



# Penetration testing report

**CLIENT NAME**

Client name: #####

Date: #####

# Contents

<b>Executive summary</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
The scope	5
Classification of vulnerabilities	5
<b>Summary of vulnerabilities</b>	<b>6</b>
<b>Vulnerabilities</b>	<b>7</b>
CD-###-01: Access with Invalid Credit Card Credentials (Critical)	7
CD-###-02: Privilege escalation (Critical)	8
CD-###-03: Lack of Input Validation (Critical)	10
CD-###-04: DDoS on the Admin Login Page (Critical)	10
CD-###-05: Security Misconfiguration – Replay Attack (Critical)	10
CD-###-06: Man in the Middle (High)	12
CD-###-07: Unrestricted File Upload (High)	13
CD-###-08: Side Channel on Registered Email Addresses (Medium)	13
CD-###-09: Metadata Exposed by the robots.txt File (Info)	13
<b>Conclusions</b>	<b>14</b>
<b>Disclosure</b>	<b>15</b>

## Document details:

Type	Penetration Testing Report
Author	
Review	
Entity	
Version	1.1
Status	FINAL
Date	30 August 2019

## Executive summary

The service provider was contracted to run a penetration test for ##### WebApp/API, Infrastructure, mobile apps platforms to help protect ##### assets by exposing the existing weaknesses before releasing the product. Our process is designed to simulate multiple attack scenarios to reveal potential vulnerabilities in the system or application. If not addressed in time (strongly recommended before going in a live production environment), these vulnerabilities could be a trigger for a cybersecurity breach.

In accordance with the agreement between Service provider and ##### to validate the current security level of the project, the following phases have been successfully completed:

- ✓ OSINT Phase
- ✓ Black-Box Web Application Penetration Testing
- ✓ White-Box Web Application Penetration Testing via Source Code Review
- ✓ Network Infrastructure Penetration Testing
- ✓ Local Infrastructure Security
- ✓ Staff Security Awareness and Social Engineering
- ✓ Code review

During the assessment, the Service provider team identified x critical vulnerabilities, x high risk problems, x medium risk vulnerabilities and x informational aspects.

The conclusions of this report present the situation during testing and does not reflect the current state automatically. Testing was performed by X, between the 1st August 2019 – 29th August 2019.

In this report, all discovered vulnerabilities and security risks are detailed together with recommendations for solving them.

The analysis includes both identifying known vulnerabilities using automated scanning tools and personalized manual attacks for the target system specifics related with threats lists Top Ten OWASP, Top Twenty SANS.

## Introduction

The Service provider team works following the next industry standards and methodologies:

- National Institute of Standards and Technology - NIST;
- Open Source Security Testing Methodology - OSSTM;
- Open Information Systems Security Group - OISSG;
- Open Web Application Security Project - OWASP.
- Offensive Security Certified Professionals - OSCP;
- Certified Ethical Hackers - CEH;

The pentesting operation was carried in two stages. The first stage highlighted the external penetration testing, operated under the black-box assumption, during which the technical team didn't have any knowledge of the implementation details of the platform. On the duration of this stage the technical team assessed the platform by mirroring a real black-hat malicious hacker that targets the platform and tries to infiltrate from the outside.

The second stage comprised the white-box penetration testing, during which the technical team had access to the source code of the platform and the design diagrams that represent it. The technical team assessed the security proofness of the platform via a process of code review, where our experts looked for flaws, mistakes and bad practices.

Throughout the external penetration testing stage the following network nodes from the infrastructure's distributed system were evaluated:

- #####
- #####
- #####

Throughout the code review stage the following repositories have been cloned and evaluated:

- #####
- #####

Service provider's architecture of the security evaluation proposes a hybrid methodology that combines both automated techniques and manual work to bypass the security controls of the platform.

## 1. The scope

The objective of this IT Security audit is to identify vulnerabilities and data breaches in the modules that compose the overall platform and to recommend patches that would remove these vulnerabilities and improve the general security robustness of the platform. Solving these highlighted issues lowers the chances of a real-world attacker to obtain any malicious benefit out of the platform or to compromise its reputation.

##### wishes to ensure that they are adhering to industry's best practices for their information assets and to meet the highest security requirements and standards to date.

The main objectives of this security assessment project for ##### is to:

- Identify if a remote attacker could penetrate the ABC...
- Determine the impact of a security breach:
  - Confidentiality measures implemented in to the application
  - Internal infrastructure assessment and availability of internal Internet Banking BackOffice system
- etc.

