IOActive
Research-fueled Security Services

Mixin Virtual Machine Bridge Security Assessment

Mixin, Ltd.

IOActive, Inc.
1426 Elliott Ave W
Seattle, WA 98119

Toll free: (866) 760-0222
Office: (206) 784-4313
Fax: (206) 784-4367

Document Management

Document Information

Client Name	Mixin, Ltd.
Project Name	Mixin Virtual Machine Bridge Security Assessment
Project Start Date	2022-09-05
Project End Date	2022-09-09

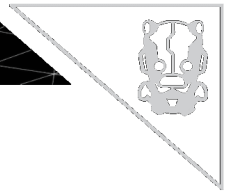
Document Revision Information

Date	Version	Author	Revision Details
2022-09-09	1.0	IOActive	Initial Version
2022-09-12	1.1	IJR	Peer Review
2022-09-12	1.2	IOActive	Edits
2022-09-12	1.3	Roy Albert	Director Review
2022-09-14	1.4	IOActive	Minor Edits
2022-10-12	1.5	IOActive	Edits
2022-10-19	1.6	IOActive	Edits and Tech Pubs QA

Project Contacts

IOActive, Inc.

Name	Title	Contact Information
Ehab Hussein	Principal Security Consultant	ehab.hussein@ioactive.com
Mohamed Samy	Security Consultant	mohamed.samy@ioactive.com
Roy Albert	Senior Director of Services	roy.albert@ioactive.com
Pratul Anand	Engagement Manager	pratul.anand@ioactive.com
Claudio Morusca	Senior Engagement Manager	claudio.morusca@ioactive.com



Contents

Executive Summary

Mixin, Ltd. (Mixin) engaged IOActive, Inc. (IOActive) to perform a code review and assess the security posture of the bridge contract and registry contract assets of the Mixin Virtual Machine (MVM) project.

IOActive reviewed the open-source smart contracts and conducted open-source intelligence (OSINT) gathering to obtain additional relevant artifacts where not immediately made available by Mixin.

Project Description

Two IOActive security consultants conducted a code review from the 5th to the 9th of September 2022. IOActive's intent for this engagement was to apply a white-box assessment methodology; however, Mixin was unable to deliver access to the backend documentation within the project's pre-requisite window, resulting in a change of methodology from white-box to gray-box.

Based on methodology employed and the observations made during this assessment, IOActive rates the overall maturity and security posture of the in-scope codebase to be excellent.

Key Takeaways

The codebase was tested extensively and was found to be sound.

Due to the lack of a Testnet blockchain, dynamic analysis was not performed. The primary focus of dynamic testing would be to ensure end-to-end integrity of transactions moving through the bridge and into the Ethereum network along with attacking associated secure wallets, which was out of scope for this engagement. IOActive recommends that a follow-up engagement be conducted to perform such testing and further recommends that it be done on a Testnet to ensure customers are not impacted by testing activities.

Analysis of Findings

Figure 1 shows the distribution of findings by risk rating.

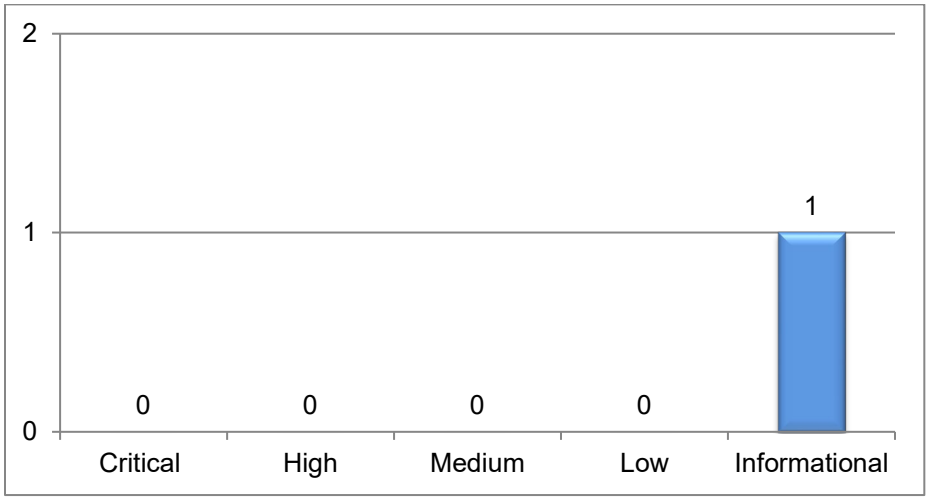


Figure 1. Distribution of Findings

The assessment identified one informational finding. `trusted-group\mvm\quorum\registry` included several outdated third-party libraries that are vulnerable to remote attacks. The general advice for such a finding is to update the libraries to their latest versions after performing regression and acceptance testing.

Table 1. Summary of findings

Finding ID	Title	Total Risk	Effort to Fix
#MVM-01	Outdated Third-party Libraries	Informational	Low

Important The effort to address vulnerabilities is an estimate reflecting the assessment team's experience; actual remediation effort may vary based on numerous factors including skill sets, process efficiency, and available resources.

Technical Summary

Scope

The following GitHub repositories were in scope:

- https://github.com/MixinNetwork/trusted-group/tree/master/mvm/quorum/bridge/contracts
- https://github.com/MixinNetwork/trusted-group/tree/master/mvm/quorum/registry/contracts

Figure 2 and Figure 3 show the in-scope repositories.

