For:

Information Security Audit for ptwh.co.il:5002/ web application

1. The audit was made at ptwh.co.il:5002/ request, with the aim of assessing the Phantom program level of security.
2. This report contains all the findings, risks, exposures and recommendations that we have found during the audit.
3. This report is relevant for the period of the audit - March 2021.
4. This survey was made by Roy kopit.
5. In the course of the audit it was found that the system does not meets the information security standard for systems of this type,there were number of deficiencies that require treatment.

Table of content:

Chapter 1 Executive Summery……...3
Chapter 2 Methodology…….5
Chapter 3 Findings Summery……..8
Chapter 4 Finding And Exposures……..10

Chapter 1 – Executive Summery

Summary

At ptwh.co.il:5002/ request, Roy Kopit has performed a security audit to the web application system. The tests were conducted to assess the security level of the system to take precautionary steps to reduce the risks to the system's integrity. The survey was conducted in a format that would enable the company to identify and deal with the main risks in a manner that would reduce or eliminate the chances of realizing the damage to the system

Method

The survey was conducted in a format that would enable the company to identify and deal with the main risk in a manner that would reduce or eliminate the chances to realize the exposure to damage to the company's system. The test structure was developed in accordance with known methodologies, but it should be noted that it is not possible to cover the possibility of exploiting the system.

Test objectives

• Performing an examination that simulates a potential attacker who tries to attack the system. In accordance with the results, to assess the range of risks in a manner that will enable the provision of solutions required to reduce or eliminate the possibility of realizing the exposure to harm the technological system.

• Receive a current and objective assessment of the system security level in a way that will enable ptwh.co.il:5002 to:

✓ Identify failures discovered outside the organization's network.

✓ Perform a risk assessment for the product and define their level of severity.
✓ Implement recommendations to improve the existing situation.
✓ Evaluate the steps required for implementing the recommendations.

General Impression

During the audit, it was found that the general level of security of the system is bad and there are number of deficiencies that require attention

Chapter 2 – Methodology

The stages of the audit
the audit includes 4 stages:

1.Planning the execution of the audit
2.information gathering
3.Analysis of results
4.Formulation of recommendations for implementation

Testing process
The test team analyzed the results in the following configuration:
1. The subject that was tested: The level of the technological security system, and the way the system was developed and deployed.

2. Description of the problem: An explanation of the Invalid findings that were detected, for example, that allows an attacker to steal user identities and commit brute force attacks.

3. Findings of the examination: Findings that clearly and concisely describe an existing situation. The purpose of the section is to document the existing situation as it was found at the time of the examination. The findings of the examination may be valid or at a level that endangers the entire system under examination at the level of exposure to damage to the continuity of activity, damage to property and people. The findings are often accompanied by screen shots

4. Severity level: How to determine the severity level is as follows: At this stage, the testing team assessed the level of risk resulting from the combination of scenarios of threats and

various defects. In some cases, the combination of a group of deficiencies created a specific risk. In other cases, a specific defect will create a specific risk.

5. Determination of exploitability: the probability in which a vulnerability could be exploited in the system defect is likely to be determined by the following factors:

o Motivation and capabilities of the source of the threat
o Effectiveness of controls to minimize threat

stage of analysis of the findings:

• Determination of the probability level of the threat: reasonableness of probability that some threat will be realized as a result of determining the level of capabilities required to realize this threat

Exploitability + Description:

-High: The source of the threat is highly motivated and highly capable, and the controls that can prevent exploiting the weakness are ineffective.
-Medium: The source of the threat is motivated and capable, but there are controls that can somewhat prevent exploiting the weakness.
-Low: The source of the threat is not motivated nor capable, or there are controls that can significantly prevent exploiting the weakness.

• Determining the level of damage:
Estimating the damage that can be caused by successful exploitation of a defect by a threatening scenario.
Estimating the damage that a threat scenario may cause

is described by the degree of damage to one or more of the information security values:

o Infringement in information Secretly- Damage caused by unauthorized disclosure of information

o Infringement in information integrity- Damage caused by unauthorized change of information

o Infringement in information availability- Damage caused by from shutdown / slowing the system

