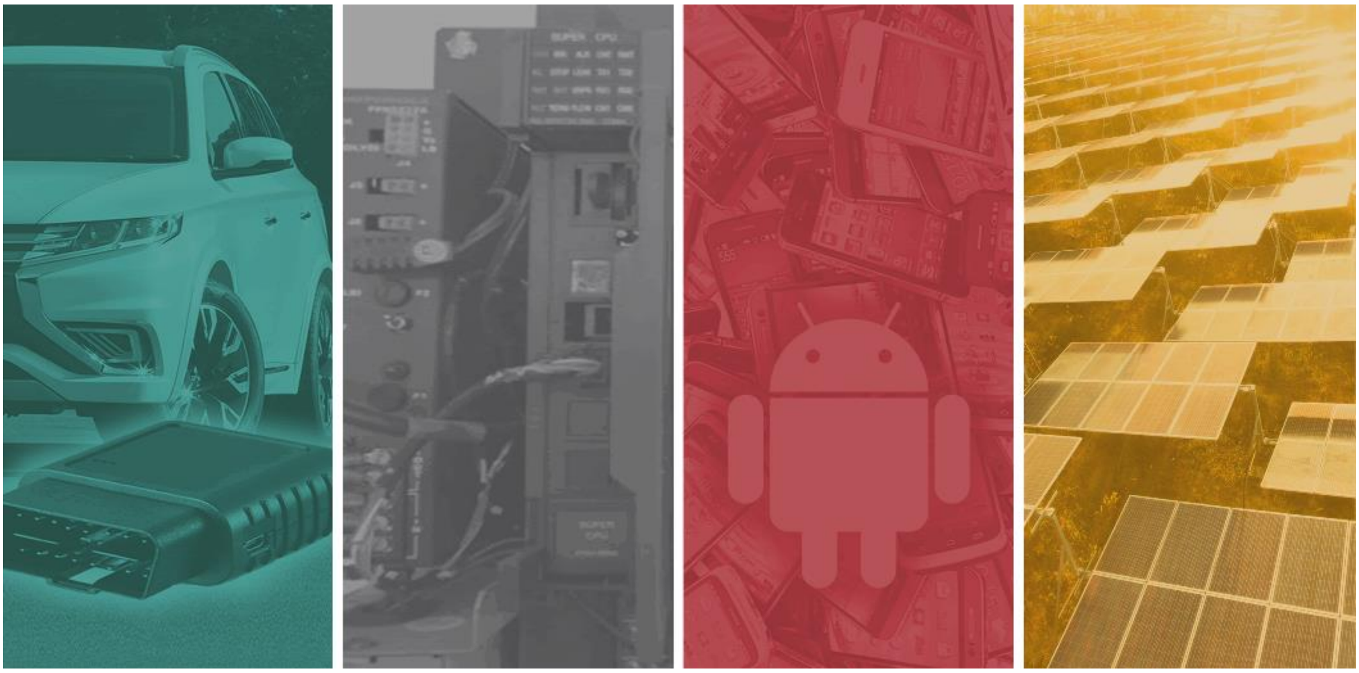
Euler dApp

for

Euler XYZ Limited



PTP Job ID: 12515
Version 1.0
14th June 2022

Technical Consultant: Charlie Moorton
Account Manager: Atif Raza

1. Business Risk Summary

Test	Risk
Euler dApp Web Application	🟩▼🟨🟨🟧🟧🟧🟥🟥🟪

Introduction

Euler XYZ Limited (Euler) engaged Pen Test Partners (PTP) to conduct a web application security assessment of their Euler dApp application, which allows users to connect to their crypto currency wallets and perform transactions. The aim of the engagement was to identify any security issues or vulnerabilities that may pose a risk to Euler's systems or users. Testing was conducted in line with PTP's web application security assessment methodology and OWASP guidelines.

Key Findings

Overall, the application has been developed to a good standard and was well protected against OWASP Top 10 common attacks. The general attack surface was very small, with the majority of the application functionality relying on API calls to a mixture of third party and Euler controlled endpoints.

A total of two issues were found during the assessment, with one medium and one low-risk finding identified. The medium-risk issue pertains to outdated TLS protocols and legacy ciphers currently being offered by the web application host, which could lead to known attacks such as LUCKY13 being carried out by an attacker. This does not pose an immediate threat to the application, due to the difficulty and complexity of executing these attacks.

A low-risk issue was also identified relating to a GraphQL endpoint used by the application and owned by Euler. The endpoint allowed remote users to run Introspection queries, which return detailed information on the possible operations and data types available within the API. While no vulnerabilities were found in the API, Euler should consider if this level of detailed information is necessary to be exposed.

Conclusion

Overall, the application was found to be well secured against attack, presenting low risk to both Euler's infrastructure and customer data. The main recommendation is that the TLS configuration of the application servers is strengthened by enforcing the use of known secure protocols and ciphers.

2. Technical Summary

The table below lists all security issues that were identified on systems within the scope of this test.

2.1. Euler dApp Web Application

Issue ID	Vulnerability	Hostnames	CVSS
WEB-M1	TLS Configuration Weaknesses	app.euler.finance	**4.2**
WEB-L2	GraphQL Introspection Enabled	app.euler.finance	**3.5**

3. Table of Contents

4. Document and Test Control

4.1. Version History

Version	Date	Author	Comment	Distribution
0.1	14/06/2022	Charlie Moorton	Author	
0.2	15/06/2022	Paul Brownridge	Technical QA	
0.3	17/06/2022	Carmen Hatch	Grammar QA	
1.0	**20/06/2022**	**Charlie Moorton**	**Release**	**Wojtek Zając**

4.2. Engagement Scope

Euler XYZ Limited engaged Pen Test Partners to perform a web application security assessment of their Euler dApp. This phase of testing took place between Wednesday 8th and Tuesday 14th June 2022 and was authorised by Wojtek Zając of Euler XYZ Limited.

This report details the following elements of work:

- Web application security assessment of https://app.euler.finance

While the application does interact with the blockchain using external services, APIs, and their own smart contracts, testing of third parties and the smart contracts was not in scope for this engagement.

The following consultant was involved in this engagement:

- Charlie Moorton - Security Consultant

No Denial-of-Service (DoS) attacks were performed.

5. Euler dApp Web Application

5.1. Introduction

The application consisted of a JavaScript front-end driving a series of calls to both third party and Euler owned APIs. In general, the Euler APIs retrieved market information while 3rd-party APIs were responsible for the transactions and interactions with the blockchain.

