risk, low impact. Risk score range: 5 through 15.

● LOW

Specifies common vulnerabilities with minimal impact. Risk score range: 2 through 4.

● INFORMATIONAL

Notes vulnerabilities of minimal risk to your business. Risk score of 1.

Post-Test Remediation

All identified findings are below with their mitigation status.

Finding ▼	Type	Severity ▼	State ▼	Resolved
#PT30328_2	Server Security Misconfiguration	● LOW	Fixed	06 May 2025
#PT30328_3	Server Security Misconfiguration	● LOW	Accepted Risk	07 May 2025
#PT30328_4	Server Security Misconfiguration	● LOW	Fixed	06 May 2025
#PT30328_1	Server Security Misconfiguration	● INFO	Fixed	06 May 2025

Accepted Risk Reasons

Finding	Accepted Risk Reason
#PT30328_3	Intended functionality Limited to 10 probes per 5 minutes. Usability tradeoff.

Terms

PLEASE NOTE: It is impossible to test networks, information systems, and people for every potential security vulnerability. This report does not form a guarantee that your assets/targets are secured from any and all threats. All assessments performed, and their results, are only from the point-of-view of Cobalt, at the time of the engagement. Cobalt is unable to ensure or guarantee that your assets/targets are or will be completely safe from every form of attack now or in the future. With the ever-changing environment of information technology, any assessment performed by Cobalt will necessarily exclude vulnerabilities in software or systems that are unknown at the time of the engagement. For a full list of terms governing the services of Cobalt, this report, and the usage thereof, please consult the Terms of your Agreement with Cobalt or www.cobalt.io/terms.

Finding Details

Misconfigured DMARC Records for Email Domain

**ur.io**

Fixed



● LOW

• Reported by [jondow](#) on Apr 27, 2025 • #PT30328_2

Vulnerability Type

Server Security Misconfiguration > Mail Server Misconfiguration > Email Spoofing to Inbox due to Missing or Misconfigured DMARC on Email Domain

Description

SPF - Sender Policy Framework - <http://www.openspf.org/> - is a simple addition you can make to your DNS servers to allow recipients to authenticate email messages you send. After you're SPF-Enabled, any phishing emails that attempt to spoof your legitimate email domain will be erased by all good anti-spam software, thus preventing victims from ever receiving the phish emails.

The DMARC policy allows sender domains to state that SPF and DKIM email protections are implemented. These can allow the receiver domain to retest or quarantine an email from a sender domain should either SPF or DKIM fail

Affected Resources

<https://ur.io>

OWASP Severity

● LOW

CVSS v3.1 Score

● MEDIUM (5.4)

Proof of Concept

- Check for DMARC Records for the domain
 - `ur.io`
using the following command and observe that the
 - `p=none`
flag is set:

```
dig txt _dmarc.ur.io
```

```
$ dig txt _dmarc.ur.io

;; <<>> DiG 9.10.6 <<>> txt _dmarc.ur.io
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5102
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;_dmarc.ur.io.                IN      TXT
;; ANSWER SECTION:
_dmarc.ur.io.                300     IN      TXT      "v=DMARC1; p=none; rua=mailto:rua@dmarc.brevo.com"
;; Query time: 84 msec
;; SERVER: 10.247.0.1#53(10.247.0.1)
;; WHEN: Sun Apr 27 16:13:38 EEST 2025
;; MSG SIZE  rcvd: 114
```

- Using a fake email website like
 - `emkei.cz`
, an attacker can send a fake email from the vulnerable domain:

FAKE'S MAILER

Free online fake mailer with attachments, encryption,
HTML editor and advanced settings...

From Name: Cobalt

From E-mail: Admin@ur.io

To: smihalache@cobaltcore.io

Subject: Important information

Attachment: Răsfoiește... Niciun fișier selectat.
Attach another file
Advanced Settings

Content-Type: ☒ text/plain ☐ text/html ☐ Editor

Text: Click the following link!

Captcha:

- The fake email lands in victim's inbox as shown

Important information

External

Inbox x

**Cobalt** <Admin@ur.io>

to me ▼

Click the following link!

Impact

This impacts any users who trusts emails from the

`ur.io`

domain. This issue is easy to exploit and successful attacks can lead to phishing or spoofing the users in order to obtain credentials or sending malicious documents, etc. An attacker can entice the victim in clicking on phishing links sent via a genuine email sender. The user would believe that the email is coming from a genuine email as the source is

`ur.io`

domain.