Figure 2. Bridge Contracts

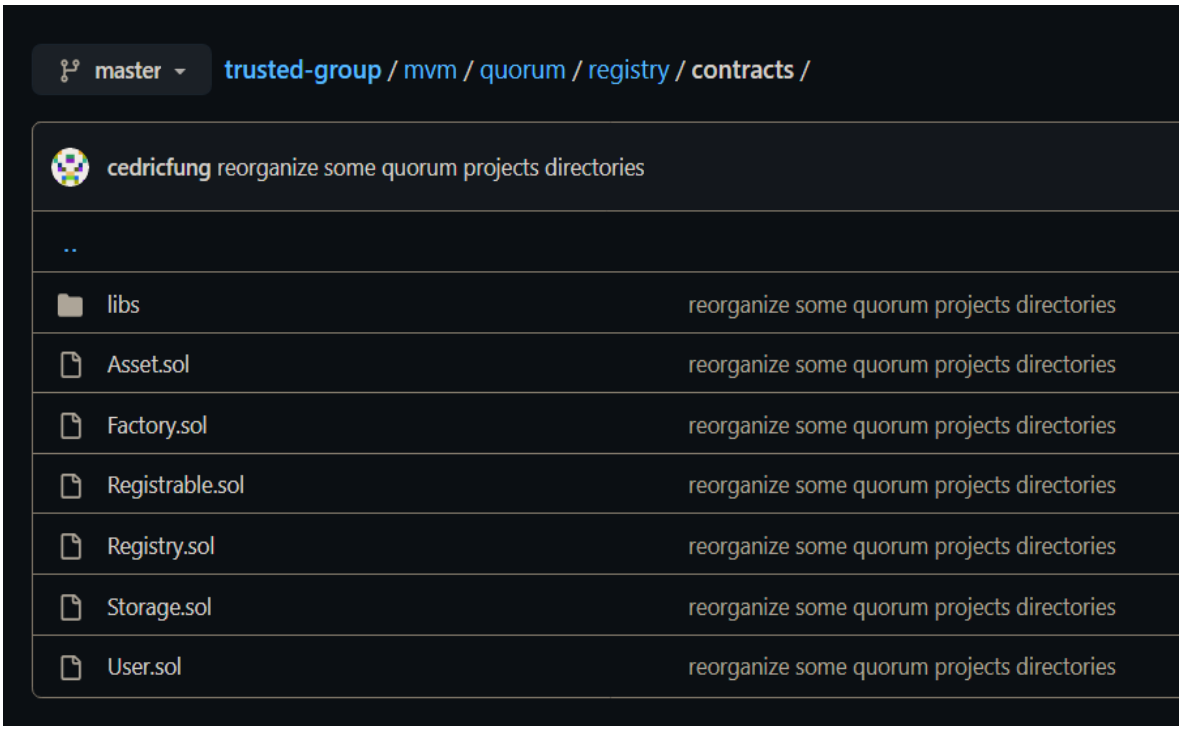
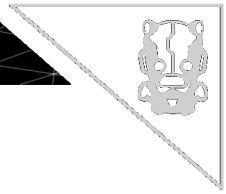


Figure 3. Registry Contracts



Project Approach

Due to a lack of backend documentation, the consultants resorted to an OSINT gathering approach and identified the following resources:

- https://bridge.mvm.dev
- https://scan.mvm.dev
- https://geth.mvm.dev
- https://developers.mixin.one

Initially, the consultants performed static analysis of the Solidity source code using VS Code with multiple plugins and EVM-compliable analyzers for manual review.

The source code was mostly dependent on off-the-shelf and battle-tested Solidity libraries and pre-compiled contracts inside the EVM itself; for example, Contract 8 for checking the elliptic curve pairing operations required to perform zkSNARK verification within the block gas limit (EIP 197).

Most of the core functionality was written in Gas-optimized and purpose-built EVM assembly code. While assembly code makes static code review much more difficult, the Gas-optimized code was deemed to be more efficient than that generated by standard Solidity compliers.

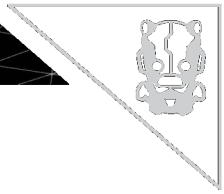
Automated testing of the assembly code generated many false positives, which the consultants reviewed. The consultants then wrote multiple test cases to confirm the necessity of writing such portions in EVM assembly code.

Next, the consultants performed dynamic analysis utilizing multiple toolchains and EVM environments. For the first stage, the consultants used Hardhat EVM, which simulates the process of deploying the smart contracts and issuing transactions in an isolated and controlled developer-style environment as well as conducting testing and fuzzing tasks. During the second stage, the consultants performed dynamic code analysis using the publicly deployed smart contracts on Mainnet through Mixin Geth Node.[1]

The consultants also performed best-effort reverse engineering of the public MVM blockchain in order to gain a better understanding of the lifecycle of bridge and registry smart contracts within Mixin's ecosystem.

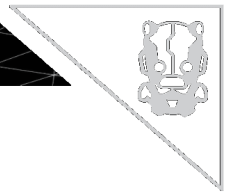
The consultants identified multiple critical code paths that handle cryptocurrency asset transfers and cryptographic key custody; however, the consultants could not audit or verify these code paths due to a lack of documentation for the off-chain communication channels.

[1] https://geth.mvm.dev:443



The lack of access to a Testnet blockchain prevented the consultants from generating or performing any blockchain ‘write’ transactions, which could aid in testing and fuzzing the deployed contracts within the Mixin ecosystem.

Given such constraints on dynamic analysis, it was deemed impossible to issue cryptographically signed transactions using the input data shown in Appendix B.



Detailed Findings

#MVM-01 - Outdated Third-party Libraries

Host(s) / File(s)	trusted-group\mvm\quorum\registry*
Category	Software Vulnerabilities
Testing Method	White Box
Tools Used	NPM Audit
Likelihood	Informational (1)
Impact	Informational (1)
Total Risk Rating	Informational (1)
Effort to Fix	Low