The application is designed to allow users to connect their wallets, for example with browser extension wallets. This provides two effective user roles: connected and unconnected. Unconnected users have access to a very limited view, displaying largely static market information:

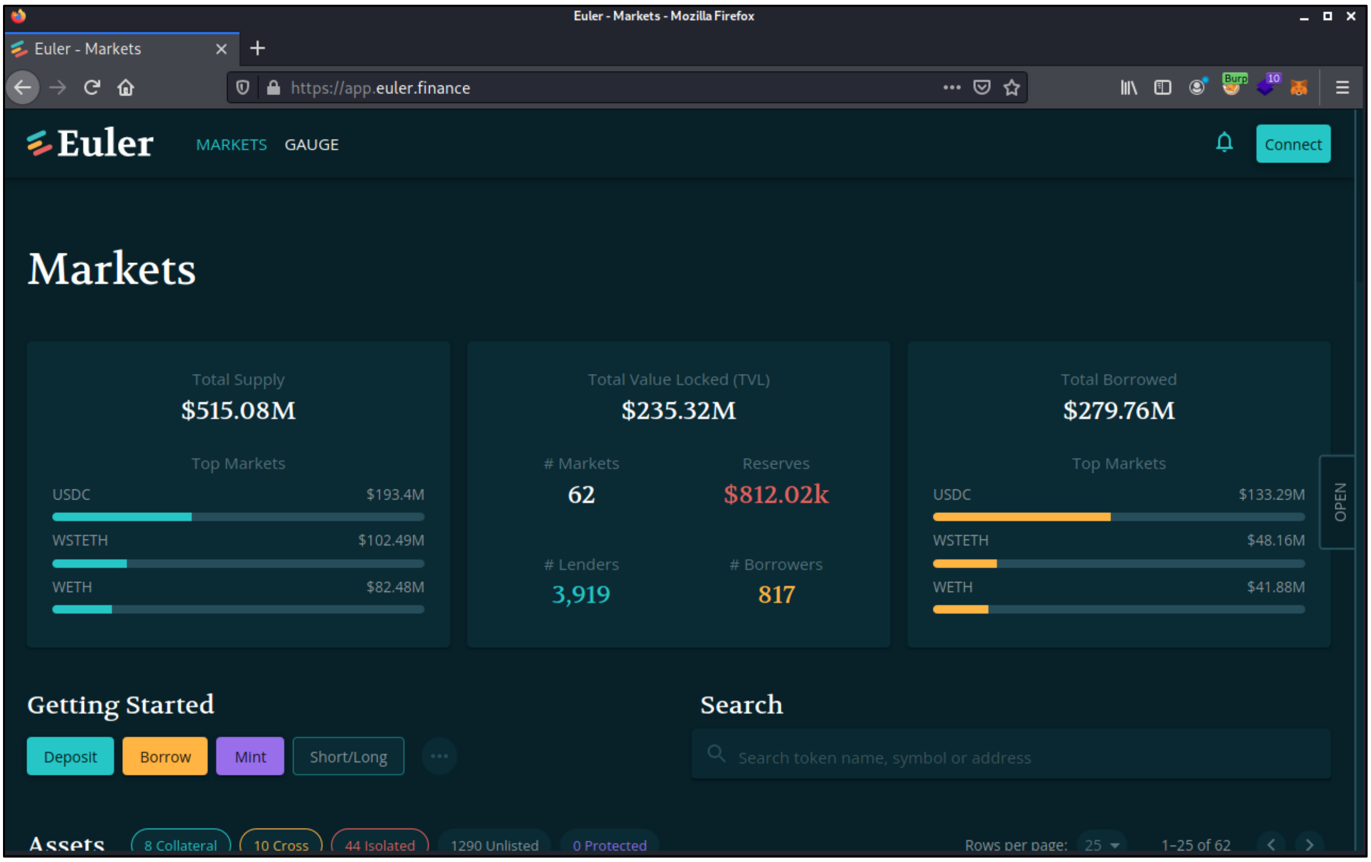
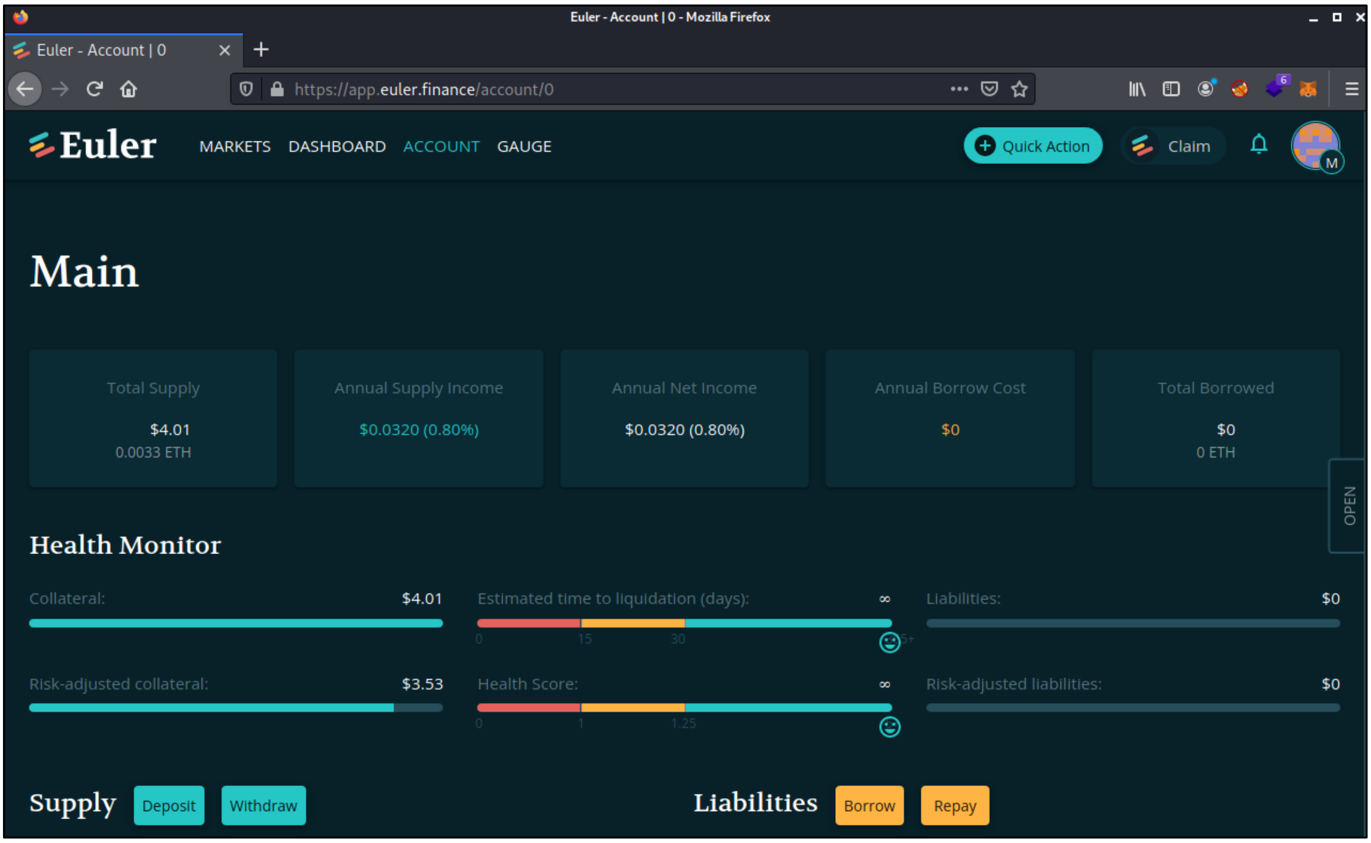


Figure 1 - View of the application from an unconnected perspective

When a wallet is connected, the application presents a number of additional screens, chiefly "*Account*" and "*Transaction Builder*". These screens retrieved and presented data about the user's wallet and balances, and allowed for the submission of blockchain transactions:



Main

Total Supply
$4.01
0.0033 ETH

Annual Supply Income
$0.0320 (0.80%)

Annual Net Income
$0.0320 (0.80%)