Suggested Fix

SPF is not a sufficient email spoofing protection in case of some of the largest email providers. Emails spoofed for domains having properly configured hard fail SPF records may still be delivered to the recipient's inbox. In order to fully prevent email spoofing create a DMARC record with

`p=reject`

policy. Please note that if your DMARC policy is not set up properly it may result in email delivery issues.



Username enumeration in Login functionality

Accepted Risk



● LOW

• Reported by [jondow](#) on Apr 29, 2025 • #PT30328_3

Vulnerability Type

Server Security Misconfiguration > Username/Email Enumeration

Description

Username enumeration is the process of iterating through a list of potential usernames to determine which are valid by comparing application responses. These differences in responses could be part of verbose error messages, differences in the contents of a response, or differences in the server's timing for responses.

An attacker could use any factor that differentiates a valid response from an invalid one to compile a list of valid usernames to target during further attacks, such as password guessing, brute-force, or attempting to exploit a reused password from an existing breach.

References

- [OWASP's Page on Testing for Account Enumeration and Guessable User Account](#)

Affected Resources

<https://api.bringyour.com/auth/login-with-password>

OWASP Severity

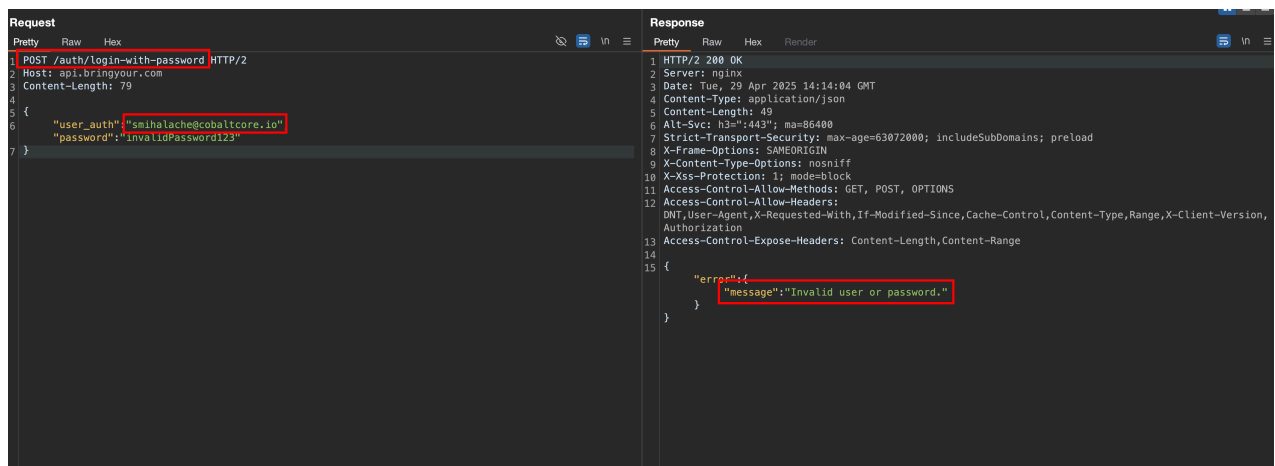
● LOW

CVSS v3.1 Score

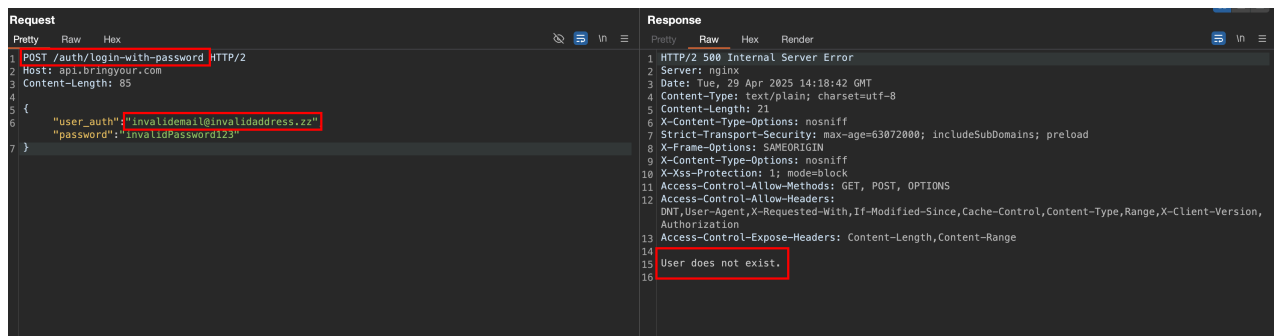
● MEDIUM (5.3)

Proof of Concept

- Send a login request to the
- `https://api.bringyour.com/auth/login-with-password` endpoint using a valid email address:



- Now enter an invalid email and observe a different response:



Impact

An attacker might abuse this vulnerability to discover valid user email addresses and try to perform other complementary attacks such as password brute-force or social engineer.