Project stakeholder's primary security concern is that its proprietary and sensitive information is not sufficiently protected from unauthorised access or disclosure.

## 2. Classification of vulnerabilities

Each vulnerability or uncovered risk was labeled under the << **FINDING** >> code and ranked on the following steps: **Critical Risk, High Risk, Medium Risk, Low Risk or Informational**, they are defined based on the following reasons.

### **CRITICAL RISK ISSUES**

These vulnerabilities need to be addressed immediately due to the high degree of threat they display to the network, users or critical infrastructure.

For this kind of vulnerabilities, exploitation does not require advanced tools or special techniques or advanced targeting knowledge.

## **HIGH RISK ISSUES**

These vulnerabilities need to be addressed immediately due to the high degree of threat they display for the network, users or data.

These vulnerabilities don't require a skilled attacker that possesses advanced tools in order to be exploited, therefore they need to be addressed as soon as possible.

## **MEDIUM RISK ISSUES**

This vulnerability category needs to be addressed in time.

Exploitation is generally difficult and requires social engineering, existing access or special circumstances.

## **LOW RISK ISSUES**

These vulnerabilities should be recorded and addressed in the future.

These issues offer limited information possibilities to an attacker and may not be a real threat.

## **INFORMATIONAL ISSUES**

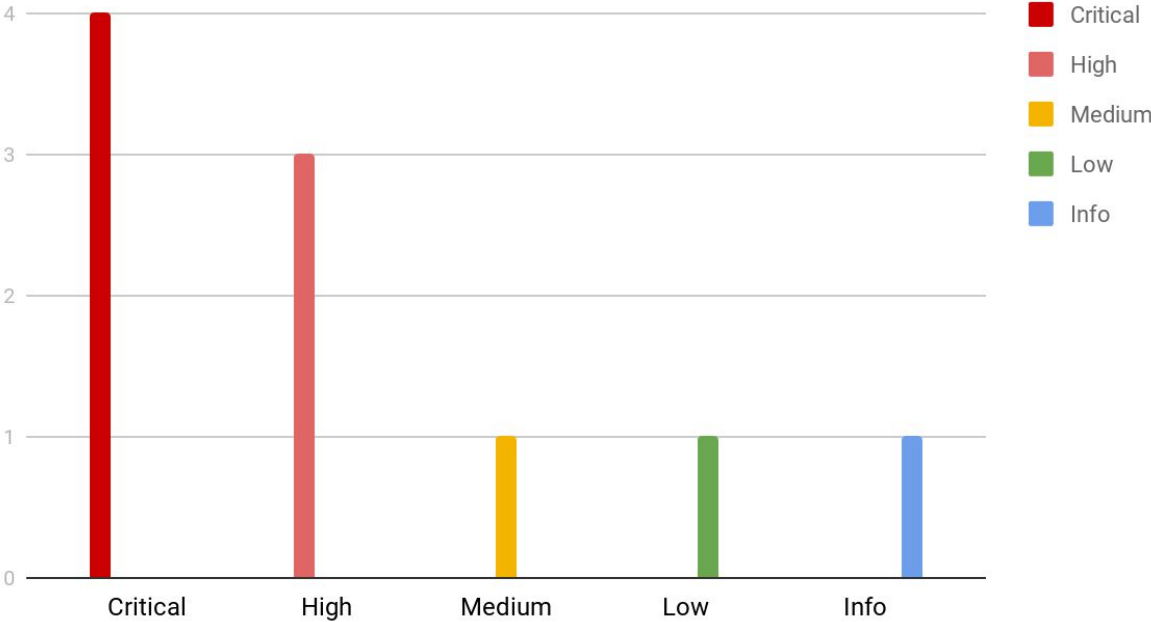
These are informational issues and have extremely low chances to be considered a real threat.

## Summary of vulnerabilities

1.CD-###-01: <b>Access with Invalid Credit Card Credentials</b>	<b>Critical</b>
2.CD-###-02: <b>Lack of Input Validation</b>	<b>Critical</b>
3. CD-###-03: <b>DDoS on the Admin Login Page</b>	<b>Critical</b>
4.CD-###-04: <b>Man in the Middle on Various Pages</b>	<b>High</b>
5.CD-###-05: <b>Unrestricted File Upload</b>	<b>High</b>

6.CD-###-06: <b>Side Channel on Registered Email Addresses</b>	Medium
7. CD-###-07: <b>Frameable response (Clickjacking)</b>	Low
8.CD-###-08: <b>Metadata Exposed by the <i>Robots.txt</i> File</b>	Info

Vulnerability summary chart



# Vulnerabilities

## 1. CD-###-01: Access with Invalid Credit Card Credentials (**Critical**)

**Vulnerability:** #####

**Description:** Upon receiving the online reservation from the hotel pages that embed the ##### API, the backend library that manages online payments doesn't authenticate the cardholder credentials, therefore, allowing a malicious user with fake card data to book rooms at its free will.