Threat and Impact

The consultants ran an NPM audit against the codebase, which identified numerous outdated third-party libraries. Using out-of-date software increases the risk of using code that has known vulnerabilities that have not been patched. No public exploit code was found; however, unpublished zero-day exploits might exist.

```
~#@? npm audit
# npm audit report

ansi-regex  3.0.0
Severity: high
Inefficient Regular Expression Complexity in chalk/ansi-
regex - https://github.com/advisories/GHSA-93q8-gq69-wqmw
fix available via `npm audit fix`
node_modules/nyc/node_modules/string-
width/node_modules/ansi-regex
node_modules/nyc/node_modules/yargs/node_modules/ansi-regex

async  2.0.0 - 2.6.3
Severity: high
Prototype Pollution in async -
https://github.com/advisories/GHSA-fwr7-v2mv-hh25
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/async

cross-fetch  <=2.2.5 || 3.0.0 - 3.1.4 || >=3.2.0-alpha.0
Severity: high
```

```
Incorrect Authorization in cross-fetch -
https://github.com/advisories/GHSA-7gc6-qh9x-w6h8
Depends on vulnerable versions of node-fetch
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/cross-fetch

elliptic  <6.5.4
Severity: moderate
Use of a Broken or Risky Cryptographic Algorithm -
https://github.com/advisories/GHSA-r9p9-mrjm-926w
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/elliptic
  @ethersproject/signing-key  <=5.0.9
  Depends on vulnerable versions of elliptic
  node_modules/ganache-
core/node_modules/@ethersproject/signing-key

got  <11.8.5
Severity: moderate
Got allows a redirect to a UNIX socket -
https://github.com/advisories/GHSA-pfrx-2q88-qq97
No fix available
node_modules/ganache-core/node_modules/got
node_modules/ganache-core/node_modules/swarm-
js/node_modules/got
  swarm-js  0.1.1 - 0.1.17 || 0.1.35 - 0.1.40
  Depends on vulnerable versions of got
  node_modules/ganache-core/node_modules/swarm-js
  web3-bzz  <=1.7.4
  Depends on vulnerable versions of got
  Depends on vulnerable versions of underscore
  node_modules/ganache-core/node_modules/web3-bzz
    web3  <=1.7.4 || 1.8.0-rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-bzz
    Depends on vulnerable versions of web3-shh
    Depends on vulnerable versions of web3-utils
    node_modules/ganache-core/node_modules/web3
      ganache-core  <=2.1.0-beta.7 || >=2.1.1
      Depends on vulnerable versions of lodash
      Depends on vulnerable versions of web3
      Depends on vulnerable versions of web3-provider-engine
      node_modules/ganache-core
        @ethereum-waffle/provider  <=4.0.1-dev.37f589d ||
4.0.2-dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-dev.0c13fb9 -
4.0.3-dev.e7e18f6
        Depends on vulnerable versions of @ethereum-
waffle/ens
        Depends on vulnerable versions of ganache-core
        node_modules/@ethereum-waffle/provider
          @ethereum-waffle/chai  2.5.0 - 4.0.0-dev.e3fa452
          Depends on vulnerable versions of @ethereum-
waffle/provider
          node_modules/@ethereum-waffle/chai
            ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
```

```
            Depends on vulnerable versions of @ethereum-
waffle/chai
            Depends on vulnerable versions of @ethereum-
waffle/provider
            node_modules/ethereum-waffle
              @nomiclabs/hardhat-waffle  *
              Depends on vulnerable versions of ethereum-
waffle
              node_modules/@nomiclabs/hardhat-waffle

handlebars  <=4.7.6
Severity: critical
Prototype Pollution in handlebars -
https://github.com/advisories/GHSA-q42p-pg8m-cqh6
Prototype Pollution in handlebars -
https://github.com/advisories/GHSA-w457-6q6x-cgp9
Arbitrary Code Execution in Handlebars -
https://github.com/advisories/GHSA-3cqr-58rm-57f8
Regular Expression Denial of Service in Handlebars -
https://github.com/advisories/GHSA-62gr-4qp9-h98f
Remote code execution in handlebars when compiling templates
- https://github.com/advisories/GHSA-f2jv-r9rf-7988
Arbitrary Code Execution in handlebars -
https://github.com/advisories/GHSA-2cf5-4w76-r9qv
Depends on vulnerable versions of optimist
fix available via `npm audit fix`
node_modules/nyc/node_modules/handlebars

hosted-git-info  <2.8.9
Severity: moderate
Regular Expression Denial of Service in hosted-git-info -
https://github.com/advisories/GHSA-43f8-2h32-f4cj
fix available via `npm audit fix`
node_modules/nyc/node_modules/hosted-git-info

json-schema  <0.4.0
Severity: critical
json-schema is vulnerable to Prototype Pollution -
https://github.com/advisories/GHSA-896r-f27r-55mw
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/json-schema
  jsprim  0.3.0 - 1.4.1 || 2.0.0 - 2.0.1
  Depends on vulnerable versions of json-schema
  node_modules/ganache-core/node_modules/jsprim

kind-of  6.0.0 - 6.0.2
Severity: high
Validation Bypass in kind-of -
https://github.com/advisories/GHSA-6c8f-qphg-qjgp
fix available via `npm audit fix`
node_modules/nyc/node_modules/base/node_modules/kind-of
node_modules/nyc/node_modules/define-
property/node_modules/kind-of
node_modules/nyc/node_modules/extglob/node_modules/kind-of
node_modules/nyc/node_modules/micromatch/node_modules/kind-
of
```

```
node_modules/nyc/node_modules/nanomatch/node_modules/kind-of
node_modules/nyc/node_modules/snapdragon-
node/node_modules/kind-of
node_modules/nyc/node_modules/test-
exclude/node_modules/kind-of
node_modules/nyc/node_modules/use/node_modules/kind-of

lodash  <=4.17.20
Severity: critical
Prototype Pollution in lodash -
https://github.com/advisories/GHSA-jf85-cpcp-j695
Regular Expression Denial of Service (ReDoS) in lodash -
https://github.com/advisories/GHSA-x5rq-j2xg-h7qm
Prototype Pollution in lodash -
https://github.com/advisories/GHSA-p6mc-m468-83gw
Command Injection in lodash -
https://github.com/advisories/GHSA-35jh-r3h4-6jhm
No fix available
node_modules/ganache-core/node_modules/lodash
node_modules/nyc/node_modules/lodash
  ganache-core  <=2.1.0-beta.7 || >=2.1.1
  Depends on vulnerable versions of lodash
  Depends on vulnerable versions of web3
  Depends on vulnerable versions of web3-provider-engine
  node_modules/ganache-core
    @ethereum-waffle/provider  <=4.0.1-dev.37f589d || 4.0.2-
dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-dev.0c13fb9 -
4.0.3-dev.e7e18f6
    Depends on vulnerable versions of @ethereum-waffle/ens
    Depends on vulnerable versions of ganache-core
    node_modules/@ethereum-waffle/provider
      @ethereum-waffle/chai  2.5.0 - 4.0.0-dev.e3fa452
      Depends on vulnerable versions of @ethereum-
waffle/provider
      node_modules/@ethereum-waffle/chai
        ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
        Depends on vulnerable versions of @ethereum-
waffle/chai
        Depends on vulnerable versions of @ethereum-
waffle/provider
        node_modules/ethereum-waffle
          @nomiclabs/hardhat-waffle  *
          Depends on vulnerable versions of ethereum-waffle
          node_modules/@nomiclabs/hardhat-waffle

mem  <4.0.0
Severity: moderate
Denial of Service in mem -
https://github.com/advisories/GHSA-4xcv-9jjx-gfj3
No fix available
node_modules/nyc/node_modules/mem
  os-locale  2.0.0 - 3.0.0
  Depends on vulnerable versions of mem
  node_modules/nyc/node_modules/os-locale
```

```
    yargs  4.0.0-alpha1 - 7.0.0-alpha.3 || 7.1.1 || 8.0.0-
candidate.0 - 12.0.5
    Depends on vulnerable versions of os-locale
    Depends on vulnerable versions of yargs-parser
    Depends on vulnerable versions of yargs-parser
    node_modules/@ensdomains/ens/node_modules/yargs
    node_modules/nyc/node_modules/yargs
      nyc  6.2.0-alpha - 13.3.0
      Depends on vulnerable versions of mkdirp
      Depends on vulnerable versions of yargs
      Depends on vulnerable versions of yargs-parser
      node_modules/nyc
        mcl-wasm  0.1.0 - 0.4.5
        Depends on vulnerable versions of nyc
        node_modules/mcl-wasm
      solc  0.3.6 - 0.4.26
      Depends on vulnerable versions of yargs
      node_modules/@ensdomains/ens/node_modules/solc
        @ensdomains/ens  *
        Depends on vulnerable versions of solc
        node_modules/@ensdomains/ens
          @ethereum-waffle/ens  <=4.0.1-dev.e7e18f6
          Depends on vulnerable versions of @ensdomains/ens
          node_modules/@ethereum-waffle/ens
            @ethereum-waffle/provider  <=4.0.1-dev.37f589d
|| 4.0.2-dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-
dev.0c13fb9 - 4.0.3-dev.e7e18f6
            Depends on vulnerable versions of @ethereum-
waffle/ens
            Depends on vulnerable versions of ganache-core
            node_modules/@ethereum-waffle/provider
              @ethereum-waffle/chai  2.5.0 - 4.0.0-
dev.e3fa452
              Depends on vulnerable versions of @ethereum-
waffle/provider
              node_modules/@ethereum-waffle/chai
                ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
                Depends on vulnerable versions of @ethereum-
waffle/chai
                Depends on vulnerable versions of @ethereum-
waffle/provider
                node_modules/ethereum-waffle
                  @nomiclabs/hardhat-waffle  *
                  Depends on vulnerable versions of
ethereum-waffle
                  node_modules/@nomiclabs/hardhat-waffle

minimist  <=1.2.5
Severity: critical
Prototype Pollution in minimist -
https://github.com/advisories/GHSA-xvch-5gv4-984h
Prototype Pollution in minimist -
https://github.com/advisories/GHSA-vh95-rmgr-6w4m
fix available via `npm audit fix --force`
Will install mcl-wasm@1.0.3, which is a breaking change
```

