



EXPERT REPORT

on the results of the security audit
of the Cindicator company's
infrastructure

Table of Contents

1	Introduction	3
	General provisions	3
	Generally accepted abbreviations	3
	Abstract	3
2	Security Audit principles	4
	IS threats	4
	Attacker model	4
	External attacker	4
	The scope of the infrastructure security audit	5
3	Attack scenarios	6
	Getting access to the Administrator's control panel	6
4	External security audit of the System	7
	Discovered vulnerabilities	7
	E-mail functionality abuse	7
	Unsafe Flask configuration	8
	Weak service password	9
	Debugging information output	9
	No protection from Clickjacking	10
	Reflected XSS	11
	Unsafe CORS configuration	12
	Information disclosure	13
5	Conclusion	15
6	License	16
	Appendix 1. Security assessment	17
	Security level analysis	17
	Vulnerability severity	17
	Ease of vulnerability exploitation	18
	Vulnerability accessibility	18
	Likelihood of exploitation	19
	Vulnerability impact	19
	Appendix 2. Security level analysis.	20
	List of the discovered vulnerabilities and security weaknesses	20

Introduction

This expert report contains the results of security audit and penetration testing of the main infrastructural websites (hereinafter referred to as “the System”) owned by Cindicator (hereinafter referred to as “the Company”) with recommendations on the current security level improvement.

Abbreviation	Meaning
DBMS	Database management system
DB	Data base
IS	Information security
DS	Data system
CIN	Corporate information network
FW	Firewall
SW	Software

Table.1 Generally accepted abbreviations

Cyber3’s experts conducted a security assessment of the System. In the scope of work, they used and examined an external attacker models.

The performed works indicated that:

1. The current security level of the System could be rated “Average”.
2. Most vulnerabilities discovered in the critical segments of the System have “average” and “low” impact levels.

The list of key recommendations is as follows:

3. Fix the discovered vulnerabilities.

Below, you can find the detailed description of the discovered security weaknesses and the risks associated with them as well as the list of recommendations on how to fix them.

Security Audit principles

There are 3 types of IS threats that can affect a Company's information resources: violations of confidentiality, integrity or availability of information.

Confidentiality violations are usually aimed at information disclosure. If a violation is successful, the information becomes known to people, who should not have access to it: unauthorized personnel of the Company, clients, partners, competitors, and third parties.

Integrity violations are aimed at modification or corruption of the information, which can lead to modification of its structure or content, and to complete or partial destruction of the data.

An availability violation (denial-of-service threat) involves data system users' inability to access the information.

The core principle of this IS audit is an estimation of exploitation likelihood of the aforementioned threats affecting the Company's information resources, within the framework of a pre-defined attacker model.

A potential attacker is an individual or a group of individuals, acting either in collusion or independently, whose intended or unintended actions can carry the aforementioned threats to the IS, infringe information resources of the System or negatively affect the Company's interests.

The IS threats are basic threats to confidentiality and integrity of information and the threat of the System denial-of-service.

Attackers can pursue the following goals (and their possible combinations):

- cause Denial of Service;
- escalate their privileges in the System;
- get unauthorized access to business-critical data.

In the scope of the security audit, the experts used an external attacker model.

The following submodel of an external attacker model was used to conduct the security audit:

- an external attacker from the Internet who, as a client, has access to the Company's website and has neither knowledge about the System nor privileges in it.

The scope of the infrastructure security audit

Table 2. List of the audited resources

Resource	Description
cindicator.com	
tokensale.cindicator.com	
40.71.xx.xx	main proxy
40.114.xx.xx	telegram bot
13.82.xx.xx	trading bot
13.82.xx.xx	admin
13.92.xx.xx	admin test
40.71.xx.xx	production site
40.71.xx.xx	stage site
52.170.xx.xx	backend-v2.6
52.170.xx.xx	backend-v2.7
40.71.xx.xx	backend-v2.8
52.168.xx.xx	backend-v2.9
13.82.xx.xx	backend-v2.10
52.170.xx.xx	backend-v2.11
52.168.xx.xx	backend-test
52.168.xx.xx	postgres-db
52.179.xx.xx	sentry
40.71.xx.xx	maintenance
52.170.xx.x	signals
40.71.xx.xx	graphite + grafana
52.170.xx.xx	mongodb
13.68.xx.xx	main-deployment-server
40.71.xx.xx	elk stack + kibana

