MNZ®

INFORMATION SECURITY SERVICES

MNZ
INFORMATION SECURITY SERVICES

finance

E-FINANCE

APPLICATION GRAY BOX PENETRATION TESTING

Submission Date: 12th, July 2020

Version: 1.0

INFORMATION SECURITY SERVICES

TABLE OF CONTENT

1. DOCUMENT PROPERTY

Content	KHALES APPLICATIONS GRAY BOX PENETRATION TESTING
Classification	CONFIDENTIAL
Department	M&Z Cyber Security Services Department
Version	1.0
Contacts	mhossam@mnztechnology.com
Author	Gamal Negm Eldin
Approved by	Mohammad Hossam

2. EXECUTIVE SUMMARY

2.1. OVERVIEW

The purpose of the assessment is to determine security flaws for Khales application. The tests are carried out assuming the identity of an attacker or a user with malicious intent.

As a result of the penetration testing exercise it was possible to confirm that the application is vulnerable to multiple security flaws such as:

- A Malicious user with no privileges can reset any user or admin password.
- A Malicious user with view privileges account can reset the admin's account password.
- Changing the final service bill price to pay less or nothing.
- Sign up a new account with others' mobile number and bypass verification.
- Access for some features and information with no authentication.
- Injecting a malicious code inside an exported file in the admin panel.
- Sending SMS messages incurs cost and affect the messages quota.
- Guessing the admin and user password by trying a huge list of password without limiting.
- Leakage for the important system information and configuration in error responses.
- Missing an important security cookies flag.
- Bypass the password policies and use very weak passwords.

Detailed vulnerabilities, related to the application are also included in this report corresponding to "Critical", "High", "Medium", and "Low" level vulnerabilities those require immediate actions.

2.2. DISCLAIMER

This security assessment was conducted to Khales on production environment, the description of findings, recommendations, and risks were valid on the date of submission of this report. Any projection to the future of the report's information is subject to risk due to the changes in the Infrastructure architecture, and it may no longer reflect its logic and controls.

MNZ Technology is not absolute and can never be guaranteed. New vulnerabilities are constantly being discovered, which means there is a need to monitor, maintain and review both policy and practice as they relate to specific use cases and operating environments on a regular basis.

3. PENETRATION TESTING METHODOLOGY

Web application penetration testing refers to a set of services used to detect various security issues with the web applications and identify vulnerabilities and risks.

Following are the some of the key areas of which the Web application is tested for such as but not limited to:

- ✓ Authentication Testing
- ✓ Authorization Testing
- ✓ Session Management Testing
- ✓ Input Validation Testing
- ✓ Testing for Error Handling
- ✓ Testing for weak Cryptography
- ✓ Business Logic Testing
- ✓ Client-Side Testing

Following are the some of the key areas of which the remote services is tested for:

- ✓ Getting information about the remote services
- ✓ Testing the service with the open source web application auditing tools
- ✓ Manually testing was carried out to discover more in-depth vulnerabilities with in service
- ✓ Verify the discovered vulnerabilities to minimize the false positives and negatives
- ✓ Perform penetration testing in accordance with approved document

For the online services, the "**OWASP**" Top ten lists served as a guide and the domains tested for are listed below but not limited to them:

A1-Injection
A2-Broken Authentication and Session Management
A3-Sensitive Data Exposure
A4-XML External Entity (XXE)
A5-Broken Access Control
A6-Security Misconfiguration
A7-Cross-Site Scripting (XSS)
A8-Insecure Deserialization
A9-Using Components with Known Vulnerabilities
A10-Insufficient Logging & Monitoring

OWASP Top 10 2017

M&Z's assessment methodology includes structured review processes based on recognized "best in-class" practices as defined by such methodologies as the ISECOM's Open Source Security Testing Methodology Manual, the Open Web Application Security Project.

M&Z used custom technical methodology for conducting vulnerability assessments and penetration testing of network and applications to provide you with the control over the attacks and its impact to meet the goals; below are the methodology details:

- ✓ Perform broad scan to identify potential areas of exposure and services that may act as entry point.
- ✓ Try to exploit the vulnerabilities to gain access, elevate privileges and use it as an attack vector to further penetrate in the network.
- ✓ Rank vulnerabilities based on threat level, loss potential, and likelihood of exploitation.
- ✓ Perform supplemental research and development activities to support analysis.
- ✓ Identify issues of immediate consequence and recommend solutions.
- ✓ Develop long-term recommendation to enhance security level.

4. RISK CALCULATION

The observations in this report are rated with risk rating (where applicable), which should be interpreted as follows:

4.1. LIKELIHOOD

The likelihood describes the knowledge, skill and physical access that would be required of an attacker in order to identify and exploit vulnerability. The ease will describe if open source or commercially available tools are required for an attacker to exploit vulnerability. Additionally, the ease will note where an extended period is required to exploit the vulnerability, such as cracking weak encryption ciphers. Vulnerability is rated upon how easily it can be identified and exploited.

4.2. POTENTIAL IMPACT

The impact section describes what an attacker could achieve from exploiting the vulnerability. The impact of vulnerability is often defined by other configuration settings that could intensify the vulnerability or partially mitigate it. The impact is rated depending on the significance of the security threat.

4.3. RISK RATING

The risk rating section describes the potential impact if vulnerability is exploited by a threat source with given "Ease of Exploitation". The risk rating is calculated by multiplying "Likelihood" and "Potential Impact".