Chapter 3 - Findings Summery

Details of all the findings, Vulnerabilities, and examined subjects

#	Examine subject	Description	Overall Risk
1	Broken Authentication	It was found that the application no preventing from automated attacks such as “credential stuffing” and allows weak passwords	High
2	Broken access control	It was found that the application and not completely blocking the option to type “only numbers” and by that allow people to stealing money	High
3	The application is vulnerable to sql-injections	It was found that the application is vulnerable to sql injections - manipulating the data base for malicious cause	High
4	Exposing a sensitive information about a user	It was found that the application has a dangerous page that is allowing to exploit user credentials	High
5	Allowing to include remote pages	The application is allowing to include pages that was not created on the original machine	High

6	Allowing xss injections	The web application is allowing to be injectable to a malicious java script code	High
7	Unrestricted File Upload	Uploaded files represent a significant risk to applications. The attack allowing the hacker to upload an external code to the system.	High
8	Sensitive Data Exposure#1	It was found the the application is allowing access to page that contains exclusion paths and sensitive information	Medium
9	Sensitive Data Exposure#2	It was found the the application is reveals unnecessarily an information about the server	Low
10	Session cookie not secure	It was found that the session cookie is being passed unsecured over an HTTP connection	Low
11	There is no directory update	It was found that the application is using an outdated and vulnerable version of jQuery directory.	Low
12	Vulnerable to Clickjacking	The application was found to be vulnerable to Clickjacking attack.	Low

Chapter 4 – finding and Exposures

1. Broken Authentication

Finding Summary:
During the audit, it was found that the application permits default, weak, or well-known passwords, such as “Password1” or “admin/admin“,Permits automated attacks such as “credential stuffing”, where the attacker has a list of valid usernames and passwords

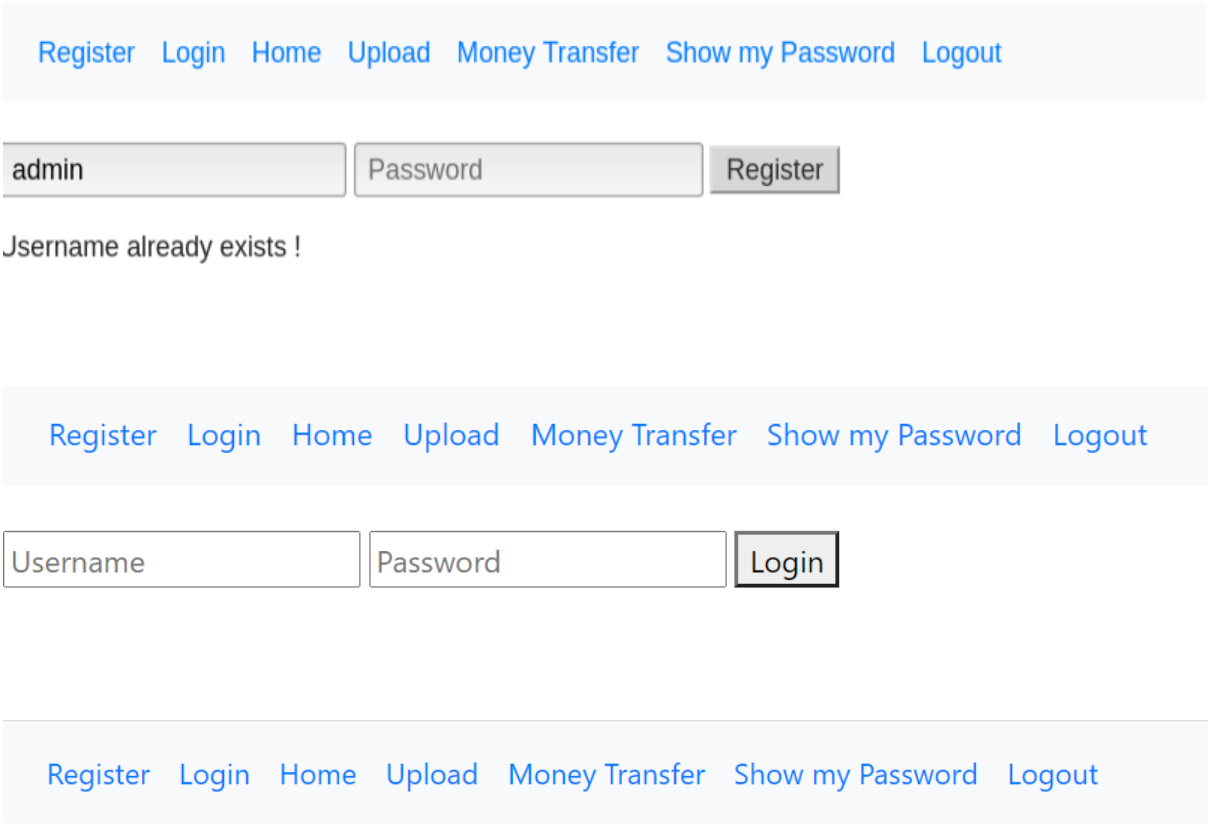
Exploitability: Medium
Severity: High
Overall Risk: High