Annual Borrow Cost
$0

Total Borrowed
$0
0 ETH

Health Monitor

Collateral: $4.01

Estimated time to liquidation (days): ∞

Liabilities: $0

Risk-adjusted collateral: $3.53

Health Score: ∞

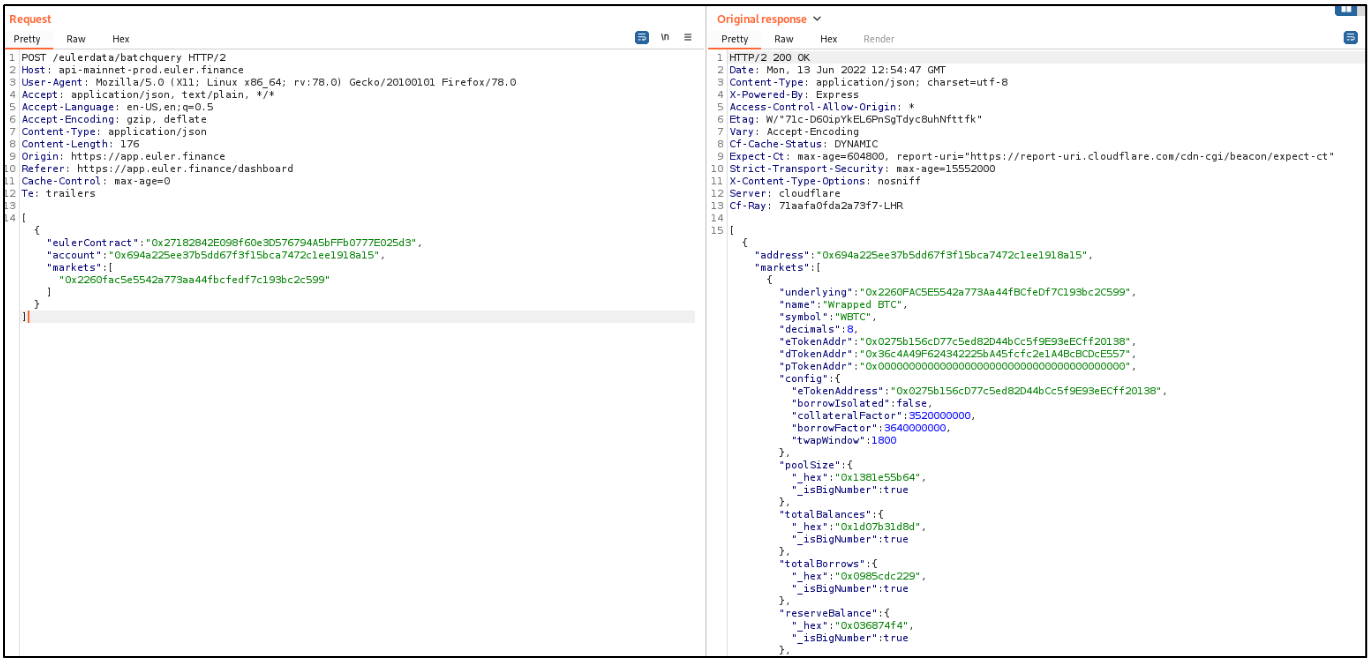
Risk-adjusted liabilities: $0

Supply

Liabilities

Figure 2 - View of the application from the perspective of a connected user, displaying the user's balance via the Account tab

Data for the application screens was retrieved via a number of APIs, with several controlled and developed by Euler. These included https://api-mainnet-prod.euler.finance and https://realm.mongo.db/api/client/v2.0/app/mainnet-realm-prod-aghkn/graphql:



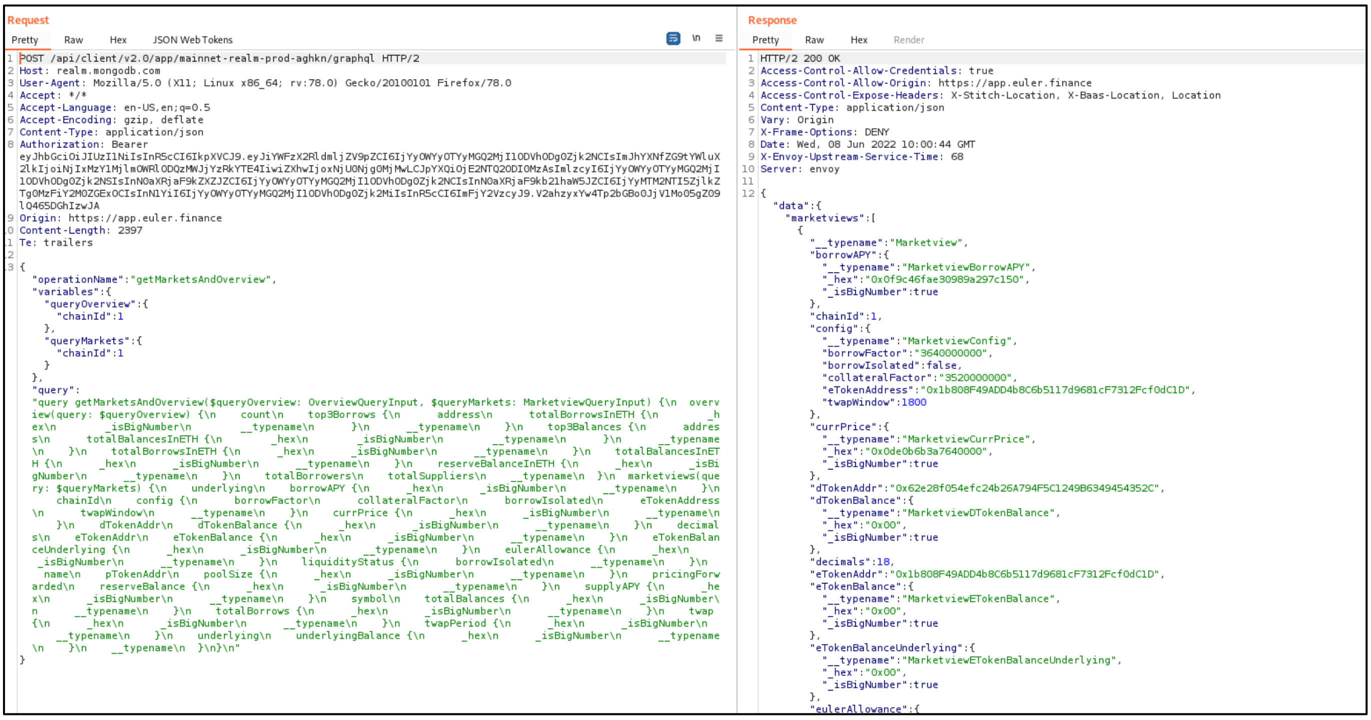
```
POST /eulerdata/batchquery HTTP/2
Host: api-mainnet-prod.euler.finance
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 176
Origin: https://app.euler.finance
Referer: https://app.euler.finance/dashboard
Cache-Control: max-age=0
Te: trailers

[
  {
    "eulerContract":"0x27182842E098f60e3D576794A5bFFb0777E025d3",
    "account":"0x694a225ee37b5dd67f3f15bca7472c1ee1918a15",
    "markets":[
      "0x2260fac5e5542a773aa44fbcfedf7c193bc2c599"
    ]
  }
]
```

```
HTTP/2 200 OK
Date: Mon, 13 Jun 2022 12:54:47 GMT
Content-Type: application/json; charset=utf-8
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"71c-D6OipYkEL6PnSgTdyc8uhNfttfk"
Vary: Accept-Encoding
Cf-Cache-Status: DYNAMIC
Expect-Ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Strict-Transport-Security: max-age=15552000
X-Content-Type-Options: nosniff
Server: cloudflare
Cf-Ray: 71aafa0fda2a73f7-LHR

[
  {
    "address":"0x694a225ee37b5dd67f3f15bca7472c1ee1918a15",
    "markets":[
      {
        "underlying":"0x2260FAC5E5542a773Aa44fBCfeDf7C193bc2C599",
        "name":"Wrapped BTC",
        "symbol":"WBTC",
        "decimals":8,
        "eTokenAddr":"0x0275b156cD77c5ed82D44bCc5f9E93eECff20138",
        "dTokenAddr":"0x36c4A49F624342225bA45fcfc2e1A4BcBCDcE557",
        "pTokenAddr":"0x0000000000000000000000000000000000000000",
        "config":{
          "eTokenAddress":"0x0275b156cD77c5ed82D44bCc5f9E93eECff20138",
          "borrowIsolated":false,
          "collateralFactor":3520000000,
          "borrowFactor":3640000000,
          "twapWindow":1800
        },
        "poolSize":{
          "_hex":"0x1381e55b64",
          "_isBigNumber":true
        },
        "totalBalances":{
          "_hex":"0x1d07b31d8d",
          "_isBigNumber":true
        },
        "totalBorrows":{
          "_hex":"0x0985cdc229",
          "_isBigNumber":true
        },
        "reserveBalance":{
          "_hex":"0x036874f4",
          "_isBigNumber":true
        },
```