This burdens the development of the client hotel since they are exposed to losing legit guests that would have booked the room in a legit way, simply because all the rooms are booked with fake guests.

**Solution:** Add verification and authentication of cardholder data before processing a transaction.

## 2. CD-###-02: Privilege escalation **Critical**)

**Vulnerability:** Our engineers have discovered that altering the “id, email or phone” values, leads to privilege escalation. A malicious user can easily create an account, make use of a valid PUT payload data but with an altered payload in the body section. Once these values are updated, the attacker just has to use the updated phone number within the login screen in order to successfully access the victim’s account.

### Exploit details:

<b>Asset(s)</b>	https:// #####
<b>Parameter(s)</b>	Id, email, phone
<b>Attack Vectors</b>	Values of another user
<b>Reference</b>	https://#####

### Evidence / How to Reproduce

Raw HTTP request used to retrieve the page.

```
PUT /api/users/3E38C1DB-1C19-4631-8A76-48A1719A0EE0 HTTP/1.1
Host: #####
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:57.0) Gecko/20100101
Firefox/57.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
content-type: application/json
Cache-Control: no-cache
Pragma: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
```



```
Authorization:Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IkRCN3UtSkRLbjRqcW5falZ2YzBGTlota
HI0cyIsImtpZCI6IkRCN3UtSkRLbjRqcW5falZ2YzBGTlotaHI0cyJ9.uRanoPaz2wU0Hac-nJkeA
UPK0KEcAvluYtj7XFurD2EFLrqv7Aoa4TJjpcqVcvZUOjoaS3VVbGw6HvLJvLKWwhzMUB6
b-_cvN4qTmcqgdrPSmFHq25wgUMVVByW99MtacUfRDX2T0jkVvXylkL_qzaxpABlgy5G6vjh
H-gdMI_3uzK9vNG69KjCvN439jcqohy3deyFvDTAW4vb3cEqz2WMdYrrFSSN202gCGVdGG
BAcRnfGUFkc9aEUBRaDFiEFupU8jvjBrZTrNn3ZaXE8qsRXfTEfM-kW44C_gaa-0Nc_Mbojvp
88zz5NpSw9__hX46UmpNvQhnSqPEAupAHw
Content-Length: 243
Connection: close
{
  "id": "ADED9C73-70E1-451C-A605-B4DDBC3C3D2B",
  "firstName": "Mike",
  "lastName": "Mike",
  "email": "#####",
  "phone": "#####",
  "gender": "Male",
  "birthDate": "1983-12-15",
  "language": "en-us"
}
```

Raw HTTP response used as the page basis.

```
HTTP/1.1 200 OK
Content-Length: 461
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=abe18183e77faf1e87f82aa4578c0ed58f288a8204c71035817c6810452761b5;Pa
th=/;HttpOnly;Domain=#####
Connection: close
{"email":"#####","avatar":null,"birthDate":"1988-12-015","gen
der":"Male","language":"en-us","isRegistrationCompleted":true,"isPrivacyStatementApproved":
true,"key":"04A132D8-058E-4E0C-9A8A-A74BB6CDC048","boxActionCount":0,"tagKey":null,"
passUri":"api/passbook/v1/passes/#####/61a24603-561c-4c
```

```
a8-bea9-eff0808ad10","id":"ADED9C73-70E1-451C-A605-B4DDBC3C3D2B","firstName":"Mihai","lastName":"Mike","phone":"+40747003223"}
```

**Solution:** Validate all accessible inputs.

### 3. CD-###-03: Lack of Input Validation (**Critical**)

**Vulnerability:** The HTTP packets that carry the reservation forms can be manually crafted to modify fields that are critical for the room reservation process.