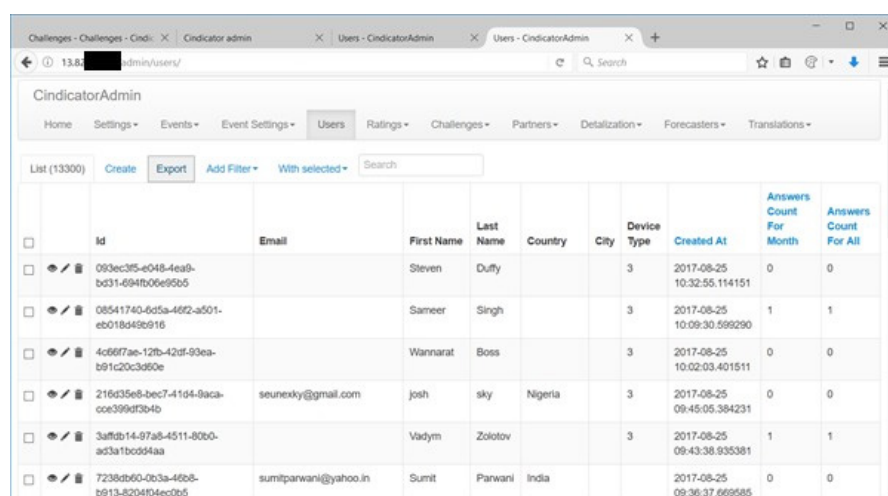
Getting access to the Administrator's control panel

Attack scenarios

Since the test server (13.92.X.X) of the Administrator's control panel has a user account with a dictionary password ([see p.9](#)), an attacker can gain access and examine its capabilities. The username of the account can be easily obtained from various sources:

- the "Team" page of the cindicator.com website
- the authorization webpage of the Sentry service (52.179.xx.xx)

Because the testing and product environments use the same `secret_key` field of the Flask framework, an attacker can use the current session in the testing environment to access the very same Administrator's account in the product environment ([see p.8](#)).



The screenshot shows the CindicatorAdmin web interface. The top navigation bar includes links for Home, Settings, Events, Event Settings, Users (active), Ratings, Challenges, Partners, Detailization, Forecasters, and Translations. Below the navigation bar, there is a search bar and a table of users. The table has columns for Id, Email, First Name, Last Name, Country, City, Device Type, Created At, Answers Count For Month, and Answers Count For All. The table lists six users with their respective details.

	Id	Email	First Name	Last Name	Country	City	Device Type	Created At	Answers Count For Month	Answers Count For All
<input type="checkbox"/>	093ec3f5-e048-4ea9-bd31-694fb06e9905		Steven	Duffy			3	2017-08-25 10:32:55.114151	0	0
<input type="checkbox"/>	08541740-6d5a-4662-a501-eb018d49b916		Sameer	Singh			3	2017-08-25 10:09:30.599290	1	1
<input type="checkbox"/>	4c66f7ae-12fb-42df-93ea-b91c20c3d60e		Wannarat	Boss			3	2017-08-25 10:02:03.401511	0	0
<input type="checkbox"/>	216d35e8-bec7-41d4-8aca-cc0c99d73b4b	seunexky@gmail.com	josh	sky	Nigeria		3	2017-08-25 09:45:05.384231	0	0
<input type="checkbox"/>	3affdb14-97a8-4511-80b0-ad3a1b0d94aa		Vadym	Zolotov			3	2017-08-25 09:43:38.935381	1	1
<input type="checkbox"/>	7238db60-0b3a-46b8-b913-8204f04ec0b5	sumitparwani@yahoo.in	Sumit	Parwani	India			2017-08-25 09:36:37.669585	0	0

Fig 1. Administrator's control panel (13.82.xx.xx)

4

Discovered
vulnerabilities

E-mail
functionality
abuse

External security audit of the System



Severity:
high



Likelihood
of exploitation:
average



Overall
impact level:
high

Description:

Due to insufficient input processing, an attacker can abuse e-mail sending functionality.