Figure 3 - API request to /eulerdata/batchquery returning information about the user's wallet



```
POST /api/client/v2.0/app/mainnet-realm-prod-aghkn/graphql HTTP/2
Host: realm.mongodb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJiYWFzX2RldmljZV9pZCI6IjYyOWYyOTYyMGQ2MjI1ODVhODg0Zjk2NCIsImJhYXNfZG9tYWluX
2lkIjoiNjIxMzY1MjlmOWRlODQzMWJjYzRkYTE4IiwiZXhwIjoxNjU0Njg0MjMwLCJpYXQiOjE2NTQ2ODI0MzAsImlzcyI6IjYyOWYyOTYyMGQ2MjI
1ODVhODg0Zjk2NSIsInN0aXRjaF9kZXZJZCI6IjYyOWYyOTYyMGQ2MjI1ODVhODg0Zjk2NCIsInN0aXRjaF9kb21haW5JZCI6IjYyMTM2NTI5ZjlkZ
Tg0MzFiY2M0ZGExOCIsInN1YiI6IjYyOWYyOTYyMGQ2MjI1ODVhODg0Zjk2MiIsInR5cCI6ImFjY2VzcyJ9.V2ahzyxYw4Tp2bGBo0JjV1Mo05gZ09
lQ465DGhIzwJA
Origin: https://app.euler.finance
Content-Length: 2397
Te: trailers

{
  "operationName":"getMarketsAndOverview",
  "variables":{
    "queryOverview":{
      "chainId":1
    },
    "queryMarkets":{
      "chainId":1
    }
  },
  "query":
  "query getMarketsAndOverview($queryOverview: OverviewQueryInput, $queryMarkets: MarketviewQueryInput) {\n  overv
  iew(query: $queryOverview) {\n    count\n    top3Borrows {\n      address\n      totalBorrowsInETH {\n        _h
  ex\n        _isBigNumber\n        __typename\n      }\n      __typename\n    }\n    top3Balances {\n      addres
  s\n      totalBalancesInETH {\n        _hex\n        _isBigNumber\n        __typename\n      }\n      __typename
  \n    }\n    totalBorrowsInETH {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    totalBalancesInET
  H {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    reserveBalanceInETH {\n      _hex\n      _isBi
  gNumber\n      __typename\n    }\n    totalBorrowers\n    totalSuppliers\n    __typename\n  }\n  marketviews(que
  ry: $queryMarkets) {\n    underlying\n    borrowAPY {\n      _hex\n      _isBigNumber\n      __typename\n    }\n
    chainId\n    config {\n      borrowFactor\n      collateralFactor\n      borrowIsolated\n      eTokenAddress
  \n      twapWindow\n      __typename\n    }\n    currPrice {\n      _hex\n      _isBigNumber\n      __typename\n
    }\n    dTokenAddr\n    dTokenBalance {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    decimal
  s\n    eTokenAddr\n    eTokenBalance {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    eTokenBalan
  ceUnderlying {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    eulerAllowance {\n      _hex\n
   _isBigNumber\n      __typename\n    }\n    liquidityStatus {\n      borrowIsolated\n      __typename\n    }\n
    name\n    pTokenAddr\n    poolSize {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    pricingForw
  arded\n    reserveBalance {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    supplyAPY {\n      _he
  x\n      _isBigNumber\n      __typename\n    }\n    symbol\n    totalBalances {\n      _hex\n      _isBigNumber\
  n      __typename\n    }\n    totalBorrows {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    twap
  {\n      _hex\n      _isBigNumber\n      __typename\n    }\n    twapPeriod {\n      _hex\n      _isBigNumber\n
      __typename\n    }\n    underlying\n    underlyingBalance {\n      _hex\n      _isBigNumber\n      __typename
  \n    }\n    __typename\n  }\n}\n"
}
```

```
HTTP/2 200 OK
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://app.euler.finance
Access-Control-Expose-Headers: X-Stitch-Location, X-Baas-Location, Location
Content-Type: application/json
Vary: Origin
X-Frame-Options: DENY
Date: Wed, 08 Jun 2022 10:00:44 GMT
X-Envoy-Upstream-Service-Time: 68
Server: envoy

{
  "data":{
    "marketviews":[
      {
        "__typename":"Marketview",
        "borrowAPY":{
          "__typename":"MarketviewBorrowAPY",
          "_hex":"0x0f9c46fae30989a297c150",
          "_isBigNumber":true
        },
        "chainId":1,
        "config":{
          "__typename":"MarketviewConfig",
          "borrowFactor":"3640000000",
          "borrowIsolated":false,
          "collateralFactor":"3520000000",
          "eTokenAddress":"0x1b808F49ADD4b8C6b5117d9681cF7312Fcf0dC1D",
          "twapWindow":1800
        },
        "currPrice":{
          "__typename":"MarketviewCurrPrice",
          "_hex":"0x0de0b6b3a7640000",
          "_isBigNumber":true
        },
        "dTokenAddr":"0x62e28f054efc24b26A794F5C1249B6349454352C",
        "dTokenBalance":{
          "__typename":"MarketviewDTokenBalance",
          "_hex":"0x00",
          "_isBigNumber":true
        },
        "decimals":18,
        "eTokenAddr":"0x1b808F49ADD4b8C6b5117d9681cF7312Fcf0dC1D",
        "eTokenBalance":{
          "__typename":"MarketviewETokenBalance",
          "_hex":"0x00",
          "_isBigNumber":true
        },
        "eTokenBalanceUnderlying":{
          "__typename":"MarketviewETokenBalanceUnderlying",
          "_hex":"0x00",
          "_isBigNumber":true
        },
        "eulerAllowance":{
```

Figure 4 - GraphQL query to https://realm.mongo.db/api/client/v2.0/app/mainnet-realm-prod-aghkn/graphql returning market data

5.2. WEB-M1: TLS Configuration Weaknesses

Multiple configuration issues were identified when attempting to establish a secure connection with the target applications or hosts. The use of deprecated protocols, weak ciphers, or untrusted certificates could aid an adversary in undermining the integrity or confidentiality of encrypted communications.