Rating	Description
Critical	The risk of vulnerability is Critical as either it is considerably easy to exploit, and the gain/impact is high. This vulnerability should be fixed on an urgent basis.
High	The risk of vulnerability is High as either it is considerably easy to exploit, or the gain/impact is high. This vulnerability should be fixed on an urgent basis.
Medium	The risk of vulnerability is medium as the ease of exploitation is Moderate and the resulting impact is considerably Moderate.
Low	The risk of vulnerability exposure is low because it requires considerable effort and skills and the resulting gain and impact is also Medium or low.

4.4. RISK DETERMINATION MATRIX

		Impact		
		High	Medium	Low
Likelihood	High	Critical	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Low

5. TECHNICAL DETAILS

5.1. ACTIVITIES PERFORMED

During the entire mobile applications penetration testing activity both manual and automated penetration testing was performed using the below-mentioned tools and techniques.

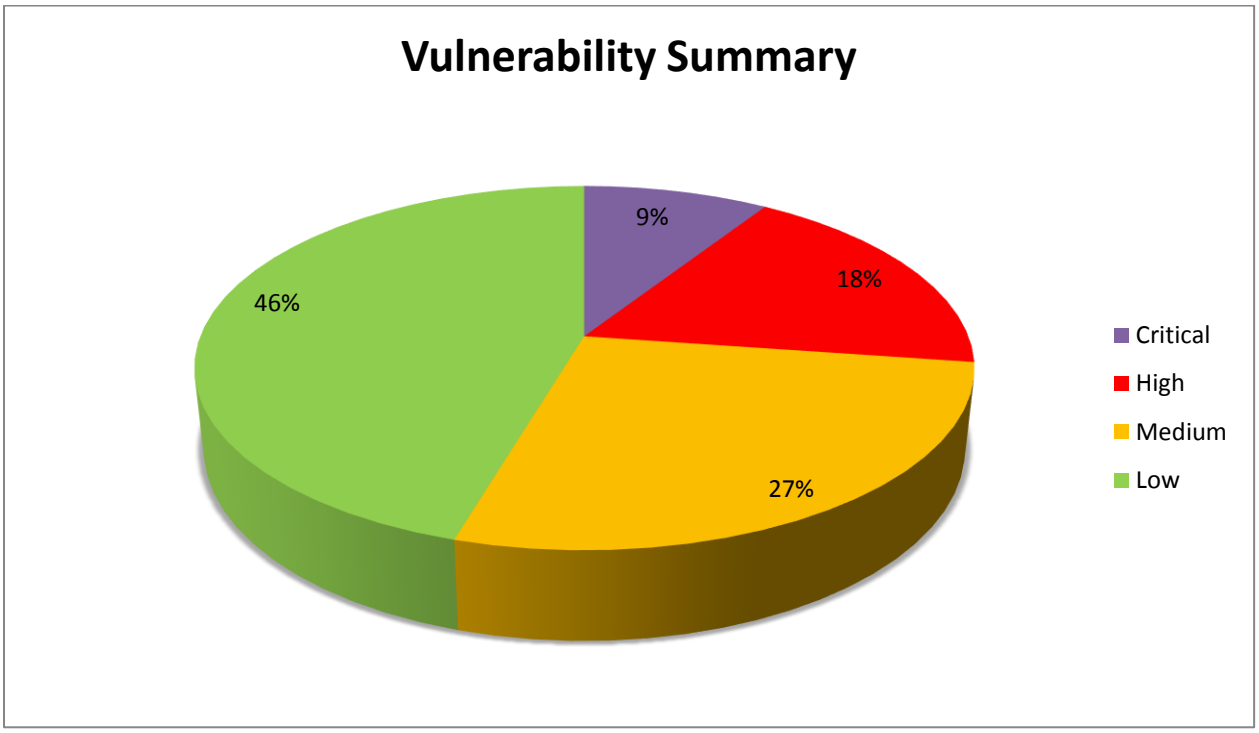
Description	Tools / Techniques
• Proxy	• BurpSuite

5.2. SCOPE OF PENETRATION TESTING

Target URL/App	Users' Permissions
khales.apk	No users provided
https://khales.paymobsolutions.com/admin	Admin and Viewer users provided

5.3. FINDINGS SUMMARY

Value	Count
Critical	1
High	2
Medium	3
Low	5



5.4. FINDINGS LIST

Finding Name	Risk
Full admin and user account takeover	Critical
Viewer user able to take over the admin's account	High
Changing the final service bill price to pay less or nothing	High
Bypass phone number verification	Medium
Unauthenticated access for some features and endpoints	Medium
XSS in HTML export feature in the admin panel	Medium
No rate limit for sending the OTP SMS	Low
No rate limit for the admin and user login	Low
Leakage for the important system information in error responses	Low
Missing important cookies attributes	Low
Bypass the password policies	Low

6. TECHNICAL FINDINGS LIST

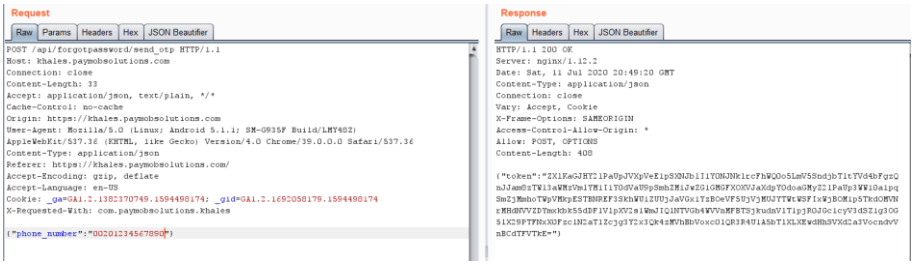
6.1. Full Admin and User Account Takeover