Impact:

An attacker can use the vulnerability to conduct attacks by social engineering and abuse the functionality to send undesired e-mails, which may lead to a legitimate e-mail address getting to a spam blacklist.

Vulnerable resource:

- cindicator.com

Technical details:

All available registration and subscription forms are susceptible to the vulnerability (as a URL path):

- /ico_adduser
- /ts_appliance
- /telegram_bot_register

The e-mail field is not duly checked. Therefore, it is possible to add other recipients for a generated e-mail.

There is also no check for an e-mail being actually registered. That is why an attacker can perform spamming until the cindicator.com domain is added to a spam list.

Request example:

```
POST /ico_adduser HTTP/1.1
```

```
Host: cindicator.com User-Agent: Mozilla/5.0 (Windows  
NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0
```

```
Accept: application/json, text/plain, */* Accept-  
Language: en-US,en;q=0.5 Referer: https://cindicator.  
com/
```

```
Content-Type: application/json;charset=utf-8 Content-  
Length: 44
```

Unsafe Flask configuration

```
Cookie: i18next=en-US; _ga=GA1.2.1342928565.1503323479;
_ym_uid=1503323480728693018; _gid=
GA1.2.1672955713.1503652627; _gat=1; _ym_isad=2
Connection: close

{«email»:»VICTIM@email.spam»,»notification»:true}
```

Recommendations:

- Validate input parameter



Severity:
high



Likelihood
of exploitation:
average



Overall
impact level
high

Description:

An attacker can use the vulnerability to escalate the privileges or bypass authentication.

Impact:

The misconfiguration of the Flask framework provides an attacker with an opportunity to bypass authorization mechanisms.

Vulnerable resource:

- 13.82.xx.xx
- 13.92.xx.xx

Technical details:

In the Flask framework, there is the same SECRET_KEY value in both the product and test environment. This value is used as a secret one during session generation. Consequently, if attackers get an account in the test environment (session) even not knowing SECRET_KEY, they will get an account in the product environment as well.

Recommendations:

- Use the SECRET_KEY value resistant to bruteforce.
- Use different SECRET_KEY values for the test and product environments.
- Do not write any authentication data (or data related to the authorization process in any way) in the source code of the application.

Weak service password



Severity:
high



Likelihood
of exploitation:
high



Overall
impact level
high

Description:

A dictionary password is used to access the service.

Impact:

An attacker can get unrestricted access to the Administrator's test panel. With the information during the examination of the testing environment, an attacker can attack the product environment more effectively.

Vulnerable resource:

- 13.92.xx.xx

Technical details:

There is a dictionary password set for the Administrator's account - X@cindicator.com. The auditors managed to use it to access the administering panel of the testing environment.

Recommendations:

- Set a strong password.

Debugging information output



Severity:
average



Likelihood
of exploitation:
average



Overall
impact level
average

Description:

In the System, debugging information output is allowed in case an operation failure occurs.

Vulnerable resource:

- cindicator.com
- 13.82.X.X

Technical details:

In case of exception conditions during the processing i18next parameter in the Cookie HTTP header on the cindicator.com website, debugging information is output. In addition, the output also occurs if the "e-mail" parameter in the POST request is processed.