All TLS services should be reconfigured in-line with security best practices, using the TLS 1.2 or TLS 1.3 protocol and offering only AES-GCM or CHACHA20-POLY1305 ciphers. Certificate should be obtained exclusively from trusted authorities.

Medium Risk
CVSS 4.2

Description

When establishing a connection to a TLS listener, the client and server negotiate to determine the strongest cipher that is supported by both. Clients often support a wide range of protocol and cipher options to facilitate convenient access to a broad array of websites, applications, and platforms. Some of these options are susceptible to publicly disclosed vulnerabilities and, in some cases, have been subject to deprecation due to the severity of these issues. If the server or application has not been configured securely, it may permit a connection which uses these weaker protocols or ciphers.

Additionally, the listener can be configured to allow or deny certain practices such as downgrading to a connection with weaker options due to latency or connectivity issues.

When evaluating the application in scope, it was discovered that the protocols, ciphers, or features advertised by the TLS services were affected by published vulnerabilities, had been deprecated, or did not otherwise meet the standards for security best practice.

The affected hosts were found to be vulnerable to the issues as described below:

- **Protocols with known weaknesses allowed**: TLS protocols version 1.0/1.1 are known to contain several design and implementation faults and should not be used. An attacker may be able to exploit these issues to conduct interception attacks or increase its chances of successfully decrypting communications between the affected parties. TLS protocols versions 1.0 and 1.1 have proof-of-concept attacks that exploit some of these flaws that have been published.

- **Susceptibility to LUCKY13 attack (supports CBC encryption cipher suites):** Cryptographic timing attack against implementations of the TLS protocol that use the CBC mode of operation.

- **Susceptibility to SWEET32 attack:** The SSL vulnerability known as SWEET32 takes advantage of the use of block ciphers with 64-bit blocks in one or more cipher suites. As a result, it may be possible to launch a man-in-the-middle (MiTM) attack with sufficient resources to then potentially break the end-to-end encryption, and disclose sensitive information such as secure session cookies.

```
Testing all IPv4 addresses (port 443): 172.67.37.200 104.22.5.126 104.22.4.126
-------------------------------------------------------------------------------------
-----------------------------------------------------------------------------
 Start 2022-06-13 10:49:48        -->> 172.67.37.200:443 (app.euler.finance) <<--

 Further IP addresses:   104.22.5.126 104.22.4.126 2606:4700:10::ac43:25c8
2606:4700:10::6816:57e 2606:4700:10::6816:47e
```

```
 rDNS (172.67.37.200):    --
 Service detected:        HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      offered
 TLS 1.1    offered
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   h2, http/1.1 (advertised)
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES encryption (w/o export)     not offered (OK)
 Weak 128 Bit ciphers (SEED, IDEA, RC[2,4])    not offered (OK)
 Triple DES Ciphers (Medium)                   offered
 High encryption (AES+Camellia, no AEAD)       offered (OK)
 Strong encryption (AEAD ciphers)              offered (OK)

Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat
extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session tickets
 ROBOT                                     not vulnerable (OK)
 Secure Renegotiation (CVE-2009-3555)      not vulnerable (OK)
 Secure Client-Initiated Renegotiation     not vulnerable (OK)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    potentially NOT ok, uses gzip HTTP
compression. - only supplied "/" tested
                                           Can be ignored for static pages or if no
secrets in the page
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK)
 TLS_FALLBACK_SCSV (RFC 7507)              Downgrade attack prevention supported
(OK)
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    VULNERABLE, uses 64 bit block ciphers
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port
(OK)
                                           make sure you don't use this certificate
elsewhere with SSLv2 enabled services

https://censys.io/ipv4?q=F31612C9BEFC1D565FFEF5AB231F69D3F9DE206EF90A90E90E9E14AF25
F11D9E could help you to find out
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT
ciphers, no DH key detected
BEAST (CVE-2011-3389)                     TLS1: ECDHE-RSA-AES128-SHA AES128-SHA
ECDHE-RSA-AES256-SHA AES256-SHA DES-CBC3-SHA
                                           VULNERABLE -- but also supports higher
protocols  TLSv1.1 TLSv1.2 (likely mitigated)
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses cipher
block chaining (CBC) ciphers with TLS. Check patches
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)

Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits     Cipher
Suite Name (RFC)
-----------------------------------------------------------------------------------
------------------------------------------
```

```
 x1302   TLS_AES_256_GCM_SHA384            ECDH 253   AESGCM      256
TLS_AES_256_GCM_SHA384
 x1303   TLS_CHACHA20_POLY1305_SHA256      ECDH 253   ChaCha20    256
TLS_CHACHA20_POLY1305_SHA256
 xcc14   ECDHE-ECDSA-CHACHA20-POLY1305-OLD ECDH 256   ChaCha20    256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256_OLD
 xcc13   ECDHE-RSA-CHACHA20-POLY1305-OLD   ECDH 256   ChaCha20    256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256_OLD
 xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 256   AESGCM      256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 xc02c   ECDHE-ECDSA-AES256-GCM-SHA384     ECDH 256   AESGCM      256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
 xc028   ECDHE-RSA-AES256-SHA384           ECDH 256   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
 xc024   ECDHE-ECDSA-AES256-SHA384         ECDH 256   AES         256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
 xc014   ECDHE-RSA-AES256-SHA              ECDH 256   AES         256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
 xc00a   ECDHE-ECDSA-AES256-SHA            ECDH 256   AES         256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
 xcca9   ECDHE-ECDSA-CHACHA20-POLY1305     ECDH 253   ChaCha20    256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
 xcca8   ECDHE-RSA-CHACHA20-POLY1305       ECDH 253   ChaCha20    256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
 x9d     AES256-GCM-SHA384                 RSA        AESGCM      256
TLS_RSA_WITH_AES_256_GCM_SHA384
 x3d     AES256-SHA256                     RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA256
 x35     AES256-SHA                        RSA        AES         256
TLS_RSA_WITH_AES_256_CBC_SHA
 x1301   TLS_AES_128_GCM_SHA256            ECDH 253   AESGCM      128
TLS_AES_128_GCM_SHA256
 xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 256   AESGCM      128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 xc02b   ECDHE-ECDSA-AES128-GCM-SHA256     ECDH 256   AESGCM      128
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
 xc027   ECDHE-RSA-AES128-SHA256           ECDH 256   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
 xc023   ECDHE-ECDSA-AES128-SHA256         ECDH 256   AES         128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
 xc013   ECDHE-RSA-AES128-SHA              ECDH 256   AES         128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
 xc009   ECDHE-ECDSA-AES128-SHA            ECDH 256   AES         128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
 x9c     AES128-GCM-SHA256                 RSA        AESGCM      128
TLS_RSA_WITH_AES_128_GCM_SHA256
 x3c     AES128-SHA256                     RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA256
 x2f     AES128-SHA                        RSA        AES         128
TLS_RSA_WITH_AES_128_CBC_SHA
 x0a     DES-CBC3-SHA                      RSA        3DES        168
TLS_RSA_WITH_3DES_EDE_CBC_SHA
```

Figure 5 – Selected output from testssl.sh

Table 1 - TLS Configuration Weaknesses

Vulnerability Host:Port	SWEET32	TLS1.0 Support	TLS1.1 Support	LUCKY13
app.euler.finance	✗	✗	✗	✗

Recommendations

Multiple issues relating to the TLS configuration of the affected hosts or services were detected and are identified separately above. However, mitigation of these issues overlaps significantly, and remediation of individual items may, in some cases, be insufficient in resolving broader transport security concerns. Therefore, this recommendation incorporates a guideline for configuring TLS services against a known good configuration which is not susceptible to any known issues.