Suggested Fix

- Configure the web application so that it does not reveal information about whether accounts are valid. Return generic error messages that respond to all requests with a message saying that account reset instructions were sent by email, regardless of whether that account exists.

HTTP Request

```
POST /auth/login-with-password HTTP/2
Host: api.bringyour.com
Content-Length: 85

{
  "user_auth":
  "invalidemail@invalidaddress.zz","password":"invalidPassword123"
}
```




Missing Rate Limiting allows Password Brute Force

Fixed



● LOW

• Reported by [jondow](#) on Apr 29, 2025 • #PT30328_4

Vulnerability Type

Server Security Misconfiguration > No Rate Limiting on Form > Login

Description

Some applications employ functionalities, such as “Login” or “Forgot Password” mechanisms, that allow users to submit requests through unauthenticated forms. If the request page allows unauthenticated users to submit a series of consecutive form submissions, an attacker could misuse this functionality to overwhelm the service or a user’s inbox with requests to consume company resources or create a Denial-of-Service (DoS) condition.

References

- [OWASP's Page on Blocking Brute-Force Attacks](#)

Affected Resources

<https://api.bringyour.com/auth/login-with-password>

OWASP Severity

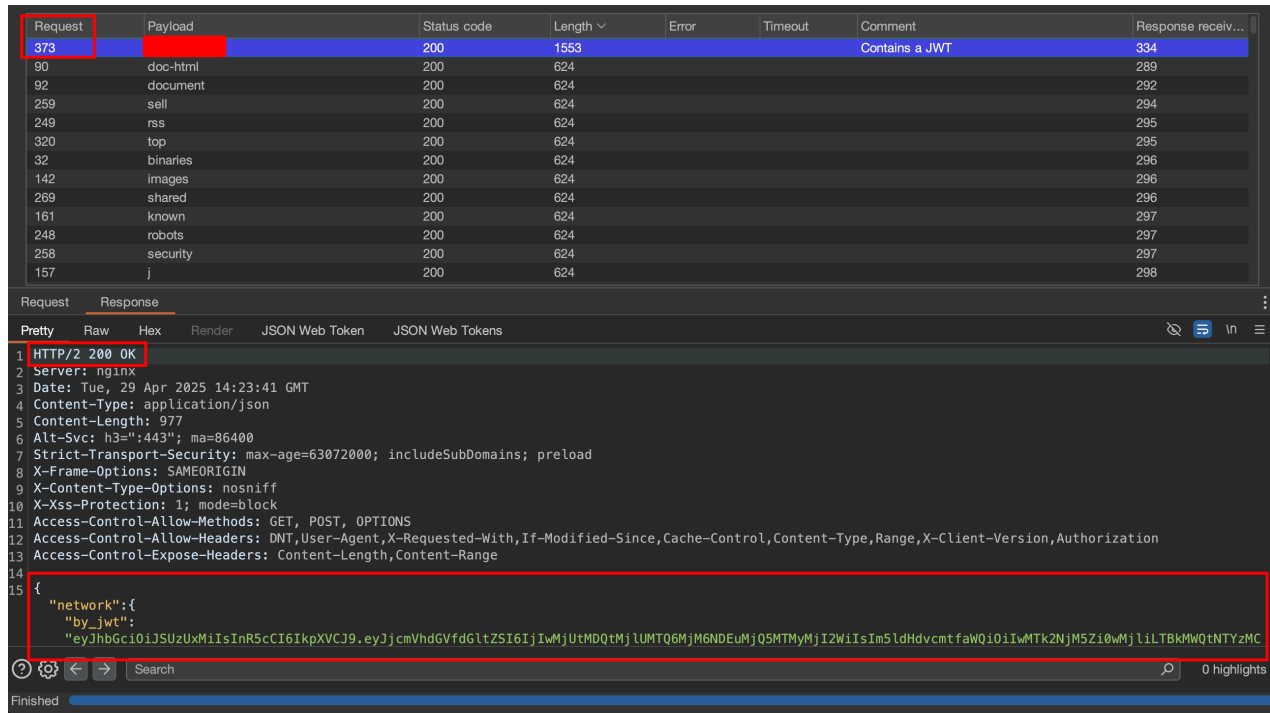
● LOW

CVSS v3.1 Score

● MEDIUM (5.3)