OWASP	Broken Access Control	Risk Rating	Critical
Likelihood	High	Impact	High
Affected App	khales.paymobsolutions.com khales.apk		
Observation	Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the limits of the user. Common access control vulnerabilities include bypassing access control checks by modifying the URL, internal application state, or the HTML page, or simply using a custom API attack tool, and from our side we have noticed that request of reset password is vulnerable to broken access control, the OTP code is not required to reset any user's password, the reset password request requires only Authorization token, and this token will be in the response of send OTP request. Reference: https://www.owasp.org/index.php/Top_10-2017_A5-Broken_Access_Control		
Implications	A successful exploit of this vulnerability can make the attacker who know the user's (user or admin) registered mobile number reset the user's password without any user interaction.		
Recommendation	We highly recommend • Use the OTP code as a parameter in the reset password request and validate its value • Generate the user's authorization token after the OTP validation, not before		

Evidence

Steps to reproduce:

1. Click **Forget password**
2. Enter the victim's mobile number and click **Reset**
3. Intercept the request



4. Write the generated token down
5. Use the token as **Authorization** header value in the request below

POST /api/forgotpassword/reset_password HTTP/1.1

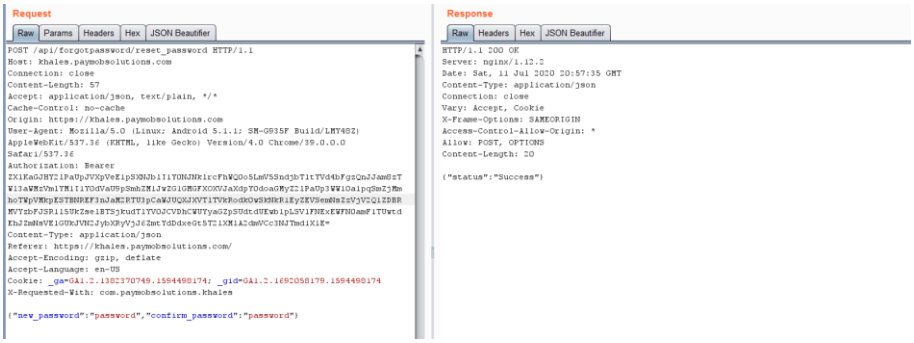
Host: khales.paymobsolutions.com

Authorization: Bearer [TOKEN]