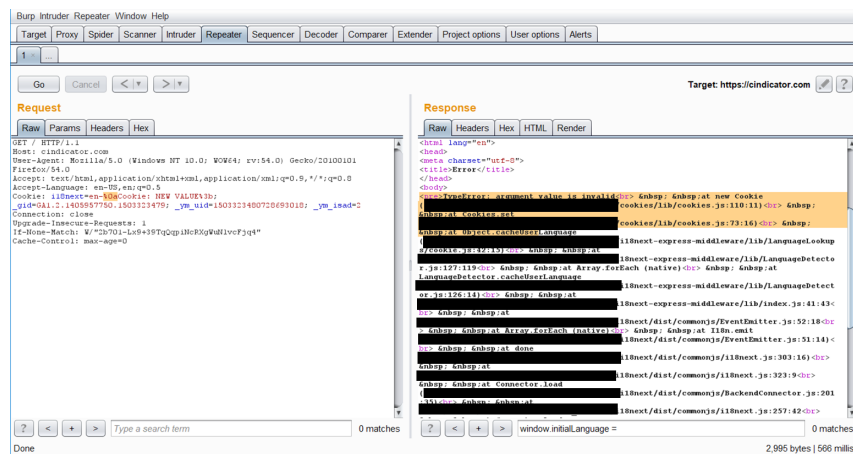


Fig 2. Debugging information output on cindicador.com

If an exception condition occurs in the Administrator's panel (13.82.X.X) while the POST request to /admin/admins/action is processed, debugging information is output.

Recommendations:

- Disable debugging information output

No protection
from Clickjacking



Severity:
average



Likelihood
of exploitation:
average



Overall
impact level
high

Description:

The web server does not use X-Frame-Options.

Impact:

The website of the System loaded in the frame on an attacker's resource may be used to launch phishing and clickjacking attacks on users and to inconceivably manipulate with the Systems. For example, a user is on an attacker's website and performs some actions there (clicks, text entering). These actions will be sent to the System and processed there (not on an attacker's website) since the site and the System are displayed on different layers.

Vulnerable resource:

- cindicador.com

Reflected XSS

Technical details:

cindicator.com and its subdomains (including those used for the ICO) are susceptible to clickjacking attacks. By exploiting this pattern of the System behavior, an attacker can tamper with the information displayed for a user on the website (e.g., Ethereum address of a contract during the ICO).

Recommendations:

Add the X-Frame-Options header with the "DENY" or "SAMEORIGIN" value in all headers of web servers responses.



Severity:
average



Likelihood
of exploitation:
low



Overall
impact level
low

Description:

An attacker can launch reflected XSS attacks on users of the System.

Impact:

If an attack is successful, an attacker will be able to get complete control of what is displayed in a user's browser and what is sent to the server. It will also be possible to emulate a user's actions on the website.

Vulnerable resource:

- cindicator.com

Technical details:

The validation of the i18next parameter of the Cookie header is insufficient. If attackers find a way to control Cookie, it will also be possible for them to exploit the vulnerability.

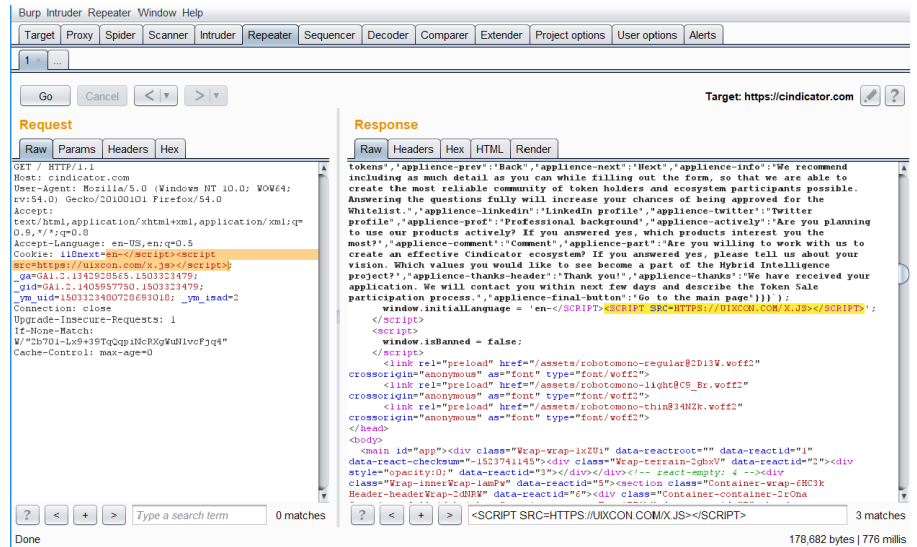


Fig 3. Request example

Recommendations:

Filter and screen all passed parameters.

Unsafe CORS configuration



Severity:
average



Likelihood
of exploitation:
low



Overall
impact level
low

Impact:

Incorrect configuration of the web server with the enabled CORS (Cross Origin Resource Sharing) enables an attacker to gain access to a website user's sensitive data.