Risk details
An attacker will try default usernames and on this web app it is very easy to verify if this user actually exist through the register page.without any anti-automation system like captcha the attacker can automate this process and make it even easier to exploit.The password policy is weak and allowing user to create passwords that are easy to exploit with brute-forcing attack or even by guessing.

Recommendations:

- Choose a stronger username especially for a strong user
- Use an anti-automation system like captcha to avoid brute-forced attack against user enumeration in the register page and in the login page
- Don’t allow weak password, make the user use symbols letters(lower and upper case) and number

The following screenshot describes exposing of information:



2. Broken access control - “stealing money”

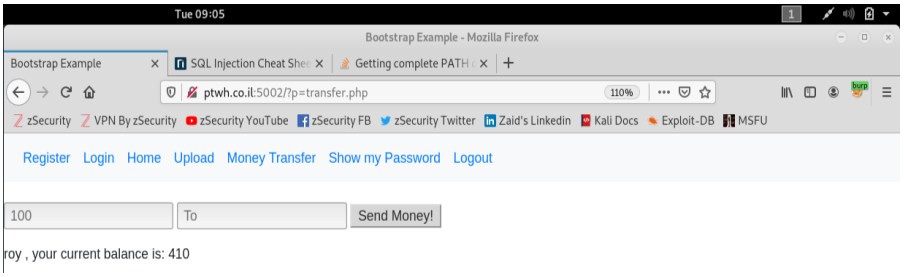
Finding Summary:
During the audit, it was found that the application It was found that the application and not completely blocking the option to type “only numbers” and by that allow people to steal money.

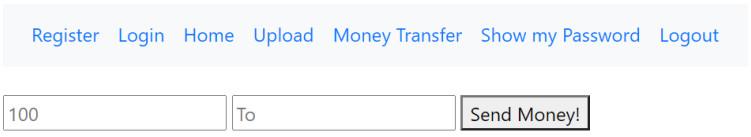
Exploitability: High
Severity: High
Overall Risk: High

Risk details:
When an attacker intercepts the post message that goes to the server he can edit the input he typed on the app and add a ‘-’ (for example) to his original input. by that he is able to change the function from sending money to taking money.

Recommendations:
Don’t allow any other input in the backhand except an ‘only digit’ input.

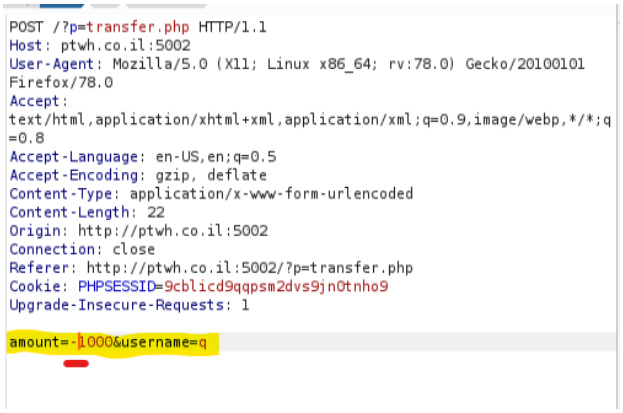
The following screenshot describes exposing of login pages:

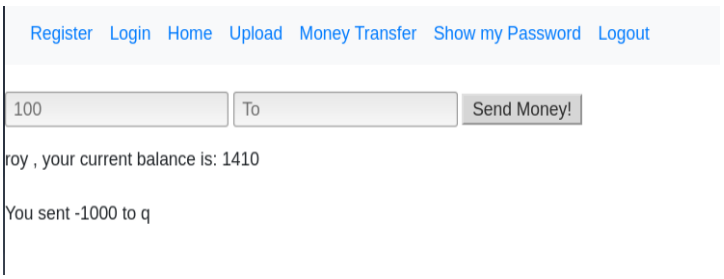


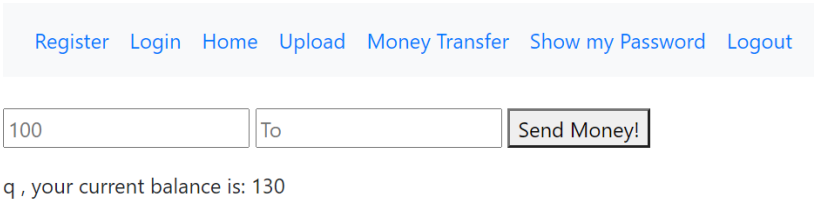


q , your current balance is: 1130

roy , your current balance is: 410







3. The application is vulnerable to sql-injections

Finding Summary:
During the audit, it was found that the application is vulnerable to sql injections - manipulating the data base for malicious cause.

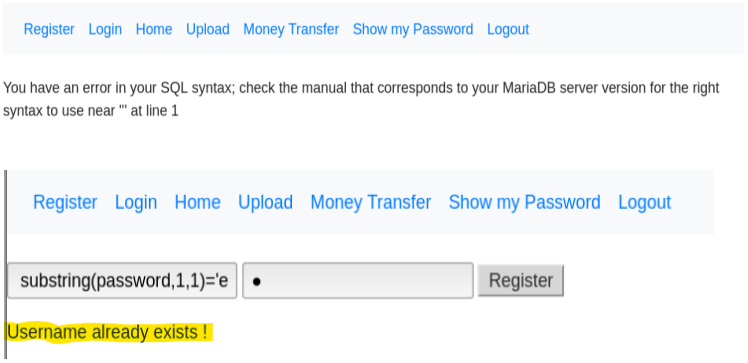
Exploitability: High
Severity: High
Overall Risk: High

Risk details
With sql injection the attacker can pull data from the servers data base and can use it to exploit/do any other harmful damage.
In this case with sql-injection I managed to get the admin credentials

Recommendations:

- **Use of Prepared Statements**
- **Use of Stored Procedures**
- **Whitelist Input Validation**
- **Escaping All User Supplied Input**
- **Enforcing Least Privilege**
- **Performing Whitelist Input Validation as a Secondary Defense**

The following screenshot describes exposing of login pages:



admin username + substring first password letter brute force

4. Exposing a sensitive information about a user

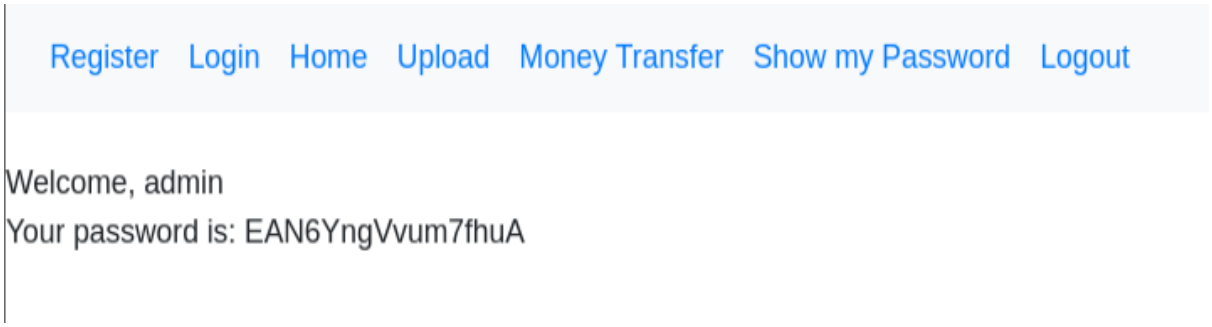
Finding Summary:
During the audit, it was found that the application has a dangerous page that is helping to exploit user credentials ‘show my password’ page

Exploitability: High
Severity: High
Overall Risk: High

Risk details
An attacker can get some sort of a control over a user,as long as he didn't get full control he is limited.’show my password’ page allowing to the attacker to escalate his privileges without any effort
Recommendations:

- Delete this page

The following screenshot describes exposing of login pages:



5. Allowing to include remote pages

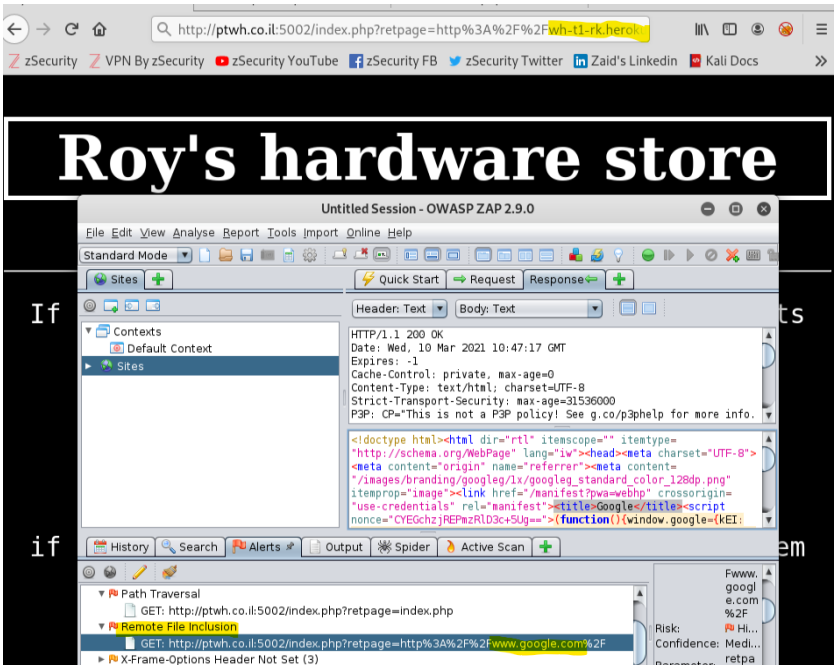
Finding Summary:
During the audit, it was found that the is allowing to remote include pages that were not created on the original machine.

Exploitability: High
Severity: High
Overall Risk: High

Risk details
With this setting open an attacker can execute almost anything he wants on the server because the app allowing to the server to read any included page the attacker would give her.
Recommendations:

- on the php.ini set the allow_url_fopen and the allow_url_include to off

The following screenshot describes exposing of login pages:



6. Allowing xss injections

Finding Summary:
During the audit, it was found that the application is injectable to a malicious java script code.

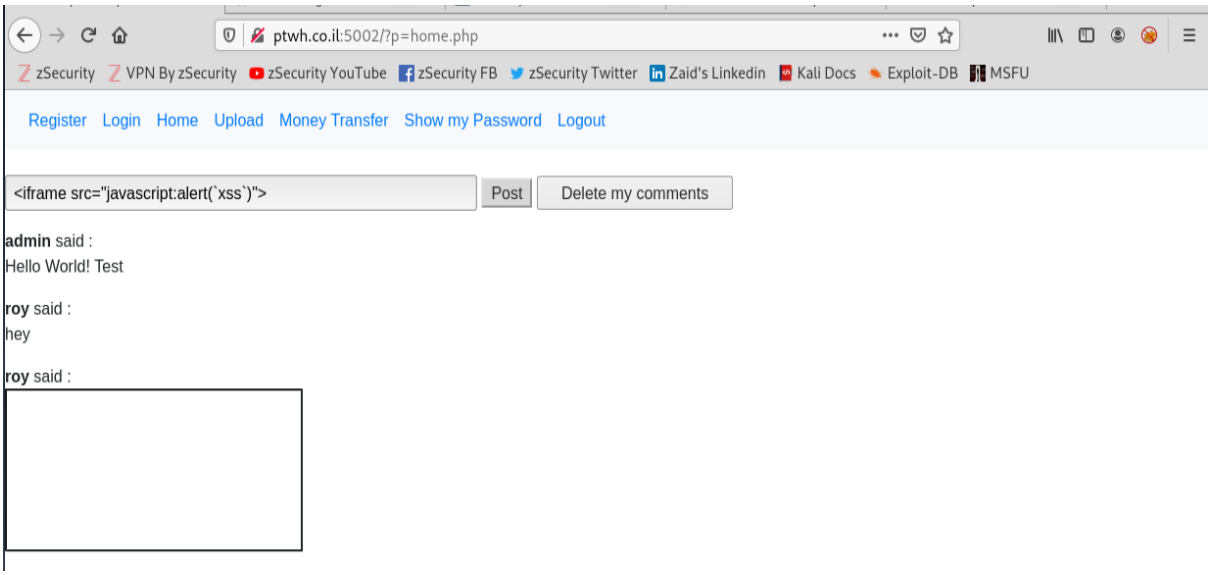
Exploitability: High
Severity: High
Overall Risk: High