Content-Type: application/json

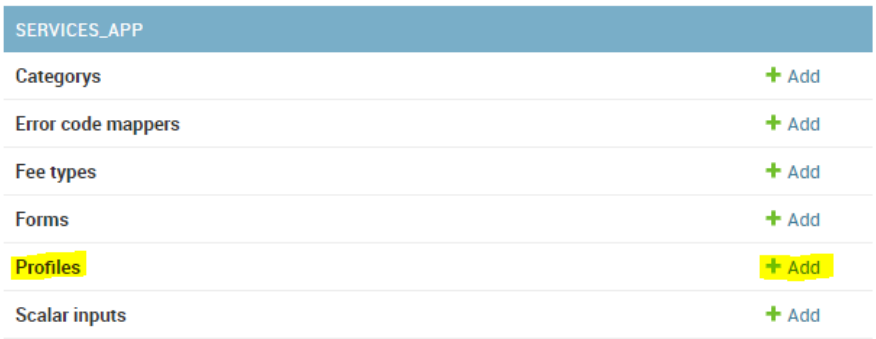
Content-Length: 51

{"new_password":"password","confirm_password":"password"}

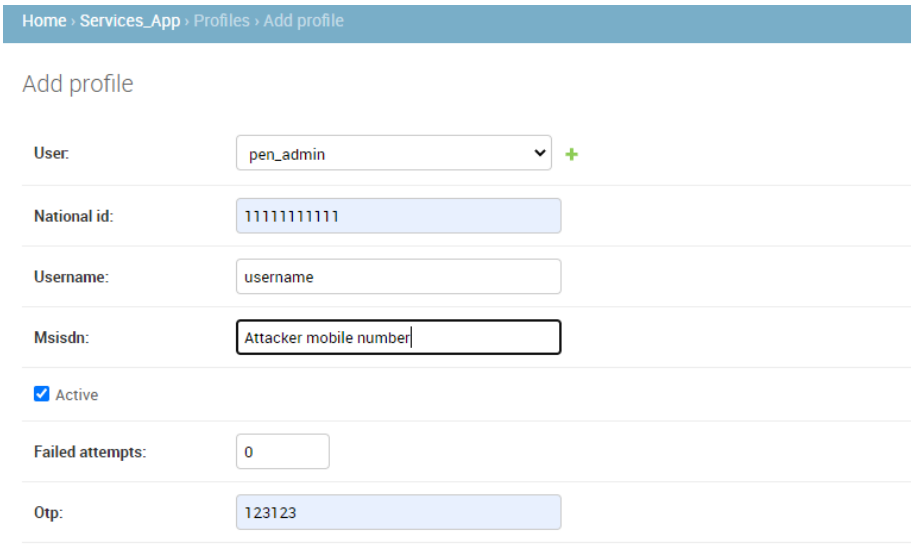


6. The user's password will be changed

6.2. Viewer User Able to take over the Admin's Account

OWASP	**Privilege escalation**	Risk Rating	High
Likelihood	High	Impact	Medium
Affected App	khales.paymobsolutions.com/admin		
Observation	Privilege escalation occurs when a user gets access to more resources or functionality than they are normally allowed, and such elevation or changes should have been prevented by the application. This is usually caused by a flaw in the application. The result is that the application performs actions with more privileges than those intended by the developer or system administrator, and from our side we have noticed that the provided account with view privileges (pen_view) able to create a profile for the admin user (pen_admin) and assign a mobile number to use it to reset the admin password and takeover the account. Reference: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/03-Testing_for_Privilege_Escalation		
Implications	A successful exploit of this vulnerability can make a viewer user change the admin's password to get a high privileges.		
Recommendation	Remove the add profile privilege from any user except the admin.		
Evidence	Steps to reproduce: 1. Go to https://khales.paymobsolutions.com/admin/ 2. Login pen_view account 3. Click Add next to profiles  4. Select the admin user in **User** input 5. Add a mobile number to receive the OTP code		

6. Fill the other inputs and click **Save**



7. Click **Save** button
8. Use the added mobile number to rest the admin password

6.3. Changing the Final Service Bill Price to Pay Less or Nothing

OWASP	Insecure Direct Object Reference	Risk Rating	High
Likelihood	Medium	Impact	High
Affected App	khales.paymobsolutions.com khales.apk		
Observation	Insecure Direct Object Reference (IDOR) occurs when an application exposes a reference to an internal implementation object. Using this way, it reveals the real identifier and format/pattern used of the element in the storage backend side, and from our side we have noticed that the **amount_cents** parameter in bill pay request is vulnerable to IDOR, the attacker able to change parameter value to pay less or nothing. Reference: https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html		
Implications	A successful exploit of this vulnerability can make the attacker change the total service bill value to pay less or nothing.		
Recommendation	We highly recommend • Use a reference id for each bill and pay process • Validate the **amount_cents** value and compare it with value stored in the database as a server side process		
Evidence	Steps to reproduce: 1. Select any service to pay 2. Enter the payment card info 3. Intercept the request 4. Change the **amount_cents** parameter value to less number, zero or negative number  5. Login to the admin panel 6. Go to Services_App > Transactions		

ID	OWNER	CREATED AT	MOBILE NUMBER	SERVICE NAME	PENDING	SUCCESS	BILL PAYMENT REFERENCE	AMOUNT WHOLE	TRANSACTION FEES
375	gnegm@mnztechnology.com	July 12, 2020, 12:40 a.m.	00201003262284	Electricity	✓	✗		0.0	2.5
374	gnegm@mnztechnology.com	July 12, 2020, 12:39 a.m.	00201003262284	Electricity	✓	✗		-10.0	0
373	gnegm@mnztechnology.com	July 12, 2020, 12:38 a.m.	00201003262284	Electricity	✓	✗		-0.01	0
372	gnegm@mnztechnology.com	July 12, 2020, 12:38 a.m.	00201003262284	Electricity	✓	✗		3.0	0
371	gnegm@mnztechnology.com	July 12, 2020, 12:38 a.m.	00201003262284	Electricity	✓	✗		1.0	0
370	gnegm@mnztechnology.com	July 12, 2020, 12:38 a.m.	00201003262284	Electricity	✓	✗		0.0	0
369	gnegm@mnztechnology.com	July 12, 2020, 12:37 a.m.	00201003262284	Electricity	✓	✗		0.0	0
368	gnegm@mnztechnology.com	July 12, 2020, 12:37 a.m.	00201003262284	Electricity	✓	✗		0.0	0
367	gnegm@mnztechnology.com	July 12, 2020, 12:37 a.m.	00201003262284	Electricity	✓	✗		0.0	0
366	gnegm@mnztechnology.com	July 12, 2020, 12:23 a.m.	00201003262284	Electricity	✓	✗		95.5	0

7. Transactions with less, zero and negative amount will be listed

6.4. Bypass Phone Number Verification

OWASP	**Insecure Direct Object Reference**	Risk Rating	Medium
Likelihood	Medium	Impact	Medium
Affected App	khales.paymobsolutions.com khales.apk		
Observation	Insecure Direct Object Reference (IDOR) occurs when an application exposes a reference to an internal implementation object. Using this way, it reveals the real identifier and format/pattern used of the element in the storage backend side, and from our side we have noticed that the **phone_number** and **otp** parameters are vulnaarble to IDOR, changing the both parameters values to the attacker's number and last otp sent to the number will activate an account registered using different number. Reference: https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html		
Implications	A successful exploit of this vulnerability can make the attacker create a new accounts with number he/she not own.		
Recommendation	We highly recommend: • Validate the OTP value with authorization header • Make sure that the phone number provided in the activation request is the same number provided in the signup process		
Evidence	Steps to reproduce: 1. Signup two accounts one with mobile number you don't own (account 1) and anther with mobile number you own (account 2) 2. Use the account 2 number and OTP received in the below request to activate account 1 POST /api/register/activation HTTP/1.1 Host: khales.paymobsolutions.com Accept: application/json, text/plain, */* Authorization: Bearer [account 1 token] {"otp":"123123","phone_number":"002012345678999"}		