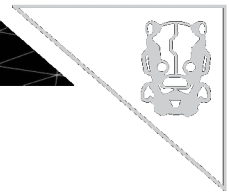
```
node_modules/ganache-core/node_modules/minimist
node_modules/nyc/node_modules/minimist
  mkdirp  0.4.1 - 0.5.1
  Depends on vulnerable versions of minimist
  node_modules/nyc/node_modules/mkdirp
    nyc  6.2.0-alpha - 13.3.0
    Depends on vulnerable versions of mkdirp
    Depends on vulnerable versions of yargs
    Depends on vulnerable versions of yargs-parser
    node_modules/nyc
      mcl-wasm  0.1.0 - 0.4.5
      Depends on vulnerable versions of nyc
      node_modules/mcl-wasm
  optimist  >=0.6.0
  Depends on vulnerable versions of minimist
  node_modules/nyc/node_modules/optimist
    handlebars  <=4.7.6
    Depends on vulnerable versions of optimist
    node_modules/nyc/node_modules/handlebars

mixin-deep  <1.3.2
Severity: critical
Prototype Pollution in mixin-deep -
https://github.com/advisories/GHSA-fhjf-83wg-r2j9
fix available via `npm audit fix`
node_modules/nyc/node_modules/mixin-deep

node-fetch  <=2.6.6
Severity: high
node-fetch is vulnerable to Exposure of Sensitive
Information to an Unauthorized Actor -
https://github.com/advisories/GHSA-r683-j2x4-v87g
The `size` option isn't honored after following a redirect
in node-fetch - https://github.com/advisories/GHSA-w7rc-
rwvf-8q5r
No fix available
node_modules/ganache-core/node_modules/fetch-
ponyfill/node_modules/node-fetch
node_modules/ganache-core/node_modules/node-fetch
  cross-fetch  <=2.2.5 || 3.0.0 - 3.1.4 || >=3.2.0-alpha.0
  Depends on vulnerable versions of node-fetch
  node_modules/ganache-core/node_modules/cross-fetch
  fetch-ponyfill  1.0.0 - 6.0.2
  Depends on vulnerable versions of node-fetch
  node_modules/ganache-core/node_modules/fetch-ponyfill
    eth-json-rpc-middleware  1.1.0 - 5.0.2
    Depends on vulnerable versions of fetch-ponyfill
    node_modules/ganache-core/node_modules/eth-json-rpc-
middleware
      eth-json-rpc-infura  <=5.0.0
      Depends on vulnerable versions of eth-json-rpc-
middleware
      node_modules/ganache-core/node_modules/eth-json-rpc-
infura
        web3-provider-engine  14.0.0 - 15.0.12
```

```
        Depends on vulnerable versions of eth-json-rpc-
infura
        node_modules/ganache-core/node_modules/web3-
provider-engine
          ganache-core  <=2.1.0-beta.7 || >=2.1.1
          Depends on vulnerable versions of lodash
          Depends on vulnerable versions of web3
          Depends on vulnerable versions of web3-provider-
engine
          node_modules/ganache-core
            @ethereum-waffle/provider  <=4.0.1-dev.37f589d
|| 4.0.2-dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-
dev.0c13fb9 - 4.0.3-dev.e7e18f6
            Depends on vulnerable versions of @ethereum-
waffle/ens
            Depends on vulnerable versions of ganache-core
            node_modules/@ethereum-waffle/provider
              @ethereum-waffle/chai  2.5.0 - 4.0.0-
dev.e3fa452
              Depends on vulnerable versions of @ethereum-
waffle/provider
              node_modules/@ethereum-waffle/chai
                ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
                Depends on vulnerable versions of @ethereum-
waffle/chai
                Depends on vulnerable versions of @ethereum-
waffle/provider
                node_modules/ethereum-waffle
                  @nomiclabs/hardhat-waffle  *
                  Depends on vulnerable versions of
ethereum-waffle
                  node_modules/@nomiclabs/hardhat-waffle

normalize-url  4.3.0 - 4.5.0
Severity: high
ReDoS in normalize-url - https://github.com/advisories/GHSA-
px4h-xg32-q955
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/normalize-url

path-parse  <1.0.7
Severity: moderate
Regular Expression Denial of Service in path-parse -
https://github.com/advisories/GHSA-hj48-42vr-x3v9
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/path-parse
node_modules/nyc/node_modules/path-parse

set-value  <2.0.1
Severity: high
Prototype Pollution in set-value -
https://github.com/advisories/GHSA-4jqc-8m5r-9rpr
fix available via `npm audit fix`
node_modules/nyc/node_modules/set-value
```

```
node_modules/nyc/node_modules/union-value/node_modules/set-
value
  union-value  <=1.0.0 || 2.0.0
  Depends on vulnerable versions of set-value
  node_modules/nyc/node_modules/union-value

simple-get  <2.8.2
Severity: high
Exposure of Sensitive Information in simple-get -
https://github.com/advisories/GHSA-wpg7-2c88-r8xv
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/simple-get

tar  <=4.4.17
Severity: high
Arbitrary File Creation/Overwrite on Windows via
insufficient relative path sanitization -
https://github.com/advisories/GHSA-5955-9wpr-37jh
Arbitrary File Creation/Overwrite via insufficient symlink
protection due to directory cache poisoning using symbolic
links - https://github.com/advisories/GHSA-9r2w-394v-53qc
Arbitrary File Creation/Overwrite due to insufficient
absolute path sanitization -
https://github.com/advisories/GHSA-3jfq-g458-7qm9
Arbitrary File Creation/Overwrite via insufficient symlink
protection due to directory cache poisoning -
https://github.com/advisories/GHSA-r628-mhmh-qjhw
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/tar

underscore  1.3.2 - 1.12.0
Severity: high
Arbitrary Code Execution in underscore -
https://github.com/advisories/GHSA-cf4h-3jhx-xvhq
No fix available
node_modules/ganache-core/node_modules/underscore
  web3-bzz  <=1.7.4
  Depends on vulnerable versions of got
  Depends on vulnerable versions of underscore
  node_modules/ganache-core/node_modules/web3-bzz
    web3  <=1.7.4 || 1.8.0-rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-bzz
    Depends on vulnerable versions of web3-shh
    Depends on vulnerable versions of web3-utils
    node_modules/ganache-core/node_modules/web3
      ganache-core  <=2.1.0-beta.7 || >=2.1.1
      Depends on vulnerable versions of lodash
      Depends on vulnerable versions of web3
      Depends on vulnerable versions of web3-provider-engine
      node_modules/ganache-core
        @ethereum-waffle/provider  <=4.0.1-dev.37f589d ||
4.0.2-dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-dev.0c13fb9 -
4.0.3-dev.e7e18f6
        Depends on vulnerable versions of @ethereum-
waffle/ens
        Depends on vulnerable versions of ganache-core
```

```
        node_modules/@ethereum-waffle/provider
          @ethereum-waffle/chai  2.5.0 - 4.0.0-dev.e3fa452
          Depends on vulnerable versions of @ethereum-
waffle/provider
          node_modules/@ethereum-waffle/chai
            ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
            Depends on vulnerable versions of @ethereum-
waffle/chai
            Depends on vulnerable versions of @ethereum-
waffle/provider
            node_modules/ethereum-waffle
              @nomiclabs/hardhat-waffle  *
              Depends on vulnerable versions of ethereum-
waffle
              node_modules/@nomiclabs/hardhat-waffle
  web3-core-helpers  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-rc.4
  Depends on vulnerable versions of underscore
  Depends on vulnerable versions of web3-utils
  node_modules/ganache-core/node_modules/web3-core-helpers
    web3-core-subscriptions  <=1.3.6-rc.2 || 1.8.0-rc.0 -
3.0.0-rc.4
    Depends on vulnerable versions of underscore
    Depends on vulnerable versions of web3-core-helpers
    node_modules/ganache-core/node_modules/web3-core-
subscriptions
      web3-core-method  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-
rc.4
      Depends on vulnerable versions of underscore
      Depends on vulnerable versions of web3-core-
subscriptions
      node_modules/ganache-core/node_modules/web3-core-
method