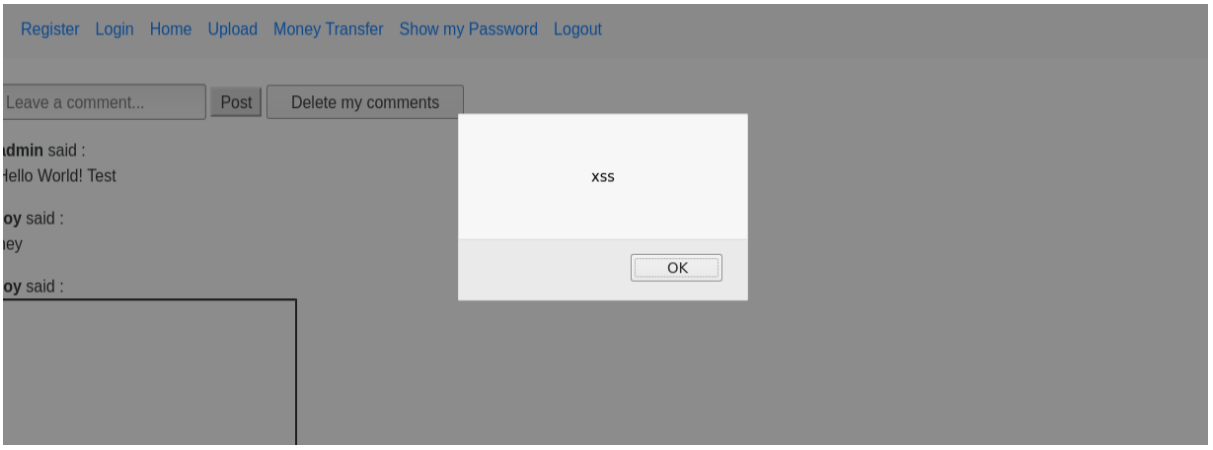
Risk details
An attacker can inject different types of code to the app that will be stored on the app until someone deletes it manually.
That code can hurt anyone who uses the app and sometimes even take over their machine through the web app

Recommendations:

- Use HTTPOnly cookie flag
- Never Insert Untrusted Data Except in Allowed Locations
- HTML Encode Before Inserting Untrusted Data into HTML Element Content
- Attribute Encode Before Inserting Untrusted Data into HTML Common Attributes
- JavaScript Encode Before Inserting Untrusted Data into JavaScript Data Values
- CSS Encode And Strictly Validate Before Inserting Untrusted Data into HTML Style Property Values
- URL Encode Before Inserting Untrusted Data into HTML URL Parameter Values
- Sanitize HTML Markup with a Library Designed for the Job
- Avoid JavaScript URLs
- Prevent DOM-based XSS

The following screenshot describes exposing of login pages:





7.Unrestricted File Upload

Finding Summary:

During the audit, it was found that the application allowing to upload a dangerous type of files(like .py, .bat .apk) to the server that can be exploited later with phishing attacks

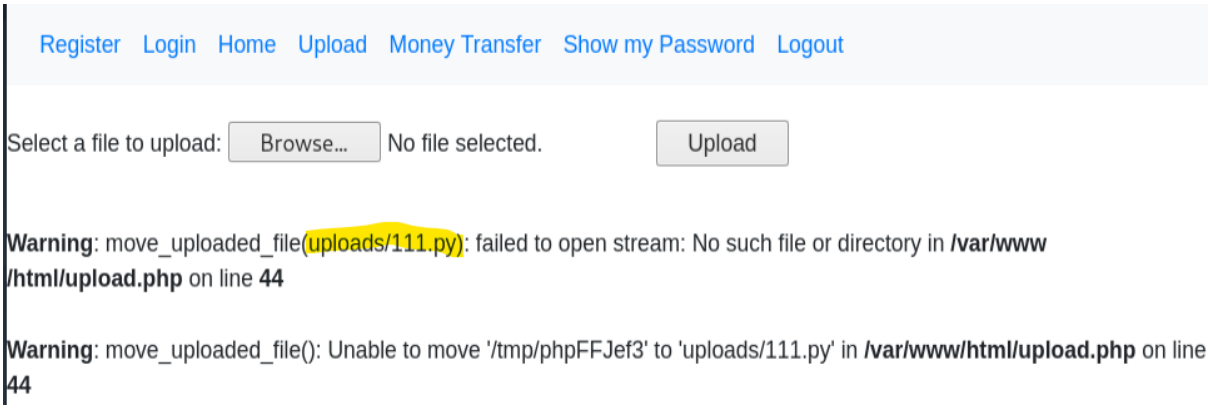
Exploitability: Medium
Severity: High
Overall Risk: High

Risk details:
Uploaded files represent a significant risk to applications. The first step in many attacks is to get some code to the system to be attacked. Then the attack only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step.