Request

Raw | Params | Headers | Hex | JSON Beautifier

```
POST /api/register/activation HTTP/1.1
Host: khales.paymobsolutions.com
Connection: close
Content-Length: 46
Accept: application/json, text/plain, */*
Cache-Control: no-cache
Authorization: Bearer
ZXlKaGJHY2lPaUpJVXpVeEipSXNJb1I1YONJNklrcFhWQOo5LmV5SndjbT1tYVd4bFgzQnJJam8iTW13aWMz
VmlYMiI1YOdVaU9pSmhZMi1wZGiGMGFXOXVJaXdpY0dcaGMyZ2lPaUp3WW10aipqSmZjMmhoTWpVMkpESTBN
REF3SkdkNUirRldObT1pZVhveVNpUjFWWhFVnpONWJUZDRlVXMwWVdoS1VqSmJhSFUzYihrWVdXOXFjMnQ1
VjFaSWNqTjBLMkOwYUPSRiBTSjkuamUzZNQfmS1WFhWMGZpbOOxVmdkZGRUaXdWUkVWd3hTQOxXUDNibON1
SG9MdFf1TWiuWGFWcOF1em5LTVd3aE9bENHS2DyVHDVC1PVWNtHzYWWc=
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/83.0.4103.116 Safari/537.36
Content-Type: application/json
Origin: https://khales.paymobsolutions.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://khales.paymobsolutions.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,ar;q=0.7
Cookie: _ga=GA1.2.309937313.1594213955; _gid=GA1.2.1888446864.1594497048;
csrftoken=3ueTWlgZXeHf7OxgVfrAGcuBeWbkCimM;
sessionid=ah7ykwezj5eesa7dcc08xybonnfsnmgf

{"otp":"fdgl7jd","phone_number":"010123456789"}
```

Response

Raw | Headers | Hex | JSON Beautifier

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Sat, 11 Jul 2020 23:15:50 GMT
Content-Type: application/json
Connection: close
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
Access-Control-Allow-Origin: *
Allow: POST, OPTIONS
Content-Length: 20