      web3-eth-contract  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-
rc.4
      Depends on vulnerable versions of underscore
      Depends on vulnerable versions of web3-core-
subscriptions
      node_modules/ganache-core/node_modules/web3-eth-
contract
        web3-eth  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-rc.4
        Depends on vulnerable versions of underscore
        Depends on vulnerable versions of web3-eth-contract
        Depends on vulnerable versions of web3-eth-ens
        node_modules/ganache-core/node_modules/web3-eth
        web3-eth-ens  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-
rc.4
        Depends on vulnerable versions of underscore
        Depends on vulnerable versions of web3-core
        Depends on vulnerable versions of web3-eth-contract
        node_modules/ganache-core/node_modules/web3-eth-ens
      web3-shh  <=1.3.5
      Depends on vulnerable versions of web3-core-
subscriptions
      Depends on vulnerable versions of web3-net
      node_modules/ganache-core/node_modules/web3-shh
```

```
    web3-providers-http  <=1.0.0 || 1.2.0 - 1.3.5 || 1.8.0-
rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-core-helpers
    node_modules/ganache-core/node_modules/web3-providers-
http
    web3-providers-ipc  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-
rc.5
    Depends on vulnerable versions of underscore
    Depends on vulnerable versions of web3-core-helpers
    node_modules/ganache-core/node_modules/web3-providers-
ipc
    web3-providers-ws  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-
rc.4
    Depends on vulnerable versions of underscore
    Depends on vulnerable versions of web3-core-helpers
    node_modules/ganache-core/node_modules/web3-providers-ws
  web3-core-requestmanager  <=1.3.5 || 1.8.0-rc.0 - 3.0.0-
rc.4
  Depends on vulnerable versions of underscore
  node_modules/ganache-core/node_modules/web3-core-
requestmanager
  web3-eth-abi  <=1.3.6-rc.2 || 1.8.0-rc.0 - 3.0.0-rc.4
  Depends on vulnerable versions of underscore
  Depends on vulnerable versions of web3-utils
  node_modules/ganache-core/node_modules/web3-eth-abi
  web3-eth-accounts  <=1.3.5 || 1.8.0-rc.0 - 3.0.0-rc.4
  Depends on vulnerable versions of underscore
  node_modules/ganache-core/node_modules/web3-eth-accounts
  web3-utils  1.0.0-beta.8 - 1.3.5 || 1.8.0-rc.0 - 3.0.0-
rc.4
  Depends on vulnerable versions of underscore
  node_modules/ganache-core/node_modules/web3-utils
    web3-core  <=1.3.5 || 1.8.0-rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-utils
    node_modules/ganache-core/node_modules/web3-core
    web3-eth-iban  <=1.3.5 || 1.8.0-rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-utils
    node_modules/ganache-core/node_modules/web3-eth-iban
    web3-eth-personal  <=1.3.5 || 1.8.0-rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-net
    Depends on vulnerable versions of web3-utils
    node_modules/ganache-core/node_modules/web3-eth-personal
    web3-net  1.2.0 - 1.3.5 || 1.8.0-rc.0 - 3.0.0-rc.4
    Depends on vulnerable versions of web3-utils
    node_modules/ganache-core/node_modules/web3-net

undici  <=5.8.1
Severity: moderate
undici before v5.8.0 vulnerable to CRLF injection in request
headers - https://github.com/advisories/GHSA-3cvr-822r-rqcc
`undici.request` vulnerable to SSRF using absolute URL on
`pathname` - https://github.com/advisories/GHSA-8qr4-xgw6-
wmr3
fix available via `npm audit fix`
node_modules/undici
```



```
web3  <=1.7.4 || 1.8.0-rc.0 - 3.0.0-rc.4
Severity: high
Insecure Credential Storage in web3 -
https://github.com/advisories/GHSA-27v7-qhfv-rqq8
Depends on vulnerable versions of web3-bzz
Depends on vulnerable versions of web3-shh
Depends on vulnerable versions of web3-utils
No fix available
node_modules/ganache-core/node_modules/web3
  ganache-core  <=2.1.0-beta.7 || >=2.1.1
  Depends on vulnerable versions of lodash
  Depends on vulnerable versions of web3
  Depends on vulnerable versions of web3-provider-engine
  node_modules/ganache-core
    @ethereum-waffle/provider  <=4.0.1-dev.37f589d || 4.0.2-
dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-dev.0c13fb9 -
4.0.3-dev.e7e18f6
    Depends on vulnerable versions of @ethereum-waffle/ens
    Depends on vulnerable versions of ganache-core
    node_modules/@ethereum-waffle/provider
      @ethereum-waffle/chai  2.5.0 - 4.0.0-dev.e3fa452
      Depends on vulnerable versions of @ethereum-
waffle/provider
      node_modules/@ethereum-waffle/chai
        ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
        Depends on vulnerable versions of @ethereum-
waffle/chai
        Depends on vulnerable versions of @ethereum-
waffle/provider
        node_modules/ethereum-waffle
          @nomiclabs/hardhat-waffle  *
          Depends on vulnerable versions of ethereum-waffle
          node_modules/@nomiclabs/hardhat-waffle

ws  5.0.0 - 5.2.2
Severity: moderate
ReDoS in Sec-Websocket-Protocol header -
https://github.com/advisories/GHSA-6fc8-4gx4-v693
fix available via `npm audit fix`
node_modules/ganache-core/node_modules/web3-provider-
engine/node_modules/ws

y18n  <3.2.2
Severity: high
Prototype Pollution in y18n -
https://github.com/advisories/GHSA-c4w7-xm78-47vh
fix available via `npm audit fix`
node_modules/nyc/node_modules/y18n

yargs-parser  <=5.0.0 || 6.0.0 - 13.1.1
Severity: moderate
yargs-parser Vulnerable to Prototype Pollution -
https://github.com/advisories/GHSA-p9pc-299p-vxgp
yargs-parser Vulnerable to Prototype Pollution -
https://github.com/advisories/GHSA-p9pc-299p-vxgp
```

```
No fix available
node_modules/@ensdomains/ens/node_modules/yargs-parser
node_modules/nyc/node_modules/yargs-parser
node_modules/nyc/node_modules/yargs/node_modules/yargs-
parser
  nyc  6.2.0-alpha - 13.3.0
  Depends on vulnerable versions of mkdirp
  Depends on vulnerable versions of yargs
  Depends on vulnerable versions of yargs-parser
  node_modules/nyc
    mcl-wasm  0.1.0 - 0.4.5
    Depends on vulnerable versions of nyc
    node_modules/mcl-wasm
  yargs  4.0.0-alpha1 - 7.0.0-alpha.3 || 7.1.1 || 8.0.0-
candidate.0 - 12.0.5
  Depends on vulnerable versions of os-locale
  Depends on vulnerable versions of yargs-parser
  Depends on vulnerable versions of yargs-parser
  node_modules/@ensdomains/ens/node_modules/yargs
  node_modules/nyc/node_modules/yargs
    solc  0.3.6 - 0.4.26
    Depends on vulnerable versions of yargs
    node_modules/@ensdomains/ens/node_modules/solc
      @ensdomains/ens  *
      Depends on vulnerable versions of solc
      node_modules/@ensdomains/ens
        @ethereum-waffle/ens  <=4.0.1-dev.e7e18f6
        Depends on vulnerable versions of @ensdomains/ens
        node_modules/@ethereum-waffle/ens
          @ethereum-waffle/provider  <=4.0.1-dev.37f589d ||
4.0.2-dev.0a87072 - 4.0.2-dev.c513a49 || 4.0.3-dev.0c13fb9 -
4.0.3-dev.e7e18f6
          Depends on vulnerable versions of @ethereum-
waffle/ens
          Depends on vulnerable versions of ganache-core
          node_modules/@ethereum-waffle/provider
            @ethereum-waffle/chai  2.5.0 - 4.0.0-dev.e3fa452
            Depends on vulnerable versions of @ethereum-
waffle/provider
            node_modules/@ethereum-waffle/chai
              ethereum-waffle  2.3.0-istanbul.0 - 4.0.0-
dev.e3fa452
              Depends on vulnerable versions of @ethereum-
waffle/chai
              Depends on vulnerable versions of @ethereum-
waffle/provider
              node_modules/ethereum-waffle
                @nomiclabs/hardhat-waffle  *
                Depends on vulnerable versions of ethereum-
waffle
                node_modules/@nomiclabs/hardhat-waffle

66 vulnerabilities (19 moderate, 41 high, 6 critical)
```