Recommendations:

- make an assessment about what types of file you should allow to upload and block the unwanted files (exactly how you did with the .php , .exe files).

The following screenshot describes exposing of login pages:



8. Sensitive Data Exposure#1

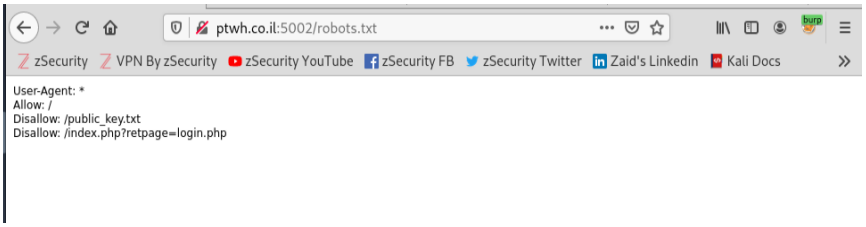
Finding Summary:
During the audit, it was found that the application is allowing access to page that contains exclusion paths and sensitive information- ‘/robots.txt’ , ‘/public_key.txt’ , ‘/index.php?retpage=login.php’

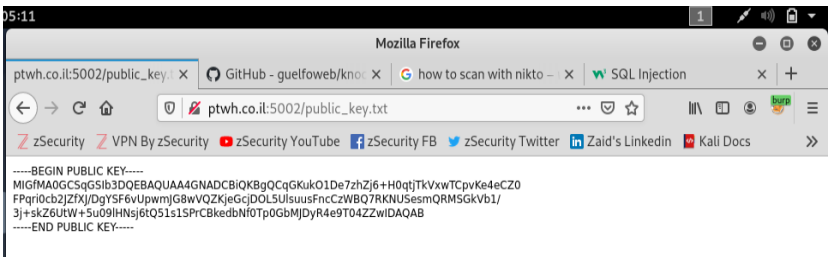
Exploitability: Low
Severity: High
Overall Risk: Medium

Risk details
‘robots.txt’ is a well known file that his purpose is to tell the browser what to exclude from the app - and for the hacker where he might find something interesting.In this case that file contained a public ssh-key but the ssh service on this machine was off so that currently not exploitable but its a type of data that you should avoid to share.
Recommendations:

- Don't put any sensitive data on robots.txt an attacker will always look there

The following screenshot describes exposing of login pages:





9. Sensitive Data Exposure#2

Finding Summary:
During the audit, it was found that the application is revealing unecessarly information about where the application pages is stored on the server & upload file locations.

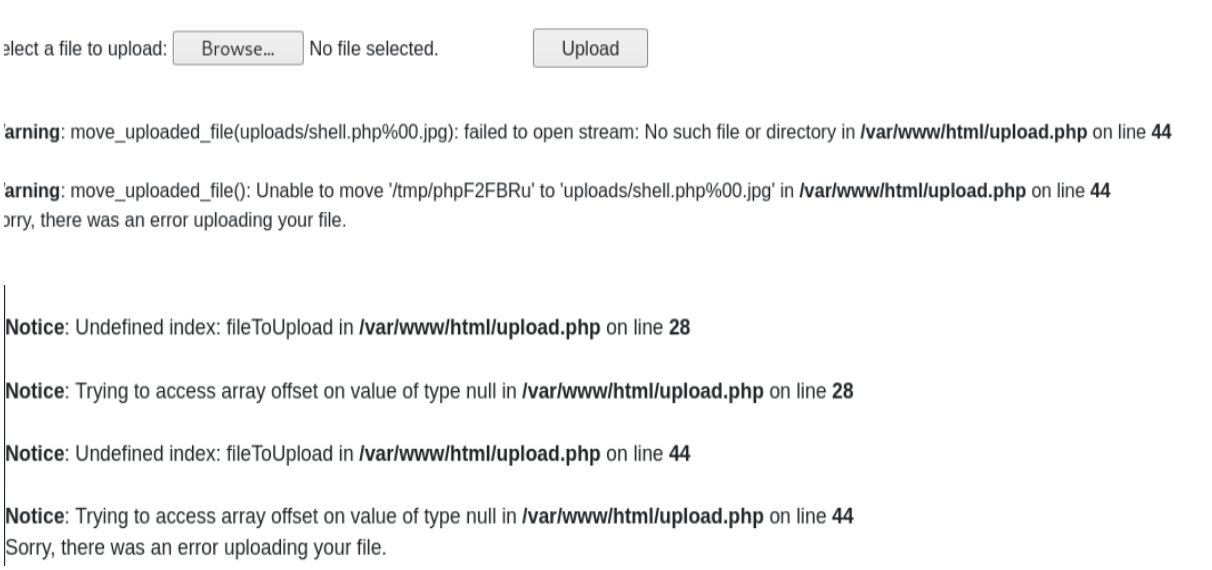
Exploitability: Low
Severity: Low
Overall Risk: Low