**Description:** When sending a reservation request to the backend server, the “total fee” field can be modified in the HTTP payload to contain undesired values, such as 0, which gets processed by the backend server and saved in the database, without validating it beforehand. Thus, malicious users can obtain reservations for free. In a similar fashion, the field “guests” can be modified after the client web-browser has fired the HTTP request carrying the reservation form, and the backend server doesn’t recompute the “total fee” field based on this new value but it processes the request, allowing a room to be overbooked with a lower price than the fair value given the number of guests.

**Solution:** The critical fields must be validated before being processed on the server, and the fields with constant values, such as fees or limits, must be retrieved from a database and not from the request coming from the client browser. The “total fee” must be computed on the server, with values queried from the database, corresponding to the room the user chose to book. The “guests” field must not overcome the maximum number of guests registered in the database for the respective room.

### 4. CD-###-04: DDoS on the Admin Login Page (**Critical**)

**Vulnerability:** A series of failed logins on the admin webpage <http://###> triggers a protection timer which blocks any other login attempt until the countdown has expired.

**Description:** A malicious user can manually craft HTTP requests in order to fire an avalanche of login requests with invalid credentials, that would trigger the protection timer making any subsequent login attempt fail. Therefore, the admin login webpage can be made completely unavailable with an automated script that makes invalid login requests.

**Solution:** Replace the protection timer with a ReCAPTCHA widget.

### 5. CD-###-05: Security Misconfiguration – Replay Attack (**Critical**)

**Vulnerability:** Due to the “replay attack” vulnerability, a malicious user can iterate the access code using the auto-generated string brute force method to gain access to a user’s account.

**Exploit details:**

<b>URL</b>	<b>https://#####/connect/token</b>
<b>Parameter(s)</b>	<b>code</b>
<b>Attack Vectors</b>	<b>Randomly generated</b>
<b>Reference</b>	<b>https://#####</b>

**Evidence / How to Reproduce**

Raw HTTP request used to retrieve the page.

```
POST /ids/connect/token HTTP/1.1
Host: #####
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://#####.com/authentication/register/account-verify;emailOrP
hone=mihai@test.com
Origin: https://#####.com
Connection: close
client_id=wap&client_secret=no_secret&grant_type=custom&scope=email%20offline_access
%20openid%20profile%20roles%20api.general&code=287316
```

Raw HTTP response

```
HTTP/1.1 200 OK
```

**Solution:**

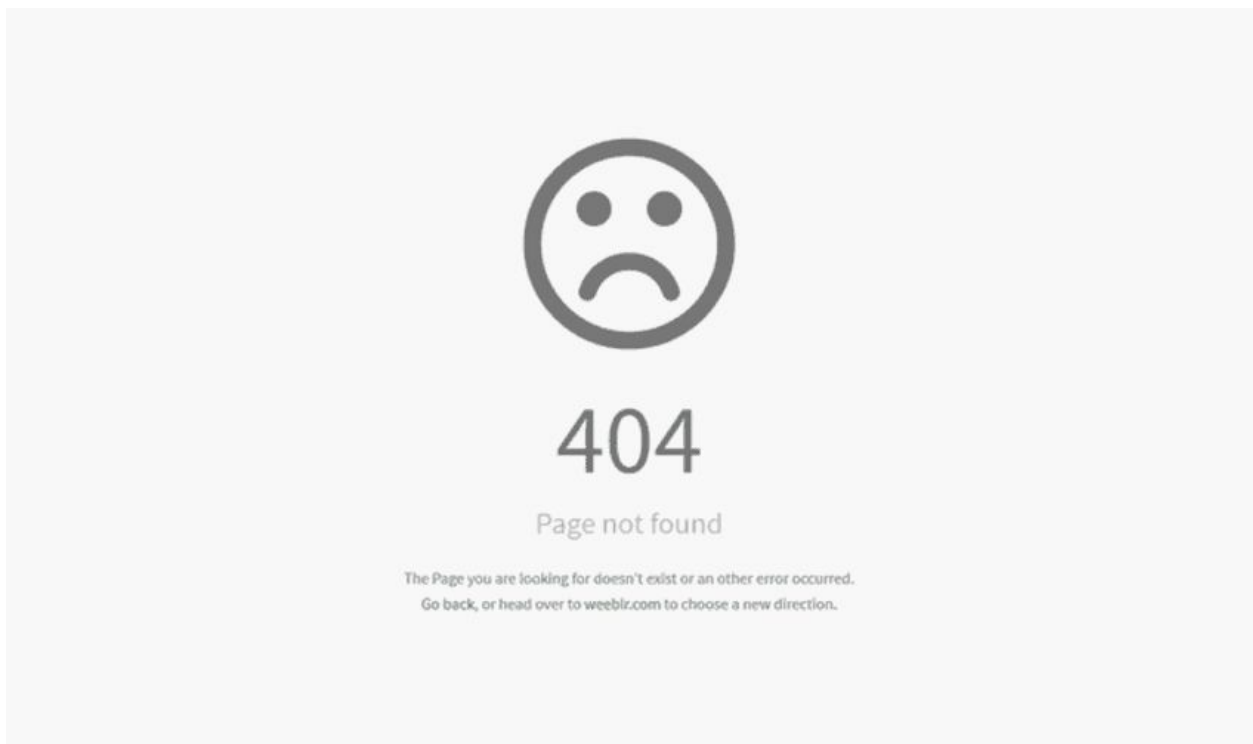
Within any of the app’s functionality, the requests should be protected in such way that replay attacks are not possible. The solution for this would be the usage of a unique token for every request.