Recommendations
Review the use of each library and associated vulnerable functionality. • If libraries are included but not used, remove associations. • If libraries are in use, implement a patch management system to maintain the libraries at their latest supported release. • If updated versions of the libraries are not available, additional mitigations should be implemented. For more information please see: https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

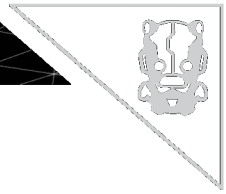
Appendix A: Overview of Risk Ratings and Finding Tables

Risk Ratings

To provide meaningful, quantitative analysis, IOActive uses an impact-versus-likelihood approach to scoring. For each finding, the assessment team assigns two ratings: one for impact and another for likelihood. Each rating corresponds to a numeric score ranging from 5 (critical) to 1 (informational). Table 2 explains each rating in terms of impact and likelihood.

Table 2. Rating and score as related to impact and likelihood

Rating (Score)	Impact	Likelihood
Critical (5)	Extreme impact to entire organization if exploited.	Vulnerability is almost certain to be exploited. Knowledge of the vulnerability and how to exploit it are in the public domain.
High (4)	Major impact to entire organization or single line of business if exploited.	Vulnerability is relatively easy to detect and exploit by an attacker with little skill.
Medium (3)	Noticeable impact to line of business if exploited.	A knowledgeable insider or expert attacker could exploit the vulnerability without much difficulty.
Low (2)	Minor damage if exploited or could be used in conjunction with other vulnerabilities to perform a more serious attack.	Exploiting the vulnerability would require considerable expertise and resources.
Informational (1)	Poor programming practice or poor design decision that may not represent an immediate risk on its own, but may have security implications if multiplied and/or combined with other vulnerabilities.	Vulnerability is not likely to be exploited on its own, but may be used to gain information for launching another attack.



IOActive calculates an aggregate risk score for each finding by multiplying its impact score by its likelihood score. For example, a finding with high likelihood and low impact would have an aggregate risk score of eight (8); that is, four (4) for high likelihood multiplied by two (2) for low impact. The aggregate risk score determines the finding's overall risk level, as shown in Table 3.

Table 3. Overall risk levels and corresponding aggregate scores

Overall Risk Level	Aggregate Risk Score (Impact multiplied by Likelihood)
Critical	20–25
High	12–19
Medium	6–11
Low	2–5
Informational	1

Finding Descriptors

IOActive's detailed findings tables provide a detailed description of what the consultants found, how those findings impact security, and what you should do to improve your security posture moving forward.

Threat and Impact. This field includes information about the vulnerability, including a specific and detailed description of the threat and what will happen if it is exploited. We include any applicable information for reproducing the finding, such as proof-of-concept code and the specific steps the consultants took to identify and exploit the finding. IOActive also provides screenshots, code blocks, static URLs, and any other relevant data that demonstrates the impact of the issue.