Vulnerable resource:

- tokensale.cindicator.com

Technical details:

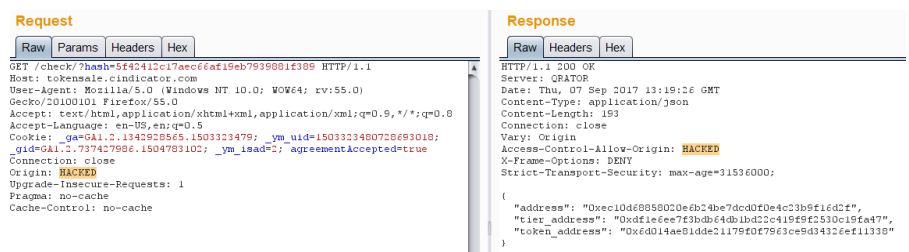


Fig 4. Insufficient processing of the Origin HTTP header

- Set a whitelist of domains that are allowed to call a target domain.
- Check entry values of the Origin header in the whitelist.



Severity:
low



Likelihood
of exploitation:
average



Overall
impact level
low

Description:

The source maps mechanism enables an attacker to easily debug the product-environment security flaws.

Impact:

An attacker can obtain all the information about the technologies used on the client side and to obtain the client-side source code of the application (before minification). Consequently, it will be easier for an adversary to find vulnerabilities.

Vulnerable resource:

- tokensale.cindicator.com

Technical details:

If not disabled, the source maps mechanism allows getting the source code of the application client side before minification.

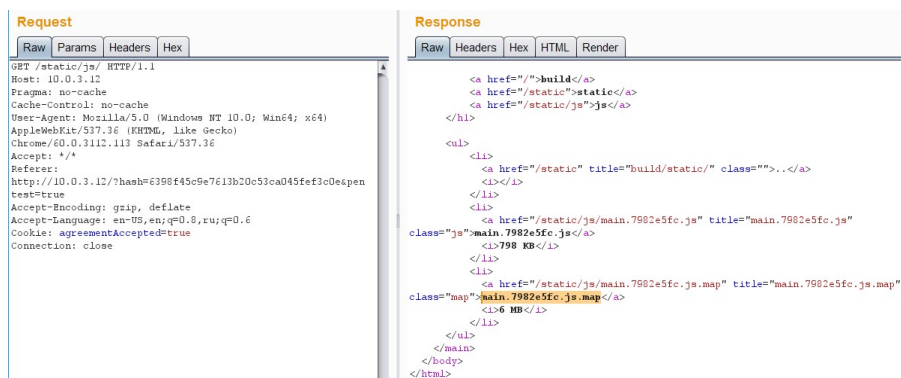


Fig 5. Call to the folder that contains the source map file

A browser processes the main.7982e5fc.js.map file and adds the information regarding the source code of the client side of the applications.

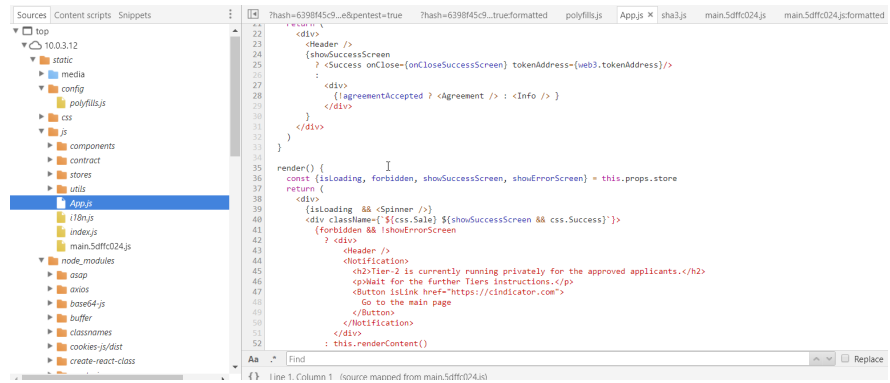


Fig 6. The source map technology in a browser

Recommendations:

- Disable source maps

Conclusion

As a result of the security audit, the experts revealed the feature of the configuration and deployment of the web application that may provide an attacker with an opportunity to compromise the System.

