# Vendor Response to Tintin Report

Release notes for 0.11.3
https://medium.com/remix-ide/remix-release-0-11-3-86bbac18b2fa

Consensys Dillengence
https://consensys.net/diligence/vulnerabilities/remix-drive-by-and-remixd-path-traversal-and-rce/#timeline

Release 0.11.3
https://github.com/ethereum/remix-project/releases

## Section 1

The findings are presented in two distinct groups:

- a Remix-IDE cross domain communication issue
- and remixd service issues

This vulnerability note shows that any other website can drop file into a users' Remix IDE workspace without their knowledge or consent. Furthermore, we outline flaws in the local filesystem integration for Remix IDE and provide PoCs that show that the local filesystem daemon `remixd` provides no security guarantees to a user even though a `sharedFolder` was configured as a basedir and the service was instructed to only provide `readOnly` access. The issues found in `remixd` range from a low risk DoS vector, path traversal vectors that allow to read/list/write-what-anywhere, arbitrary remote function calls that allows the source website to change the basedir or even turn a `readOnly` share into `read/writeable`, to remote shell command execution.

# 2 RemixIDE - Cross-domain communication

### 2.1 Drive-by workspace manipulation without the users consent (high)

**REMIX UPDATED:**
Restrict / remove global cross-domain message handler. window.addEventListener('message')
PR: https://github.com/ethereum/remix-project/pull/1046

There had been a feature to allow users to copy their contracts to another instance of Remix. The UI had not been there for a long time - but this PR removes the old code for this feature.

# 3 remixd - WebSocket communication

## 3.1 Post Auth Denial Of Service (low)

**REMIX UPDATED:**
Post Auth Denial Of Service (low) PR: https://github.com/ethereum/remix-plugin/pull/349
The fix was about making sure that a bad request wouldn't crash the remixd daemon.

## 3.2 Websocket and UI relative path traversal (read/write-what-where) (critical)

- list any folder (outside basedir)
- write to any file/folder on disk (if not in readonly mode)

**REMIX UPDATED:**
Prevent creation of files outside workspaces from remixD provider (D) PR:
https://github.com/ethereum/remix-project/pull/1052

## 3.3 Origin can call arbitrary methods of `remixdClient.ts`/`gitClient.ts` - and remotely

## disable `readOnly` mode or change to a different basedir. (critical)

- change `sharedFolder` to any folder on disk without the users consent
    - Enforce sharedFolder restriction. (D) PR: https://github.com/ethereum/remix-project/pull/1052
        - **REMIX UPDATED**: Fix: Enforce sharedFolder restriction. (D) PR: https://github.com/ethereum/remix-project/pull/1052
- remotely remove read-only mode
    - **REMIX UPDATED**: Make readonly mode immutable after websocket instance is initialised. (D) PR: https://github.com/ethereum/remix-project/pull/1053
- call any other function of the client implementations
    - **REMIX UPDATED**: Disallow requests to methods not exposed (Y) PR: https://github.com/ethereum/remix-project/pull/1045 , https://github.com/ethereum/remix-plugin/pull/348

## 3.4 Arbitrary shell command injection (critical)

The origin can execute arbitrary shell commands on behalf of the user running remixd.

The filtering for commands can easily be bypassed by embedding subcommendas with backticks.

### REMIX UPDATED

**Fix:** Arbitrary shell command injection PR- disable git
https://github.com/ethereum/remix-project/pull/1047

Now the way to connect to Remixd is to put the parameter which will define which instance has the right to access Remixd.

## 3.5 General Remarks

- the design decision that the websocket service is unauthenticated (only protected by spoofable origin checks) is dangerous and may allow local privilege escalation
- communication from browser to service is not transport secured

* Drive-by workspace manipulation without the users consent (high)
=>  old code has been removed
https://github.com/ethereum/remix-project/pull/1046

* Post Auth Denial Of Service (low)
=> added a try catch to not crash the CLI when a msg cannot be read
https://github.com/ethereum/remix-plugin/pull/349

* Websocket and UI relative path traversal (read/write-what-where) (critical)
=> @ioedeveloper PR
https://github.com/ethereum/remix-project/pull/1052

* Origin can call arbitrary methods of remixdClient.ts/gitClient.ts - ....
=> fixed
https://github.com/ethereum/remix-project/pull/1045
,
https://github.com/ethereum/remix-plugin/pull/348

* Arbitrary shell command injection (critical)  => git plugin has been disabled until we have a safer way to handle communication between remix ide and remix
https://github.com/ethereum/remix-project/pull/1047

and from 3.5 General Remarks
* the design decision that the websocket service is unauthenticated (only protected by spoofable origin checks) is dangerous and may allow local privilege escalation
 => git disabled until we have a safer way to handle communication between remix ide and remix
* communication from browser to service is not transport secured
 => we aren't used wss, no solution right now

## Conclusion

Finally, we appreciate Tintin's clever,  thorough, detailed, and well documented work.  He has made Remix IDE better.