// HALBORN

dTrade

Frontend Penetration Test Executive Summary

Prepared by: **Halborn**

Date of Engagement: **June 10, 2021 - June 30, 2021**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	6/29/2021	Piotr Cielas
0.2	Document Edits	6/30/2021	Piotr Cielas
0.9	Final Version	6/30/2021	Piotr Cielas
1.0	Final Review	07/05/2021	Steven Walbroehl
1.1	Remediation Plan	07/05/2021	Piotr Cielas

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

dTrade is a decentralised exchange powered by Substrate and ZK-Rollups, and designed to function as a fully decentralized public utility, owned and controlled by the community of token holders.

dTrade engaged Halborn to conduct a security assessment on their frontend web applications beginning on June 10th, 2021 and ending June 30th, 2021. This security assessment was scoped to the frontend web applications only. Halborn was provided access to the source code of the application and the testing environment in order to conduct security testing using tools to scan, detect, validate possible vulnerabilities found in the application and report the findings at the end of the engagement.

Halborn recommends performing further testing to validate extended safety and correctness in context to the whole infrastructure when issues are fixed and new features added. Part of this recommendation includes a follow-up Black-Box audit that will be performed by a separate team of testers at Halborn to perform a red team perspective on the security.

1.2 AUDIT SUMMARY

Overall, it is noted that the frontend applications and their intended functionalities are well documented by the team at dTrade. The scope of the test was focused mainly on the JavaScript (React) code used, and the main features currently available to users. A staging version of the environment is deployed to allow a level of interaction and transaction activity between the three solutions. While the apps in their current form are relatively secure, Halborn recommends future testing to review the integration between the governance, insurance and bridge apps for any risks that may occur through communication between the three platforms.

All three applications leverage React, a popular JavaScript library for building user interfaces, limiting the frontend-backend data transmission and moving the business logic to the application client. Thus, all React and JavaScript vulnerabilities can be inherited by the applications. As

a result, the Halborn auditors checked all possible vulnerabilities in these technologies, and a review of the best practices used in React applications specifically.

The most significant security findings is PROPOSALS GET API DENIAL OF SERVICE. Other lower risk observations were documented by the security auditors, and are provided in this report to assist the developers in code quality, code coverage, and risk mitigation.

In summary, Halborn identified 8 security risks, and recommends performing further testing to validate extended safety and correctness in context to the whole set of assets in the dTrade ecosystem. External threats, such as economic attacks, oracle attacks, DoS attacks and inter-contract functions such as transactions, memos, and calls coming from the applications and relevant smart contracts should be validated for expected logic and state.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy regarding the scope of the pentest. While manual testing is recommended to uncover flaws in logic, process and implementation, automated testing techniques assist enhance coverage of the infrastructure and can quickly identify flaws in it.

The following phases and associated tools were used throughout the term of the audit:

- Mapping Application Content and Functionality
- Side/Browser Based Control Auditing
- Application Logic Flaws
- Access Handling
- Light Brute Force Attacks

- Input Handling
- Fuzzing of all input parameters
- Test for Injection (SQL/JSON/HTML/Command)
- Technology stack specific vulnerabilities and Code Audit
- Known vulnerabilities in 3rd party / OSS dependencies.

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident, and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. It's quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that was used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

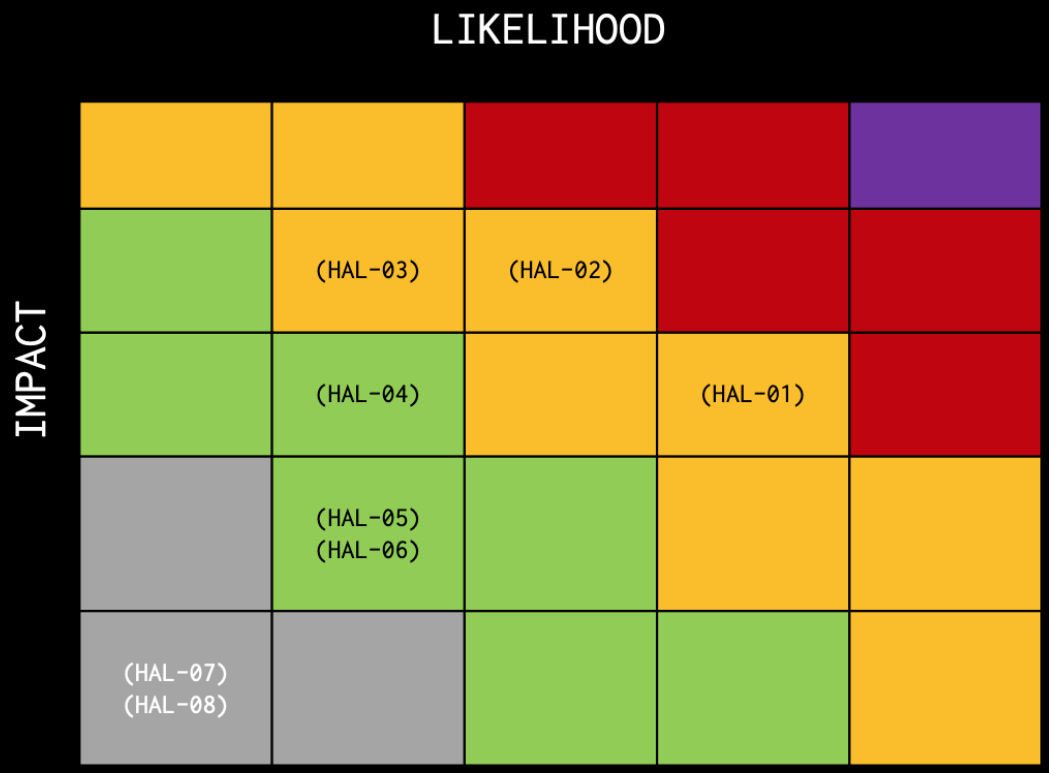
1.4 SCOPE

This security assessment was scoped to the following applications:

- FQDN: beta-bridge.dtrade.org
  - Repository: https://github.com/dtradeorg/edg-bridge-ui
  - Commit ID: c2dd18039f8a1f0b6d57e7a1a9e6a14a1088f0b1
- FQDN: beta-insurance.dtrade.org
  - Repository: https://github.com/dtradeorg/insurance-mining-ui
  - Commit ID: 99739307040cc1c080430a3accc5859b2ceafcfe
- FQDN: beta-gov.dtrade.org
  - Repository: https://github.com/dtradeorg/governance-ui
  - Commit ID: c7f7f5ac945e109c27f041d67f82485a4069ffde

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	3	3	2



EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) PROPOSALS GET API DENIAL OF SERVICE	Medium	SOLVED: 7/14/2021
(HAL-02) CLICKJACKING	Medium	SOLVED 7/13/2021
(HAL-03) USING COMPONENTS WITH KNOWN VULNERABILITIES	Medium	ACKNOWLEDGED: 7/13/2021
(HAL-04) MARKDOWN INJECTION	Low	SOLVED: 7/14/2021
(HAL-05) ACCESS CONTROL MISSING	Low	ACKNOWLEDGED: 7/13/2021
(HAL-06) HTTP RESPONSE HEADERS MISSING	Low	SOLVED: 7/13/2021
(HAL-07) INSECURE SECRET STORAGE	Informational	SOLVED: 7/11/2021
(HAL-08) COMPILER FLAGS MISSING	Informational	SOLVED: 7/13/2021

THANK YOU FOR CHOOSING

// HALBORN