{"status":"Success"}
```

6.5. Unauthenticated access for some features and endpoints

OWASP	**Broken Access Control**	Risk Rating	Medium
Likelihood	Medium	Impact	Medium
Affected App	khales.paymobsolutions.com/api		
Observation	Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the limits of the user, and from our side we have noticed that some API endpoint accessible without any authentication headers. Reference: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A5-Broken_Access_Control		
Implications	A successful exploit of this vulnerability can make the unauthenticated user use some of the application features and access data.		
Recommendation	We highly recommend • Make the Authorization header mandatory for every request • Validate the Authorization header value in all request methods GET, POST, PUT and DELETE		
Evidence	Steps to reproduce: 1. Go to https://khales.paymobsolutions.com/api/ 2. Click the listed links 3. Some of the links will respond with application data without any authentication 4. For example the endpoint /api/services will list all the services info		

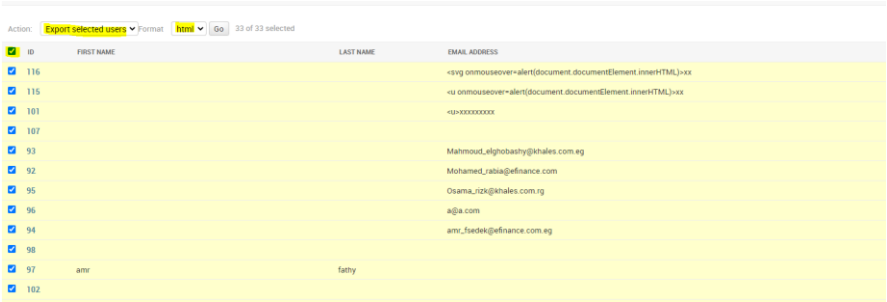
```
{
    count: 6,
    next: null,
    previous: null,
  - results: [
    - {
        id: 27,
      - forms: [
          - {
              id: 89,
            - inputs: [
                - {
                    id: 11,
                    _cls: "ScalarInput",
                    name: "stu",
                    variable_name: "s_Id",
                    custom_validator: "",
                    required: true,
                    place_holder: "sId",
                    place_holder_ar: "رقم الطالب",
                    type_hint: "text",
                    get_contacts: false,
                    regex_validate: ""
                  }
              ],
              title: "Admission Fees",
              title_ar: "",
              description: "",
              submit_text: "submit",
              submit_text_ar: "",
              image_url: "https://i.postimg.cc/fLGXsg2q/smartphone.png",
              image: null,
              name: "bill_reference"
            }
        ],
        category: "Universities-khales",
        provider: "khales",
        name: "Cairo University",
        name_ar: "جامعة القاهره",
        image_url: "https://i.postimg.cc/fLGXsg2q/smartphone.png",
        image: null,
        type: null,
        is_favorite: false
      },
    - {
        id: 6,
      - forms: [
          - {
              id: 12,
            - inputs: [
                - {
                    id: 3,
                    _cls: "ScalarInput",
                    name: "bill_reference",
                    variable_name: "bill_reference",
                    custom_validator: "",
```


results

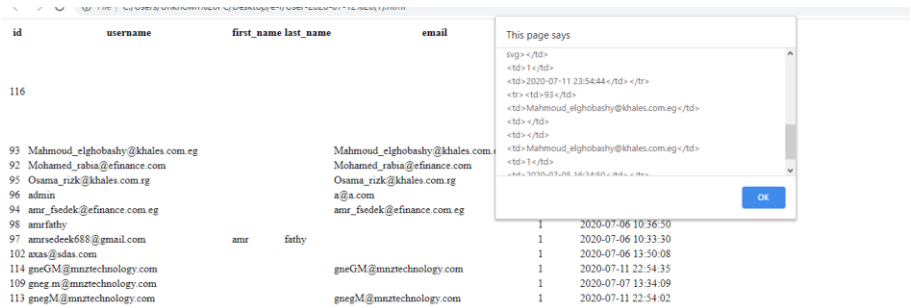
6.6. XSS in HTML Export Feature in the Admin Panel

OWASP	Cross-Site scripting	Risk Rating	Medium
Likelihood	Medium	Impact	Medium
Affected App	khales.paymobsolutions.com/admin		
Observation	Stored XSS occurs when a web application gathers input from a user which might be malicious, and then stores that input in a data store for later use. The input that is stored is not correctly filtered. As a consequence, the malicious data will appear to be part of the web site and run within the user's browser under the privileges of the web application. Since this vulnerability typically involves at least two requests to the application, this may also called second-order XSS, and from our side we have noticed that the HTML file generated form the html export feature in the admin panel in vulnerable to XSS, the attacker can register an account with username contains a malicious HTML and JS code to steal all generated file content. Reference: https://www.owasp.org/index.php/Testing_for_Stored_Cross_site_scripting_(OTG-INPVAL-002)		
Implications	A successful exploit of this vulnerability can make the attacker inject a malicious HTML and JS code to steal data from the admin side.		
Recommendation	We highly recommend implementing one of the following filtering techniques against the body of the emails: • Escaping. • Validating. • Sanitizing. Please refer to OWASP XSS prevention guide: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.md		