The following settings should be applied for all services offering TLS. Currently, these measures are considered the most robust compromise of security versus client compatibility.

Patching

- Ensure that all software is fully patched; in particular (but not limited to), TLS/SSL shared libraries (e.g., OpenSSL, GnuTLS, wolfSSL), to mitigate implementation vulnerabilities.

Certificates

- All certificates should be valid and signed by a trusted Certification Authority and signed with a strong hashing algorithm such as SHA-256.
- All certificates should be signed to a specific fully qualified domain name (FQDN). Wildcard and alternative names should be avoided when possible.

Protocols

- TLS 1.3 should be enabled and set to preferred; however, if used exclusively, this could incur a penalty around compatibility with the client base.
- TLS 1.2 should be enabled to ensure broader compatibility.
- TLS 1.1, TLS 1.0, SSL 3.0, and SSL 2.0 should be disabled.

Cipher suites

- AEAD cipher suites should be enabled and set to preferred. These should be used exclusively where client compatibility is not affected significantly. All AEAD cipher suites implement Perfect Forward Secrecy (PFS).
- GCM based ciphers should be enabled and set to preferred.
- All weak strength cipher sets should be disabled (any cipher using keys of 128 bits or shorter).
- 64-bit block ciphers should never be permitted (DES, 3DES and Blowfish).
- All cipher sets that rely on RSA for encryption-based key exchange should be disabled (*TLS_RSA_** or *RSA-**).
- Export grade Diffie-Hellman (*DHE_EXPORT*) key exchange keys and RSA cipher sets (*RSA_EXPORT*) should be disabled.
- PFS cipher sets using Elliptic Curve Diffie-Hellman Exchange (ECDHE) should be enabled and set to preferred.
    - PFS cipher sets using Diffie-Hellman Exchange (DHE) should be disabled due to potentially using weak public keys (1024 bits).

The list below contains those cipher suites which are not affected by issues known at the time of writing this report and which meet industry best practice guidelines. Please review the References section for more details.

- TLS 1.3
    - TLS-CHACHA20-POLY1305-SHA256
    - TLS-AES-256-GCM-SHA384
    - TLS-AES-128-GCM-SHA256
- TLS 1.2 (ideal, all AEAD cipher suites)
    - ECDHE-ECDSA-CHACHA20-POLY1305-SHA256
    - ECDHE-ECDSA-CHACHA20-POLY1305
    - ECDHE-ECDSA-AES256-GCM-SHA384

    - ECDHE-ECDSA-AES128-GCM-SHA256
    - ECDHE-RSA-CHACHA20-POLY1305
- TLS 1.2 (older, more compatible cipher suites, all AEAD)
    - DHE-RSA-AES256-GCM-SHA384
    - DHE-RSA-AES128-GCM-SHA256
    - ECDHE-RSA-AES256-GCM-SHA384
    - ECDHE-RSA-AES128-GCM-SHA256

Affected	app.euler.finance
References & CVSSv3 Metrics	Mozilla: Server Side TLS: https://wiki.mozilla.org/Security/Server_Side_TLS SSL and TLS Deployment Best Practices: https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices Sweet32: https://sweet32.info/ Using TLS to protect data - NCSC: https://www.ncsc.gov.uk/guidance/tls-external-facing-services Root Cause: Encryption Base Metrics: AV:N/AC:H/PR:H/UI:R/S:U/C:H/I:L/A:N (4.8) Temporal Metrics: E:U/RL:O/RC:C (4.2) Environmental Metrics: CR:M/IR:M/AR:M (4.2)

5.3. WEB-L2: GraphQL Introspection Enabled

Introspection was found to be enabled on the GraphQL endpoint. This discloses a detailed enumeration of the implementation's data types and schema. This could allow an attacker to discover and exploit weaknesses in the API.

Disable introspection if possible.

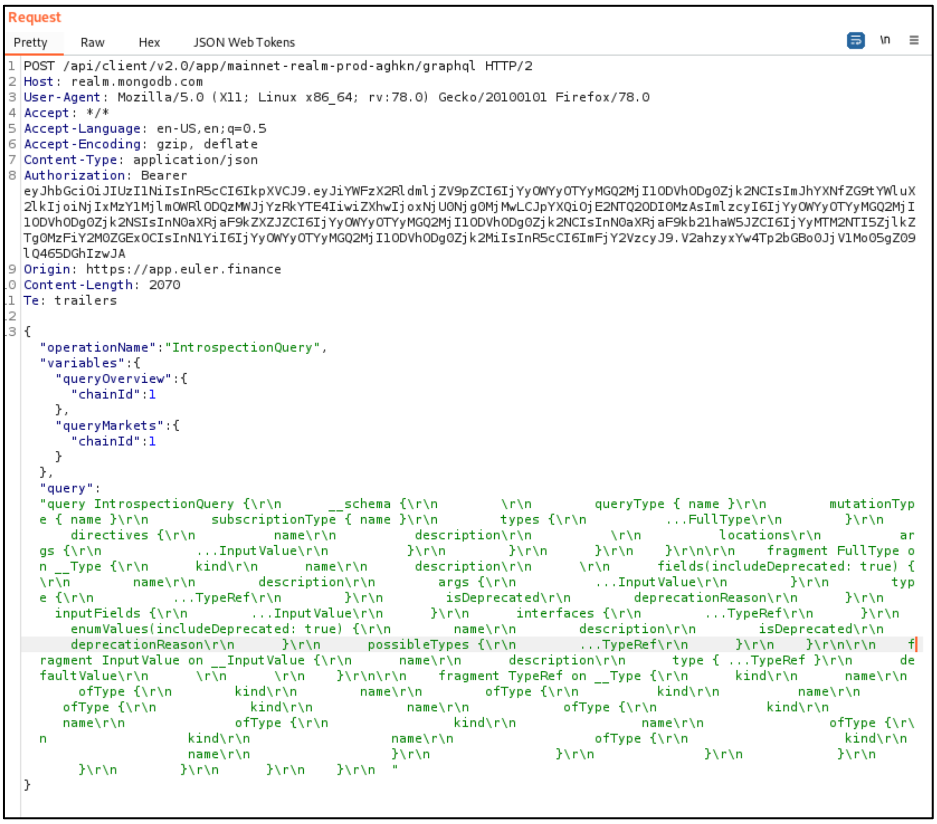
Low Risk
CVSS 3.5

Description

GraphQL based APIs use a method called introspection to publish details about the data set schema and data types required to operate the API. This enumerates all queries and mutations and any associated fields and data types involved that the API implements.

Conceptually, this is only an information leak, since obscuring the functionality and implementation details of the API should be relied on as a security feature. However, should the API have weaknesses, they will be found much more effectively by analysing this information. Conversely, if the API is securely implemented, it is of no material consequence to disclose this information, as access controls and other protection layers would ensure data is delivered to authorised users only.