The testing server was accessible only during the timeframe of the given security audit. However, since there are other vulnerabilities in the System, its overall security level was rated “average”.

License

This document is the subject to Copyright 2018 Cyber3 under the Creative Commons Attribution NonCommercial NoDerivs (CC-NC-ND) license. You may share and repost the PDF without modification on other sites as long as you put a clear reference link to github.com

Appendix 1. Security assessment

Security level analysis

To analyze the security level of the System, it is necessary to measure the severity and likelihood of exploitation of the detected vulnerabilities. The likelihood of exploitation is measured, according to the ease of vulnerability exploitation and the accessibility of a vulnerability.

Vulnerability severity

The “Severity” property of a vulnerability describes possible results of this vulnerability exploitation, regarding confidentiality, integrity, and availability of information processed on a vulnerable resource. Severity levels are described in the Table A-1.

Level	Confidentiality violation	Integrity violation	Availability violation
None	Does not happen	Does not happen	Does not happen
Low	Obtaining access to noncritical information by an attacker through privilege escalation	Integrity violation of noncritical information by an attacker with basic user rights in the System	Short-time denial-of-service of a mission-critical application
Average	Confidentiality violation of sensitive data by an attacker with basic user rights in the System	Integrity violation of sensitive data by an attacker with basic user rights in the System	Denial of service of a mission-critical application or a short-time denial of service of the System
High	Confidentiality violation of critical information by an attacker with administrator rights in the System	Integrity violation of critical information by an attacker with administrator rights in the System	Denial of Service of the System

Table A-1. Vulnerability severity levels

Ease of vulnerability exploitation

The “Ease of exploitation” property of a vulnerability defines which hardware and software, time and computing resources, and professional skills are required to exploit a vulnerability (Table A-2).

Level	Description
Low	Vulnerability exploitation requires high computing powers, significant time resources, developing new software, configuration analysis of the System, determination and testing possible ways and conditions of successful exploitation of this vulnerability.
Average	Vulnerability exploitation requires high-performance computing, extensive time resources, special hardware and software, and analysis of a violated system configuration. An attacker does not have to have deep knowledge of the system or professional skills to perform an attack.
High	Vulnerability exploitation does not require the use of any special hardware or software, high-performance computing, time resources or any professional skills to perform an attack.

Table A-2. Ease of vulnerability exploitation levels

Vulnerability accessibility

The “Accessibility” property of a vulnerability defines what user classes have access to a vulnerable resource (Table A-3).

Level	Description
Low	Privileged users
Average	Registered users
High	All users

Table A-3. Accessibility levels

Likelihood of exploitation

The likelihood of exploitation is calculated according to “ease of exploitation” and “accessibility” levels (Table A-4).

Likelihood of exploitation		Ease of exploitation ↓		
		Low	Average	High
Accessibility →	Low	Low	Low	Average
	Average	Low	Average	High
	High	Average	High	High

Table A-4. Likelihood of exploitation levels

Vulnerability impact

Vulnerability impact (for one of the existing threats) is measured, according to vulnerability severity (for one of the existing threats) and the likelihood of exploitation of a vulnerability (Table A-5).

Vulnerability impact		Likelihood of exploitation ↓		
		Low	Average	High
Vulnerability severity →	Low	Low	Low	Average
	Average	Low	Average	High
	High	Average	High	High

Table A-5. Impact levels

Appendix 2. Security level analysis.

List of the discovered vulnerabilities and security weaknesses

The vulnerabilities and security weaknesses discovered in the System during the security audit are listed in Table B-1.

Discovered vulnerabilities		
Vulnerability	Overall impact	See
E-mail functionality abuse	High	p.7
Unsafe Flask configuration	High	p.8
Weak service password	High	p.9
Debugging information output	Average	p.9
No protection from Clickjacking	Average	p.10
Reflected XSS	Low	p.11
Unsafe CORS configuration	Low	p.12
Information disclosure	Low	p.13

Table B-1. List of discovered vulnerabilities and security weaknesses