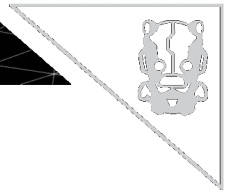
Recommendations. This field describes the actions required to prevent the vulnerability from being exploited. It may include specific step-by-step recommendations based on the assessment team’s experience or more general recommendations based upon standard industry solutions.

Finding Categories

IOActive categorizes findings using the vulnerability concepts described in Table 4.

Table 4. Vulnerability concepts

Concept	Description
Authentication	Confirming a user's identity or ensuring that a program can be trusted.
Access Controls	Methods used to authenticate the identity of a user, such as username and password combinations.
Broken Authentication and Session Management	Account credentials/session tokens are not protected properly, so attackers compromise passwords or keys to assume identities.
Configuration	How securely servers, devices, and software are chosen and implemented or deployed.
Cross-site Request Forgery	A browser is forced to send a pre-authenticated request to a vulnerable application, which then forces the browser to perform a hostile action that benefits the attacker.
Cross-site Scripting	When an application accepts user-supplied data and sends it to a web browser without first validating or encoding that content.
Cryptography and Insecure Storage	Applications rarely use mathematical data protections properly; attackers can conduct identity theft and credit card fraud.
Data Validation	Ensuring that a program operates on clean, correct, useful, and secure data.
Denial of Service	Anything that makes a computer resource unavailable to its intended users.
Failure to Restrict URL Access	When an application protects sensitive functionality by preventing its display as opposed to restricting access.
Information Leakage and Improper Error Handling	When an application exposes information about its configuration or internal function, or violates user privacy.
Insecure Communication	When an application fails to encrypt sensitive network traffic.
Insecure Direct Object Reference	When a reference to an internal implementation object (file, directory, database record, key, URL, etc.) is exposed.
Malicious File Execution	Code that is vulnerable to remote file inclusion allows attackers to include hostile code and data.
Session Management	The process of tracking a user's activity across sessions of interaction with a computer system.



Appendix B: Input Data for Fuzzing

The consultants created the following input data for fuzz testing but were unable to use it as Mixin did not provide access to a Testnet blockchain.

Storage

```
$ myth analyze -a 0x0915EaE769D68128EEd9711A0bc4097831BE57F3 --rpc
"geth.mvm.dev:443" --rpctls TRUE
params is: ['0x0915EaE769D68128EEd9711A0bc4097831BE57F3', 'latest']
params is: ['0x0915EaE769D68128EEd9711A0bc4097831BE57F3', 'latest']
params is: ['0x0915eae769d68128eed9711a0bc4097831be57f3', 'latest']
params is: ['0x0915eae769d68128eed9711a0bc4097831be57f3', 'latest']
params is: ['0x3c84b6c98fbeb813e05a7a7813f0442883450b1f', 'latest']
params is: ['0x3c84b6c98fbeb813e05a7a7813f0442883450b1f', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F', 'latest']
params is: ['0x181251d3a501961d4af2af46e33c71a5d808c25b', 'latest']
params is: ['0x181251d3a501961d4af2af46e33c71a5d808c25b', 'latest']
params is: ['0x181251D3A501961D4AF2AF46E33C71A5D808C25B', 'latest']
params is: ['0x181251D3A501961D4AF2AF46E33C71A5D808C25B', 'latest']
==== Integer Arithmetic Bugs ====
SWC ID: 101
Severity: High
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: pass(address,uint256)
PC address: 1467
Estimated Gas Usage: 5539 - 75865
The arithmetic operator can underflow.
It is possible to cause an integer overflow or underflow in the
arithmetic operation.
--------------------
Initial State:

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [CREATOR], function: bind(address), txdata:
0x81bac14f0000000000000000000000008040100402108004200202088040400402041020, value: 0x0
Caller: [CREATOR], function: pass(address,uint256), txdata:
0x0ed1db9f0000000000000000000000000000000000000000000000000000000000000000000001014020802010010101010101020102400401010180010120010104100101, value: 0x0

==== External Call To User-Supplied Address ====
SWC ID: 107
Severity: Low
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: pass(address,uint256)
PC address: 1554
```

```
Estimated Gas Usage: 5539 - 75865
A call to a user-supplied address is executed.
An external message call to an address specified by the caller is
executed. Note that the callee account might contain arbitrary code
and could re-enter any function within this contract. Reentering
the contract in an intermediate state may lead to unexpected
behaviour. Make sure that no state modifications are executed after
this call and/or reentrancy guards are in place.
--------------------
Initial State:

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [SOMEGUY], function: bind(address), txdata:
0x81bac14f0000000000000000000000008080808080808080808080808080808
0808004, value: 0x0
Caller: [SOMEGUY], function: pass(address,uint256), txdata:
0x0ed1db9f0000000000000000000000000deadbeefdeadbeefdeadbeefdeadbeefd
eadbeef0101010101010101010101010101010101010101010101010101010101010
0101, value: 0x0

==== Multiple Calls in a Single Transaction ====
SWC ID: 113
Severity: Low
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: pass(address,uint256)
PC address: 1554
Estimated Gas Usage: 5539 - 75865
Multiple calls are executed in the same transaction.
This call is executed following another call within the same
transaction. It is possible that the call never gets executed if a
prior call fails permanently. This might be caused intentionally by
a malicious callee. If possible, refactor the code such that each
transaction only executes one external call or make sure that all
callees can be trusted (i.e. they're part of your own codebase).
--------------------
Initial State:

Account: [ATTACKER], balance: 0x24, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [ATTACKER], function: bind(address), txdata:
0x81bac14f0000000000000000000000000000000000000000000000004000
0000000, value: 0x0
Caller: [ATTACKER], function: pass(address,uint256), txdata:
0x0ed1db9f0000000000000000000000000000000000000000000000000000000
0000000802008010140800101040108100101010108010101014010800120100
0204, value: 0x0
```

```
==== Multiple Calls in a Single Transaction ====
SWC ID: 113
Severity: Low
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: vault(address,uint256)
PC address: 1895
Estimated Gas Usage: 3841 - 73977
Multiple calls are executed in the same transaction.
This call is executed following another call within the same
transaction. It is possible that the call never gets executed if a
prior call fails permanently. This might be caused intentionally by
a malicious callee. If possible, refactor the code such that each
transaction only executes one external call or make sure that all
callees can be trusted (i.e. they're part of your own codebase).
--------------------
Initial State:

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [SOMEGUY], function: vault(address,uint256), txdata:
0x3fa16d990000000000000000000000181251d3a501961d4af2af46e33c71a5d
808c25b0000000000000000000000000000000000000000000000000000000000
0000, value: 0x0
```