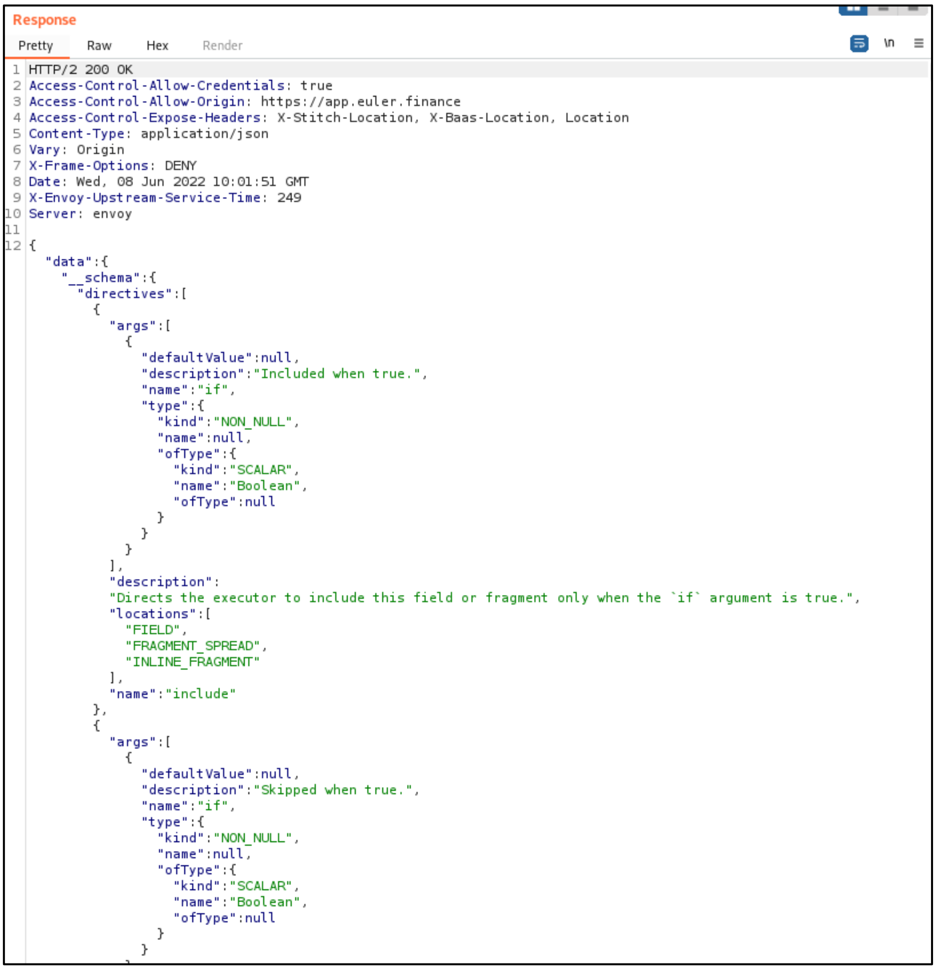
During the engagement, no means to compromise the GraphQL endpoint was discovered. The access tokens provided to the user did not have adequate permissions to perform state changing mutations, while queries returned only public information.



```
POST /api/client/v2.0/app/mainnet-realm-prod-aghkn/graphql HTTP/2
Host: realm.mongodb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJiYWFzX2RldmljZV9pZCI6IjYyOWYyOTYyMGQ2MjI1ODVhODg0Zjk2NCIsImJhYXNfZG9tYWluX
2lkIjoiNjIxMzY1MjlmOWRlODQzMWJjYzRkYTE4IiwiZXhwIjoxNjU0Njg0MjMwLCJpYXQiOjE2NTQ2ODI0MzAsImlzcyI6IjYyOWYyOTYyMGQ2MjI
1ODVhODg0Zjk2NSIsInN0aXRjaF9kZXZJZCI6IjYyOWYyOTYyMGQ2MjI1ODVhODg0Zjk2NCIsInN0aXRjaF9kb21haW5JZCI6IjYyMTM2NTI5ZjlkZ
Tg0MzFiY2M0ZGExOCIsInN1YiI6IjYyOWYyOTYyMGQ2MjI1ODVhODg0Zjk2MiIsInR5cCI6ImFjY2VzcyJ9.V2ahzyxYw4Tp2bGBo0JjV1Mo05gZ09
lQ465DGhIzwJA
Origin: https://app.euler.finance
Content-Length: 2070
Te: trailers

{
  "operationName":"IntrospectionQuery",
  "variables":{
    "queryOverview":{
      "chainId":1
    },
    "queryMarkets":{
      "chainId":1
    }
  },
  "query":
  "query IntrospectionQuery {\r\n      __schema {\r\n        \r\n        queryType { name }\r\n        mutationTyp
  e { name }\r\n        subscriptionType { name }\r\n        types {\r\n          ...FullType\r\n        }\r\n
    directives {\r\n          name\r\n          description\r\n          \r\n          locations\r\n          ar
  gs {\r\n            ...InputValue\r\n          }\r\n        }\r\n      }\r\n    }\r\n\r\n    fragment FullType o
  n __Type {\r\n      kind\r\n      name\r\n      description\r\n      \r\n      fields(includeDeprecated: true) {
  \r\n        name\r\n        description\r\n        args {\r\n          ...InputValue\r\n        }\r\n        typ
  e {\r\n          ...TypeRef\r\n        }\r\n        isDeprecated\r\n        deprecationReason\r\n      }\r\n
   inputFields {\r\n        ...InputValue\r\n      }\r\n      interfaces {\r\n        ...TypeRef\r\n      }\r\n
    enumValues(includeDeprecated: true) {\r\n        name\r\n        description\r\n        isDeprecated\r\n
    deprecationReason\r\n      }\r\n      possibleTypes {\r\n        ...TypeRef\r\n      }\r\n    }\r\n\r\n    f
  ragment InputValue on __InputValue {\r\n      name\r\n      description\r\n      type { ...TypeRef }\r\n      de
  faultValue\r\n      \r\n      \r\n    }\r\n\r\n    fragment TypeRef on __Type {\r\n      kind\r\n      name\r\n
     ofType {\r\n        kind\r\n        name\r\n        ofType {\r\n          kind\r\n          name\r\n
   ofType {\r\n            kind\r\n            name\r\n            ofType {\r\n              kind\r\n
   name\r\n              ofType {\r\n                kind\r\n                name\r\n                ofType {\r\
  n                  kind\r\n                  name\r\n                  ofType {\r\n                    kind\r\n
                     name\r\n                    }\r\n                  }\r\n                }\r\n              }\r\n
     }\r\n          }\r\n        }\r\n      }\r\n    "
}
```