Evidence	Steps to reproduce: 1. Signup a new account 2. Intercept the request 3. Change the username to payload <svg onload=alert(document.documentElement.innerHTML)> 4. Login to the admin panel and go to Authentication and Authorization > Users		

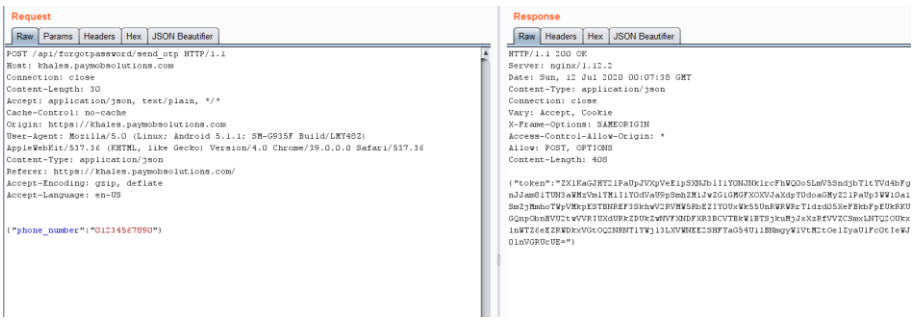
5. Select all user choice Export selected users and html format and click Go

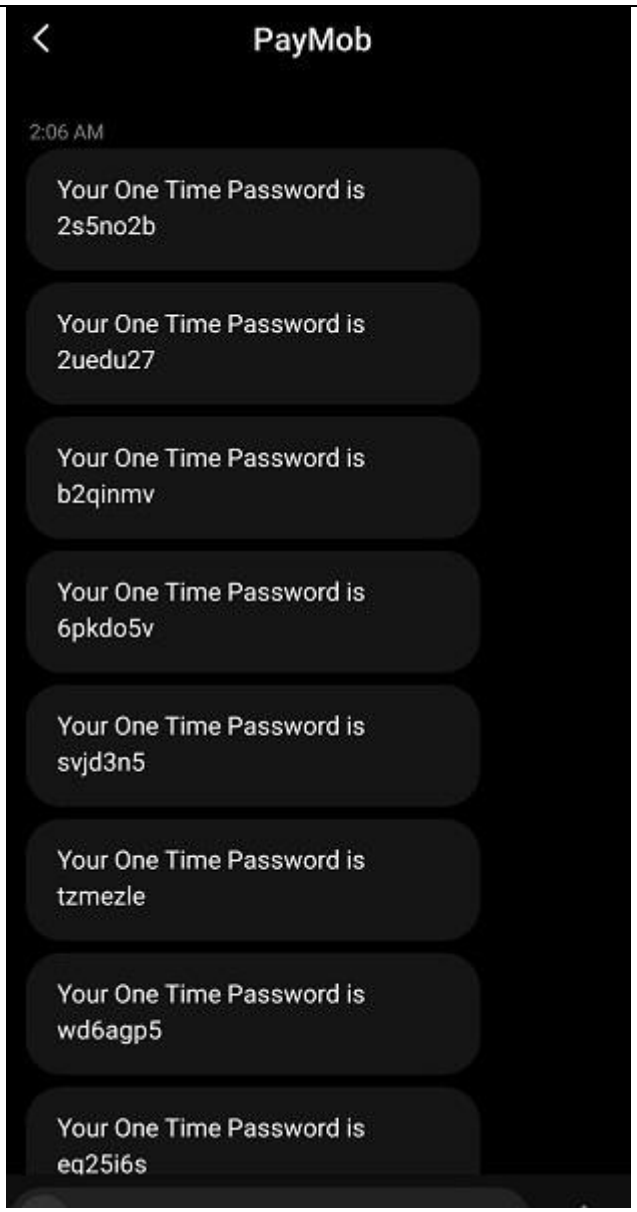


6. HTML file will be download
7. Open the file
8. The payload will be executed and alert all the user's data



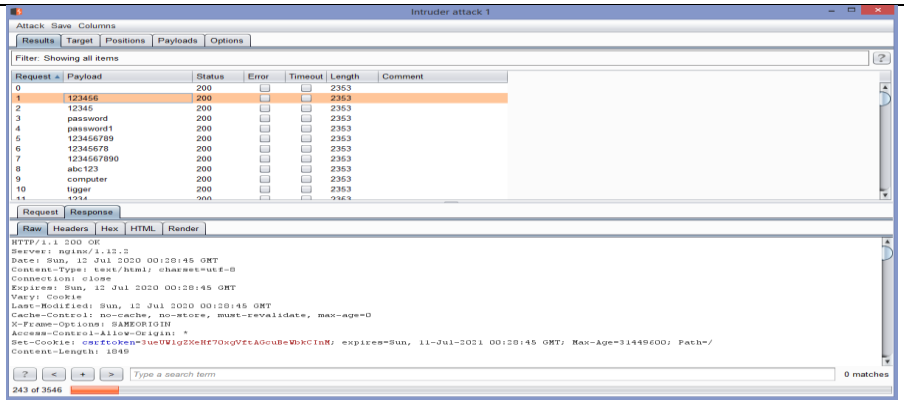
6.7. No rate limit for sending the OTP SMS

OWASP	**Rate limit**	Risk Rating	Low
Likelihood	Low	Impact	Low
Affected App	khales.paymobsolutions.com khales.apk		
Observation	Sending SMS messages incurs cost and affect the messages quota so it's should use a rate limit for sending a number of messages to the same number or in short time period, and from our side we have noticed that the rest OTP code and signup SMS messages have no rate limit and attacker able to send thousands of messages to the same number and affect the messages quota. Reference: https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html		
Implications	Attacker able to send thousands of messages to the same number in short time and affect the messages quota, costs more expenses or flood the user's SMS inbox with thousands of messages.		
Recommendation	We highly recommend to use a rate limit for sending a number of messages to the same number or in short time period.		
Evidence	Steps to reproduce: 1. Click Forget password 2. Enter a mobile number for an exist user and click **Reset** 3. Intercept the request by burp proxy 4. Send the request to the repeater  5. Repeat the request more and more 6. SMS message will be sent in short time without any rate limit		



6.8. No rate limit for the admin and user login

OWASP	Brute force attack	Risk Rating	Low
Likelihood	Low	Impact	Low
Affected App	khales.paymobsolutions.com khales.apk		
Observation	A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters and numbers, and from our side we have noticed that the admin and user login requests are vulnerable to brute force attack, the attacker can use a huge list of passwords to guess the admin and user passwords. Reference: https://www.owasp.org/index.php/Test_HTTP_Strict_Transport_Security_(OTG-CONFIG-007)		
Implications	A successful brute force exploit can lead the attacker to guess the password of the admins and users.		
Recommendation	We highly recommend • Adding a limit for submitting a wrong password • Add a captcha challenge to prevent the automated scripts from use this endpoint Reference: https://owasp.org/index.php/Blocking_Brute_Force_Attacks		
Evidence	Steps to reproduce: 1. Go to https://khales.paymobsolutions.com/admin 2. Open any proxy program like burpsuite 3. Login by the provided account with wrong password 4. send the request to the intruder 5. Strat the attack		



6. All the passwords will be tried without any rate limit

6.9. Leakage for the Important System Information in Error Responses

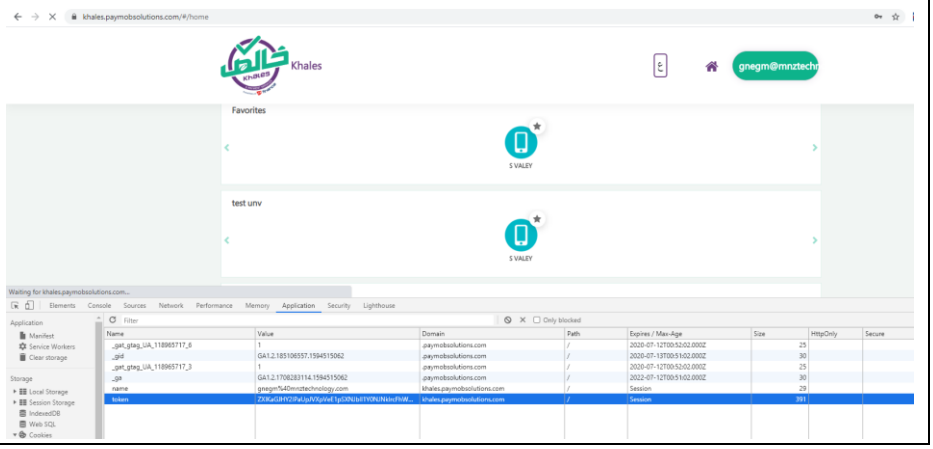
OWASP	Fingerprint Web Server	Risk Rating	Low
Likelihood	Low	Impact	Low
Affected App	khales.paymobsolutions.com khales.apk		
Observation	During our fingerprinting phase for the remote server, it was observed that the backend server reveals the some important system information in response of pages that return an error. Reference: https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTGINFO-002)		
Implications	Displaying version information of software information and some security configurations could allow an attacker to determine which vulnerabilities are present in the software, particularly if an outdated software version is in use with published vulnerabilities		
Recommendation	We highly recommend removing any error response which reveals any of backend and server information and configuration and replace it with text error message and error code.		
Evidence	Steps to reproduce: 1. Go to https://khales.paymobsolutions.com/api/updatedata 2. You will get an error response contain a sensitive info • Django Version • Exception Location • STATIC_ROOT • DATABASES • CORS_ORIGIN_WHITELIST • SERVER_EMAIL		

Settings

Using settings module `wallet_services.settings`

Setting	Value
JWT_SECRET	u'**********'
SECURE_BROWSER_XSS_FILTER	False