Registry

```
$ myth analyze -a 0x3c84B6C98FBeB813e05a7A7813F0442883450B1F --rpc
"geth.mvm.dev:443" --rpctls TRUE
params is: ['0x3c84B6C98FBeB813e05a7A7813F0442883450B1F', 'latest']
params is: ['0x3c84B6C98FBeB813e05a7A7813F0442883450B1F', 'latest']
params is: ['0x3c84b6c98fbeb813e05a7a7813f0442883450b1f', 'latest']
params is: ['0x3c84b6c98fbeb813e05a7a7813f0442883450b1f', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F', '0x7',
'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F',
'0x0000000000000000000000000000000000000000000000000000000000000007
', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F',
'0x0000000000000000000000000000000000000000000000000000000000000007
', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F',
'0x1e225ab6b4b37f0eb3bb2d694ff09fda59ae3daeafb23c93db11ddfb7511a59e
', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F',
'0x1e225ab6b4b37f0eb3bb2d694ff09fda59ae3daeafb23c93db11ddfb7511a59e
', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F',
'0x1e225ab6b4b37f0eb3bb2d694ff09fda59ae3daeafb23c93db11ddfb7511a59e
', 'latest']
```

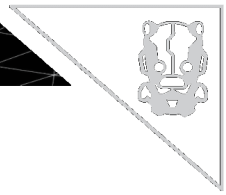
```
The analysis was completed successfully. No issues were detected.
```

Withdrawal

```
$ myth analyze -a 0xb27C8e0665D2Afa10F50A7CF4D2B9B6B461FD438 --rpc
"geth.mvm.dev:443" --rpctls TRUE
params is: ['0xb27C8e0665D2Afa10F50A7CF4D2B9B6B461FD438', 'latest']
params is: ['0xb27C8e0665D2Afa10F50A7CF4D2B9B6B461FD438', 'latest']
params is: ['0xb27c8e0665d2afa10f50a7cf4d2b9b6b461fd438', 'latest']
params is: ['0xb27c8e0665d2afa10f50a7cf4d2b9b6b461fd438', 'latest']
The analysis was completed successfully. No issues were detected.
```

Bridge

```
$ myth analyze -a 0x0915EaE769D68128EEd9711A0bc4097831BE57F3 --rpc
"geth.mvm.dev:443" --rpctls TRUE
params is: ['0x0915EaE769D68128EEd9711A0bc4097831BE57F3', 'latest']
params is: ['0x0915EaE769D68128EEd9711A0bc4097831BE57F3', 'latest']
params is: ['0x0915eae769d68128eed9711a0bc4097831be57f3', 'latest']
params is: ['0x0915eae769d68128eed9711a0bc4097831be57f3', 'latest']
params is: ['0x3c84b6c98fbeb813e05a7a7813f0442883450b1f', 'latest']
params is: ['0x3c84b6c98fbeb813e05a7a7813f0442883450b1f', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F', 'latest']
params is: ['0x3C84B6C98FBEB813E05A7A7813F0442883450B1F', 'latest']
params is: ['0x181251d3a501961d4af2af46e33c71a5d808c25b', 'latest']
params is: ['0x181251d3a501961d4af2af46e33c71a5d808c25b', 'latest']
params is: ['0x181251D3A501961D4AF2AF46E33C71A5D808C25B', 'latest']
params is: ['0x181251D3A501961D4AF2AF46E33C71A5D808C25B', 'latest']
==== Integer Arithmetic Bugs ====
SWC ID: 101
Severity: High
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: pass(address,uint256)
PC address: 1467
Estimated Gas Usage: 5539 - 75865
The arithmetic operator can underflow.
It is possible to cause an integer overflow or underflow in the
arithmetic operation.
--------------------
Initial State:

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [SOMEGUY], function: bind(address), txdata:
0x81bac14f0000000000000000000000008010102008808002040880200208802010
0040204, value: 0x0
Caller: [SOMEGUY], function: pass(address,uint256), txdata:
0x0ed1db9f0000000000000000000000000000000000000000000000000000000000
0000000101010101040101022020080101040810010101010201080101400201010
0408, value: 0x0
```



```
==== External Call To User-Supplied Address ====
SWC ID: 107
Severity: Low
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: pass(address,uint256)
PC address: 1554
Estimated Gas Usage: 5539 - 75865
A call to a user-supplied address is executed.
An external message call to an address specified by the caller is
executed. Note that the callee account might contain arbitrary code
and could re-enter any function within this contract. Reentering
the contract in an intermediate state may lead to unexpected
behaviour. Make sure that no state modifications are executed after
this call and/or reentrancy guards are in place.
--------------------
Initial State:

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [SOMEGUY], function: bind(address), txdata:
0x81bac14f0000000000000000000000008080808080808080808080808080808
0808004, value: 0x0
Caller: [SOMEGUY], function: pass(address,uint256), txdata:
0x0ed1db9f000000000000000000000000deadbeefdeadbeefdeadbeefdeadbeefd
eadbeef0101010101010101010101010101010101010101010101010101010101
0101, value: 0x0

==== Multiple Calls in a Single Transaction ====
SWC ID: 113
Severity: Low
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: pass(address,uint256)
PC address: 1554
Estimated Gas Usage: 5539 - 75865
Multiple calls are executed in the same transaction.
This call is executed following another call within the same
transaction. It is possible that the call never gets executed if a
prior call fails permanently. This might be caused intentionally by
a malicious callee. If possible, refactor the code such that each
transaction only executes one external call or make sure that all
callees can be trusted (i.e. they're part of your own codebase).
--------------------
Initial State:

Account: [ATTACKER], balance: 0x20, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:
```

```
Caller: [ATTACKER], function: bind(address), txdata:
0x81bac14f000000000000000000000000010101100101010820028020800101080
1010120, value: 0x0
Caller: [ATTACKER], function: pass(address,uint256), txdata:
0x0ed1db9f000000000000000000000000000000000000000000000000000000000
000000202040010101014001800180201001021004010180100401010101040180
0101, value: 0x0

==== Multiple Calls in a Single Transaction ====
SWC ID: 113
Severity: Low
Contract: 0x0915EaE769D68128EEd9711A0bc4097831BE57F3
Function name: vault(address,uint256)
PC address: 1895
Estimated Gas Usage: 3841 - 73977
Multiple calls are executed in the same transaction.
This call is executed following another call within the same
transaction. It is possible that the call never gets executed if a
prior call fails permanently. This might be caused intentionally by
a malicious callee. If possible, refactor the code such that each
transaction only executes one external call or make sure that all
callees can be trusted (i.e. they're part of your own codebase).
--------------------
Initial State:

Account: [ATTACKER], balance: 0x0, nonce:0, storage:{}
Account: [SOMEGUY], balance: 0x0, nonce:0, storage:{}

Transaction Sequence:

Caller: [SOMEGUY], function: vault(address,uint256), txdata:
0x3fa16d990000000000000000000000000181251d3a501961d4af2af46e33c71a5d
808c25b000000000000000000000000000000000000000000000000000000000000
0000, value: 0x0
```