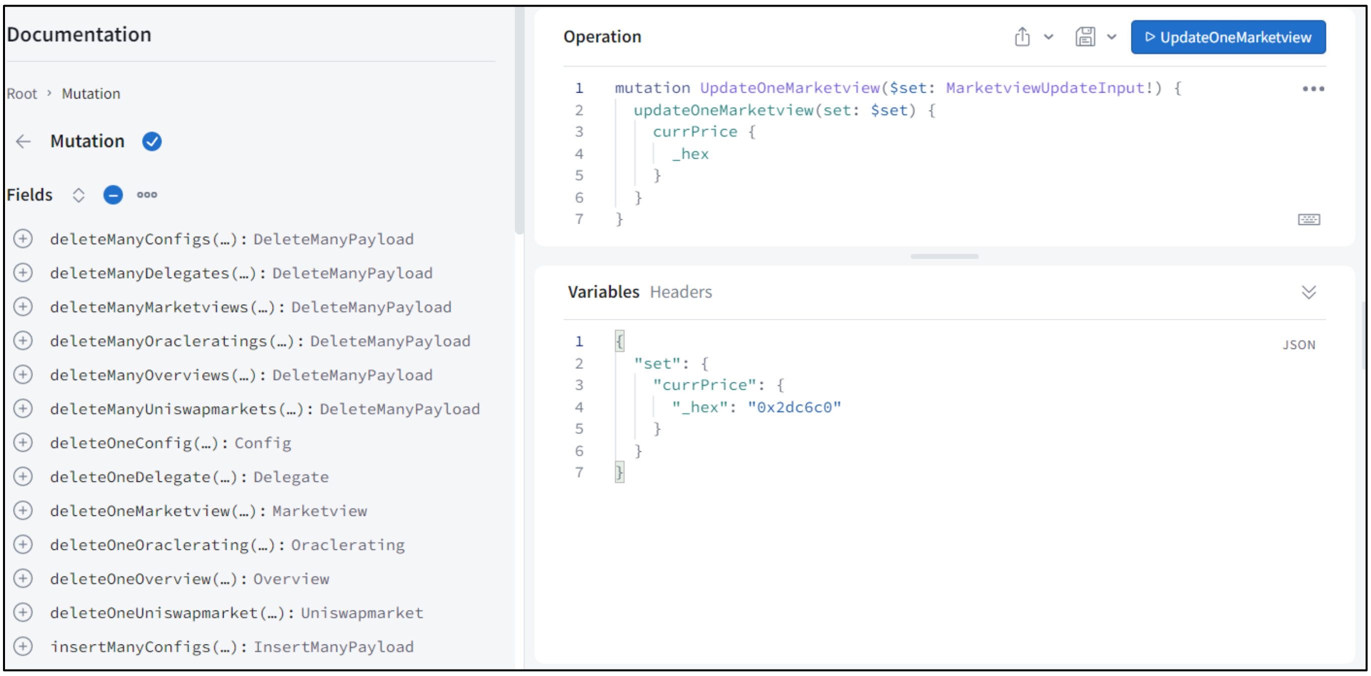
Figure 6 **- GraphQL introspection query**



```
HTTP/2 200 OK
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://app.euler.finance
Access-Control-Expose-Headers: X-Stitch-Location, X-Baas-Location, Location
Content-Type: application/json
Vary: Origin
X-Frame-Options: DENY
Date: Wed, 08 Jun 2022 10:01:51 GMT
X-Envoy-Upstream-Service-Time: 249
Server: envoy

{
  "data":{
    "__schema":{
      "directives":[
        {
          "args":[
            {
              "defaultValue":null,
              "description":"Included when true.",
              "name":"if",
              "type":{
                "kind":"NON_NULL",
                "name":null,
                "ofType":{
                  "kind":"SCALAR",
                  "name":"Boolean",
                  "ofType":null
                }
              }
            }
          ],
          "description":
          "Directs the executor to include this field or fragment only when the `if` argument is true.",
          "locations":[
            "FIELD",
            "FRAGMENT_SPREAD",
            "INLINE_FRAGMENT"
          ],
          "name":"include"
        },
        {
          "args":[
            {
              "defaultValue":null,
              "description":"Skipped when true.",
              "name":"if",
              "type":{
                "kind":"NON_NULL",
                "name":null,
                "ofType":{
                  "kind":"SCALAR",
                  "name":"Boolean",
                  "ofType":null
                }
              }
```

Figure 7 - Response from the server providing detailed information on the API schema



deleteManyConfigs(…): DeleteManyPayload
deleteManyDelegates(…): DeleteManyPayload
deleteManyMarketviews(…): DeleteManyPayload
deleteManyOracleratings(…): DeleteManyPayload
deleteManyOverviews(…): DeleteManyPayload
deleteManyUniswapmarkets(…): DeleteManyPayload
deleteOneConfig(…): Config
deleteOneDelegate(…): Delegate
deleteOneMarketview(…): Marketview
deleteOneOraclerating(…): Oraclerating
deleteOneOverview(…): Overview
deleteOneUniswapmarket(…): Uniswapmarket
insertManyConfigs(…): InsertManyPayload

```
mutation UpdateOneMarketview($set: MarketviewUpdateInput!) {
  updateOneMarketview(set: $set) {
    currPrice {
      _hex
    }
  }
}
```

```
{
  "set": {
    "currPrice": {
      "_hex": "0x2dc6c0"
    }
  }
}
```

Figure 8 - GraphQL schema imported into Apollo Sandbox, allowing for analysis and creation of requests

Recommendations

Introspection should be disabled if possible. If required or desired to facilitate agile development, very thorough assessment of the API's functionality should be conducted to ensure that it is not vulnerable to exploitation.

Affected	app.euler.finance
References & CVSSv3 Metrics	What is Introspection: https://graphql.org/learn/introspection/ OWASP: GraphQL cheat sheet: https://cheatsheetseries.owasp.org/cheatsheets/GraphQL_Cheat_Sheet.html Common Weakness Enumeration (CWEs) IDs - 200 Root Cause: Implementation Base Metrics: AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N (3.7) Temporal Metrics: E:F/RL:W/RC:C (3.5) Environmental Metrics: CR:M/IR:M/AR:M (3.5)

6. Appendix

Risk Rating

This report scores vulnerabilities using CVSS v3, the latest industry standard. It combines this with the simplicity of colour coding. This enables access to this report by all levels of management.

Issue Alerts

"Issue Alerts" allow the reader to quickly and easily identify issues and their associated severities. In each section, the reader can read a detailed description of the issue, how it was identified, and the associated mitigation that has been recommended.

Issues are rated either critical, high, medium or low risk depending on their CVSS v3 score. Informational recommendations may also be made that do not relate to a specific vulnerability or associated risk. Each risk group is assigned its own colour as shown below:

Informational	**Low Risk**	**Medium Risk**	**High Risk**	**Critical Risk**

CVSS v3 Explanation

CVSS (currently version 3) is the Common Vulnerability Scoring System. This is a vendor independent way of scoring vulnerabilities in a more granular way than just being assigned as a critical, high, medium, or low risk.

This system takes a variety of factors (known as metrics) into account such as the level of complexity required to reach the affected system, whether or not exploit code exists, the impact successful exploitation of the issue would have on the business and the type of area of concern (availability, confidentiality and integrity).

By applying these factors to each unique vulnerability, a score from 0 to 10 is calculated and assigned.
Pen Test Partners assigns critical, high, medium or low to each vulnerability based on the following criteria:

Critical:	Any issue with a CVSS score of 9.0 or higher
High:	Any issue with a CVSS score of 7.0 or higher but lower than 9.0
Medium:	Any issue with a CVSS score of 4.0 or higher but lower than 7.0
Low:	Any issue with a CVSS score lower than 4.0

This assures that each vulnerability has been tailored to the client, as each vulnerability affects each client in different ways.

For example, an SQL injection issue affecting a public facing website would be an extremely high risk. That same issue on an internal host with adequate firewall configurations could be classed as a medium risk. A high-risk issue on a low impact server may carry a lower CVSS score than a medium risk issue on a critical server.

For more information on CVSS please refer to the First.org website link: http://www.first.org/cvss/.