## 6. CD-###-06: Man in the Middle (High)

**Vulnerability:** Various web pages that interact with user's sensitive data lack implementation of a secure encrypted protocol, therefore sending data through the network in plaintext.

**Description:** The lack of an encrypted protocol on pages that interact with user's sensitive personal information lead to transmitting packets that travel un-encrypted through the malicious network. This allows any malicious attacker that sits between the source client and the destination backend to intercept the packets and read the sensitive data in the plain. The vulnerable web nodes are the following:

- <http://###> - an attacker can extract the login credentials of any hotel admin that logs in to the page
- <http://###> - an attacker can extract the login credentials of any user that administers the webpage of a hotel




Print screen examples

**Solution:** Integrate an encrypted HTTP protocol such as HTTPS.

## 7. CD-###-07: Unrestricted File Upload (High)

**Vulnerability:** A user logged in the admin panel can upload any file with any format via the fields *My Properties* (<http://###>) and *Extra* (<http://###>). Among the most dangerous formats are executable scripts that affect the end-user or can compromise the entire backend.

**Description:** After a malicious user has obtained access to the admin panel, it has the freedom to upload any type of file with any format through the fields *My Properties* and *Extra*. These open a vulnerable surface that can be exploited via Cross-Site Scripting methods which allows an attacker to execute arbitrary code on the user's web client. At the same time, the attacker can drop various executable scripts in the backend file system, which by mistake could be executed by a sloppy backend administrator.

 At the current time, it is not possible to obtain *Remote Code Execution* in modern web browsers, but a new platform feature further developed in the future, might allow executing scripts on the backend server. This is why, it is of uttermost importance to check the type of the file when uploading it and reject it if it's of an undesired format.

**Solution:** Incorporate validations on the uploaded files' formats via a third-party library.

## 8. CD-###-08: Side Channel on Registered Email Addresses (Medium)

**Vulnerability:** The authentication server on <http://###> replies differently for login requests that have an invalid username and login requests that have a wrong password.