Proof of Concept

- Send a login request to the application and intercept it using a proxy such as BurpSuite.
- We can perform a password brute-force attempt by changing the password parameter and sending multiple login requests for each password:



- As can be seen in the image above, after 373 consecutive login requests, we can still obtain the JWT due to the fact that the API has no rate limiting present on the login request.

Impact

This issue can be abused to send hundreds of requests in a few seconds, thus performing a password brute-force attack on valid user email addresses.

Suggested Fix

- Implement CAPTCHA controls to validate that requests come from people, rather than automated tools. Consider also implementing appropriate rate-limiting, such as IP address, time, or email-based limitations, to ensure that individual sources cannot request forms and submissions more frequently than the servers can reasonably handle.

HTTP Request

```
POST /auth/login-with-password HTTP/2
```

```
Host: api.bringyour.com
```

```
Content-Length: 79
```

```
{  
  "user_auth":  
  "smihalache@cobaltcore.io", "password": "invalidPassword123"  
}
```

Server Banner Disclosure



Fixed



● INFORMATIONAL

• Reported by [jondow](#) on Apr 27,

2025 • #PT30328_1

Vulnerability Type

Server Security Misconfiguration > Fingerprinting/Banner Disclosure

Description

If you are running a web server, it often shows the world what type of server it is, its version number, and the operating system. This information is available in header fields and can be acquired using a web browser to make a simple HTTP request to any web application. It is often called the web server banner and is ignored by most people with the exception of malicious ones.

Attackers can perform banner grabbing using even simple TCP tools like telnet or netcat. Then they launch targeted attacks against your web server and version. In addition, if a particular web server version is known to be vulnerable to a specific exploit, the attacker would just need to use that exploit as part of their assault on the target web server.

Affected Resources

<https://65.19.157.34/>

OWASP Severity

● INFORMATIONAL

CVSS v3.1 Score

● MEDIUM (5.6)

Proof of Concept

- Make a HTTP request to the
- `https://65.19.157.34/`
- URL
- Observe the HTTP Header response that leaks the
- `nginx/1.18.0 (Ubuntu)`
- Server Banner:

The screenshot shows a web browser's developer tools with the 'Network' tab selected. A request to `https://65.19.157.34/` is highlighted. The 'Response' pane shows the HTTP response headers, including `Server: nginx/1.18.0 (Ubuntu)`, which is the server banner.

```

Request
1 GET / HTTP/1.1
2 Host: 65.19.157.34
3 Sec-Ch-Ua: "Chromium";v="135", "Not-A.Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "macOS"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16 Connection: keep-alive

Response
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 27 Apr 2025 13:01:58 GMT
4 Content-Type: text/html
5 Connection: keep-alive
6 Content-Length: 175694
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta charset="utf-8">
12 <title>
13 Wikimedia dumps last five Mirror
14 </title>
15 <meta name="viewport" content="width=device-width, initial-scale=1">
16 <link rel="preconnect" href="https://fonts.gstatic.com">
17 <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;400;700&display=swap" rel="stylesheet">
18
19 <body>
20 <div class="webkit-text-size-adjust-100% overflow-wrap-break-word font-family-'Cairo', sans-serif font-size-12pt margin-10px">
21
22 <div class="status-bar padding-24px background-color-rgb(238,238,238)">
23
24 <div class="status-bar-code background-color-rgb(0,0,0) color-rgb(255,255,255) padding-4px user-select-all">
25
26 <div class="status-bar-div margin-bottom-12px">
27

```

Impact

The disclosed information might be used by attackers to identify vulnerabilities or weaknesses specific to the disclosed software versions.

Suggested Fix

As a best practice, configure your web server to prevent information leakage from the Server header.

For more information please refer to:

<https://techcommunity.microsoft.com/t5/iis-support-blog/remove-unwanted-http-r esponse-headers/ba-p/369710>