USE_X_FORWARDED_PORT	False
USE_THOUSAND_SEPARATOR	False
CSRF_COOKIE_SECURE	False
LANGUAGE_CODE	'en-us'
ROOT_URLCONF	'wallet_services.urls'
MANAGERS	[]
EMAIL_HOST_PASSWORD	u'**********'
SILENCED_SYSTEM_CHECKS	[]
DEFAULT_CHARSET	'utf-8'
SESSION_SERIALIZER	'django.contrib.sessions.serializers.JSONSerializer'
STATIC_ROOT	'/var/www/html/wallet-billpayment-services-backend/static/'
ALLOWED_HOSTS	['*']
MESSAGE_STORAGE	'django.contrib.messages.storage.fallback.FallbackStorage'
EMAIL_SUBJECT_PREFIX	'[Django] '
SERVER_EMAIL	'noreplyactivation6@gmail.com'
SECURE_HSTS_SECONDS	0
STATICFILES_FINDERS	['django.contrib.staticfiles.finders.FileSystemFinder', 'django.contrib.staticfiles.finders.AppDirectoriesFinder']
SESSION_CACHE_ALIAS	'default'
SESSION_COOKIE_DOMAIN	None
SESSION_COOKIE_NAME	'sessionid'
TIME_INPUT_FORMATS	['%H:%M:%S', '%H:%M:%S.%f', '%H:%M']
SECURE_REDIRECT_EXEMPT	[]
DATABASES	{'default': {'ATOMIC_REQUESTS': False, 'AUTOCOMMIT': True, 'CONN_MAX_AGE': 0, 'ENGINE': 'django.db.backends.sqlite3', 'HOST': '', 'NAME': '/var/www/html/wallet-billpayment-services-backend/db.sqlite3', 'OPTIONS': {}, 'PASSWORD': u'**********', 'PORT': '', 'TEST': {'CHARSET': None, 'COLLATION': None, 'MIRROR': None, 'NAME': None}, 'TIME_ZONE': None, 'USER': ''}}
EMAIL_SSL_KEYFILE	u'**********'
FILE_UPLOAD_DIRECTORY_PERMISSIONS	None
FILE_UPLOAD_PERMISSIONS	None
FILE_UPLOAD_HANDLERS	['django.core.files.uploadhandler.MemoryFileUploadHandler', 'django.core.files.uploadhandler.TemporaryFileUploadHandler']
DEFAULT_CONTENT_TYPE	'text/html'
APPEND_SLASH	False
FIRST_DAY_OF_WEEK	0
DATABASE_ROUTERS	[]
DEFAULT_TABLESPACE	''
YEAR_MONTH_FORMAT	'F Y'
STATICFILES_STORAGE	'django.contrib.staticfiles.storage.StaticFilesStorage'

6.10. Missing Important Cookies Attributes

OWASP	Fingerprint Web Server	Risk Rating	Low
Likelihood	Low	Impact	Low
Affected App	khales.paymobsolutions.com khales.apk		
Observation	HttpOnly and Secure are an additional flags included in a Set-Cookie HTTP response header. Using the HttpOnly flag when generating a cookie helps mitigate the risk of client-side script accessing the protected cookie (if the browser supports it), and from our side we have noticed that the HttpOnly and Secure attributes are missing in requests cookies Reference: https://www.owasp.org/index.php/HttpOnly		
Implications	Missing HttpOnly and Secure flags from HTTP response header, the cookie can be accessed through client-side script, even if a cross-site scripting (XSS) flaw exists, and a user accidentally accesses a link that exploits this flaw, the browser will not reveal the cookie to a third party.		
Recommendation	We highly recommend to add the HttpOnly and Secure flags in cookies.		
Evidence	Steps to reproduce: 1. login to user account 2. Navigate to any page and open the browser DevTool 3. You will see the HttpOnly and Secure flags are missing in cookies 		

6.11. Bypass the Password Policies

OWASP	Fingerprint Web Server	Risk Rating	Low
Likelihood	Low	Impact	Low
Affected App	khales.paymobsolutions.com/admin khales.apk		
Observation	A password policy is a set of rules designed to enhance computer security by encouraging users to employ strong passwords and use them properly. A password policy is often part of an organization's official regulations and may be taught as part of security awareness training, without enforcing password policy user account will be more vulnerable to brute-forcing attacks, and from our side, we have noticed that the password policy can be bypassed in the signup and rese password requests. Reference: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication		
Implications	A weak password policy increases the probability of an attacker having success using brute force and dictionary attacks against user accounts. An attacker who can determine user passwords can take over a user's account and potentially access sensitive data in the application.		
Recommendation	We highly recommend to double check the password if achieve all the password polices in both front end and backend.		
Evidence	Steps to reproduce: 1. Go to https://khales.paymobsolutions.com/#/register 2. Fill the inputs and set password with all policies 3. Intercept the request 4. Change password with any weak password “123”  5. Account will be registered with very weak password		

END OF REPORT