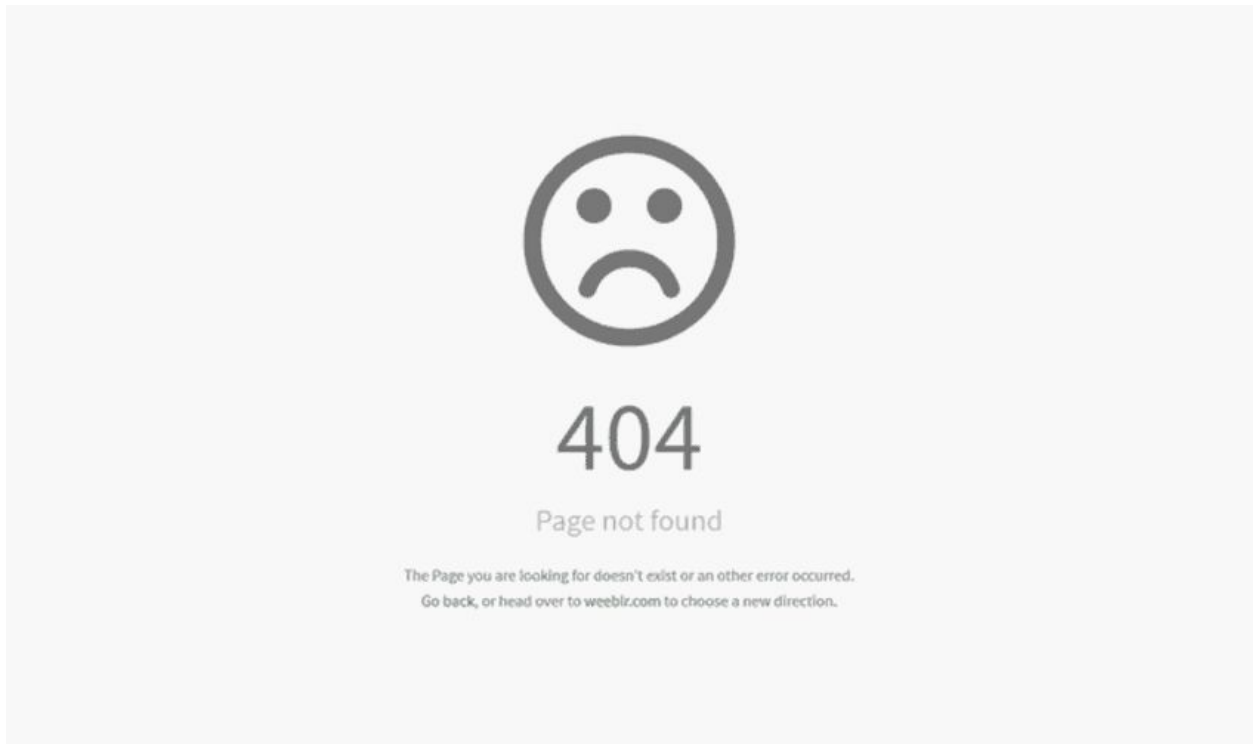
**Description:** This type of vulnerability allows an attacker to obtain additional information that may lead together with other weaknesses to exploit a more dangerous vulnerability, for example those mentioned above. The authentication page <http://###> replies with an error message *EMAIL\_NOT\_FOUND* for login requests with invalid email and with *INVALID\_PASSWORD* for requests that have a wrong password. Thus, an attacker can learn which email addresses are registered in the platform allowing them to proceed on step closer to their malicious goal.

**Solution:** Configuring the authentication server to reply with a single type of login error message that would make impossible to differentiate between invalid username and password - *INVALID\_LOGIN*

## 9. CD-###-09: Metadata Exposed by the *robots.txt* File (Info)

**Vulnerability:** The *robots.txt* file provides a map of the website to the automated bot crawlers indicating them which paths should be indexed by search engines and which not.

**Description:** The file <https://###/robots.txt> is publicly accessible and offers details about the infrastructure of the web platform and the *xml* map of the website. A malicious user can misuse these information as resources in a further attempt to exploit a more critical vulnerability.



Print screen examples

**Solution:** Protecting the resources that are meant to have restricted access via *Access Control Lists* and not via specifying rules for them in the *robots.txt* file.

## Conclusions

1. The evolution of the platform with advanced further features will increase the risk that new vulnerabilities will emerge on the backend side.
2. Integrating the *###* API in further client websites increases the risks that new vulnerabilities will rise on the client's side.
3. The reputation boost of the platform will attract more and more skilled malicious users.
4. The presence of more than 3 critical vulnerabilities yield for taking immediate action regarding the security of the system.

5. Continuous maintenance and evaluation of the system with respect to contemporary and modern threats will allow the platform to evolve at a high pace.

## Disclosure

Legal note:

- This report may contain confidential and private data on the Service provider or the client.
- This report cannot be submitted to third parties without formal consent of the provider.
- Unauthorized use or reproduction of this document is prohibited.
- Tests and evaluations were carried out by IT Security Department.
- This report presents all relevant vulnerabilities known at the time of its writing.

Penetration tests are a specific representation in time, of the existing vulnerabilities, they are coordinated over a specific application, identified as "Target". All products are prone to a multitude of errors, omissions, and defects that are divided into different severity and quantity categories.

The technical team performs a series of pre-established tests in relation to the purpose of the project. The service provider uses an approach and tactics based on technical expertise, past experiences and industry best practices at the time of the test.

The testing team can guarantee the identification of security issues that are known at an international level, with the indication that some problems may not be fully identified due to the lack of knowledge and tools to discover them, at a global scale. The main argument for this is because new vulnerabilities and ways to perform tests can appear and will appear in the future.

In conclusion, through this audit, the Service provider performs a test that involves all the knowledge and best practices, that the team has in its ability and technical expertise to identify vulnerabilities limited by the purpose of the test and the understanding reached between both legal entities.