Risk details
on the upload page you can see the full paths to the current page on the server.this information is not useful to the customer and in some cases can be useful to the attacker that his intention is to harm the app/company .

Recommendations:

- use a generic error message like ‘file has failed to upload’
- don’t expose any unnecessary information

The following screenshot describes exposing of login pages:



10. Session cookie not secure

Finding Summary:
During the audit, it was found that the session cookie of host name ptwh.co.il:5002 not marked as secure.

Exploitability: Medium
Severity: Low
Overall Risk: Low

Risk details
This cookie will be transmitted over a HTTP connection, therefore if this cookie is important (such as a session cookie), an attacker might intercept it and hijack a victim's session. If the attacker can carry out a man-in-the-middle attack, he can force the victim to make an HTTP request to steal the cookie

Recommendations:
• Mark all cookies used within the application as secure. (If the cookie is not related to authentication or does not carry any personal information, you do not have to mark it as secure.)

The following screenshot describes exposing of login pages:

11. There is no directory update

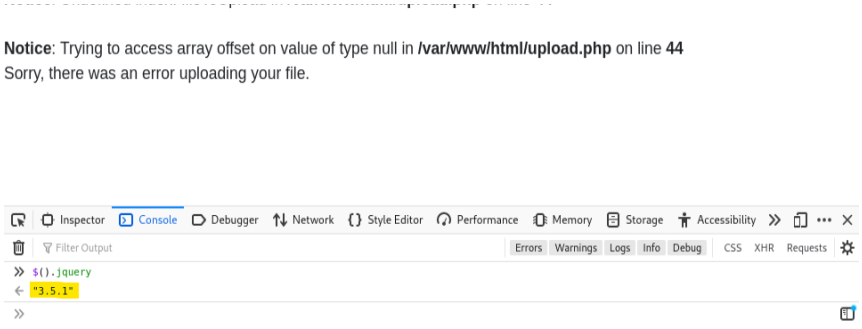
Finding Summary:
During the audit, it was found that the application is using an outdated and vulnerable version of jquery - version 3.5.1 (Latest version 3.6.0).

Exploitability: Low
Severity: Low
Overall Risk: Low

Risk details
In the case of working with non-current versions we are exposed to the exploitation of various vulnerabilities and known by potential attackers. The main reason for downloading and installing the latest update is to stay protected from security threats. Older software will continue to have the same bugs and exploitable holes in the code that allow hackers and cyber criminals to get up to good. This is made even more serious by the fact that all these exploitable entry points have generally been made public after the release of updates.

Recommendations:
• Upgrade to the latest official jquery version.

The following screenshot describes exposing of login pages:



12. Vulnerable to Clickjacking

Finding Summary:
During the audit, it was found that the application is vulnerable to Clickjacking attack.

Exploitability: Low
Severity: Low
Overall Risk: Low

Risk details
Clickjacking allows an attacker to attack users with "interactive" clicks through transparent or hidden layers, which can be placed on vectors of possible attacks, such as buttons and links, that can mislead users into interacting with an attacker's malicious code.

Recommendations:
• Use X-Frame-Options: SAMEORIGIN on pages (or whole website) which are not intended to be viewed inside frames. • Use a covering
if we want to allow our pages to be shown in iframes, but still stay safe.

The following screenshot describes exposing of login pages: