ZEALYNX SECURITY

Penetration Testing

Hopium

Security Assessment

July 12th, 2024 - Prepared by Zealynx Security

Jose Fernando	@0xMrjory

Disclaimer

The information in this document is confidential and meant for use only by the intended recipient. Every effort has been made to ensure that the information contained in this document is true and correct at the time of publication. However, the products, specifications, and content in general that are described in this document are subject to continuous development and improvement, and therefore the reporter cannot accept liability for any loss or damage of any nature whatsoever arising or resulting from the use of or reliance on outdated information or particulars.

Changelog

Revision	Date	Change
1	June 10, 2024	Initial revision

Contents

Executive Summary

Summary

The reporter performed a security assessment of the hopium site between June 10, 2024 and June 17, 2024. The purpose of the assessment was to identify security vulnerabilities and recommend remediations.

The assessment was performed with a black-box, dynamic (browser based) approach.

The reporter does not warrant that the material contained in this documentation is free of errors, please note that it is not possible to find all vulnerabilities and vectors during an assessment. This report should be taken as-is and not as an exhaustive list of all security issues. With the ever-changing environment of information technology, tests performed will exclude vulnerabilities in software or systems that are unknown at the time of the assessment.

Methodology

The assessment is conducted with the following phases:

- Pre-engagement Interactions
- Enumeration
- Vulnerability Discovery
- Exploitation
- Post Exploitation
- Reporting
- Post-Engagement interaction

The reporter uses a combination of automated and manual methods and follows a testing methodology based on the PTES Technical Guidelines and OWASP Testing Guide.

Findings Overview

Vulnerability Classification (see Appendix A)	No. of active vulnerabilities
Critical	0
High	0
Medium	3
Low	1
Informational	1

Key Findings

The penetration test identified **5 VULNERABILITIES** that require remediation:

1. Business logic flaw
2. Input validation issue
3. Strict Transport not enforced
4. Rounding issues

5.. Redundant functionality

General Recommendations

To increase the security posture, the reporter recommends the following actions be taken:

1. Develop a plan of action and mitigation to remediate all other vulnerabilities according to a specific process of software patching. For more info: https://owaspsamm.org/model/operations/environment-management/stream-b/
2. Perform routine testing for the applications on a semi-annual basis.

Risk Score

The risk score for Bitsight response scan is 7 of a possible 25, which is rated at **LOW RISK**.

A LOW risk score indicates the target system or data is at a very low risk of being compromised and no immediate action is required. (See Risk score calculation at the end of the report)

Assessment Objectives

The security assessment attempted to gain information in three areas:

1. Identify security risks and gain system level access.
2. Identify areas of infrastructure weakness.
3. Recommend remediations to mitigate risks and eliminate vulnerabilities.

Assessment Scope

The assessment was performed on dedprz.virtual.tech and beecasino.virtual.tech.

Assessment Approach

The assessment was conducted in five phases:

1. Reconnaissance and information gathering.
2. Review reconnaissance data and perform analysis.
3. Using Tools like proxies and interceptors to test injections and other issues.
4. Assess systems and determine which may be vulnerable to exploitation.
5. Documentation of findings and recommendations.

Findings Summary

The tables below summarize vulnerabilities discovered during the assessment. More information about vulnerability classification can be found in PortSwigger.

	Severity	Remediated	Finding
	Medium	No	Disable deposit function
	Medium	No	Input validation in deposit
	Medium	No	Decimals not required in deposit
	Informational	No	Strict transport not enforced
	Informational	No	Reset function is redundant

Application summary

The Hopium website (hopium.virtual.tech) is a Web2 application powered by Virtual Rollups, which are ZeroGas rollups offering one-click trading, millisecond finality, and user-validated execution. This platform is specifically designed to integrate with a smart contract on the backend. It enables players to participate in a coin flip game where bets are placed according to options provided by the interface. These bets are deducted from the session balance of a connected wallet. Players choose between options such as astroman or alien (equivalent to heads or tails), specifying their bet amount. They then await the application's response to determine if they have won or lost. Depending on the outcome, their wallet balance is adjusted accordingly, either increasing or decreasing based on the result of the coin flip game.

Technical Details

Deposit function should be disabled before starting a game

Summary

Enabling the deposit function before initiating the game with "Start Game" can lead to unnecessary transactions and potential gas griefing.

Classification

Medium : CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:H/A:L

Affected Hosts

hopium.virtual.tech

Recommendation

Ensure that the deposit function is disabled for the user before starting a game.

Lack of input validation in deposit function will trigger 0 value transactions

Summary

The deposit function can trigger transactions even if the amount field is empty or if the user inputs special characters instead of numbers. These transactions result in gas griefing by consuming gas without transferring any value.

Classification

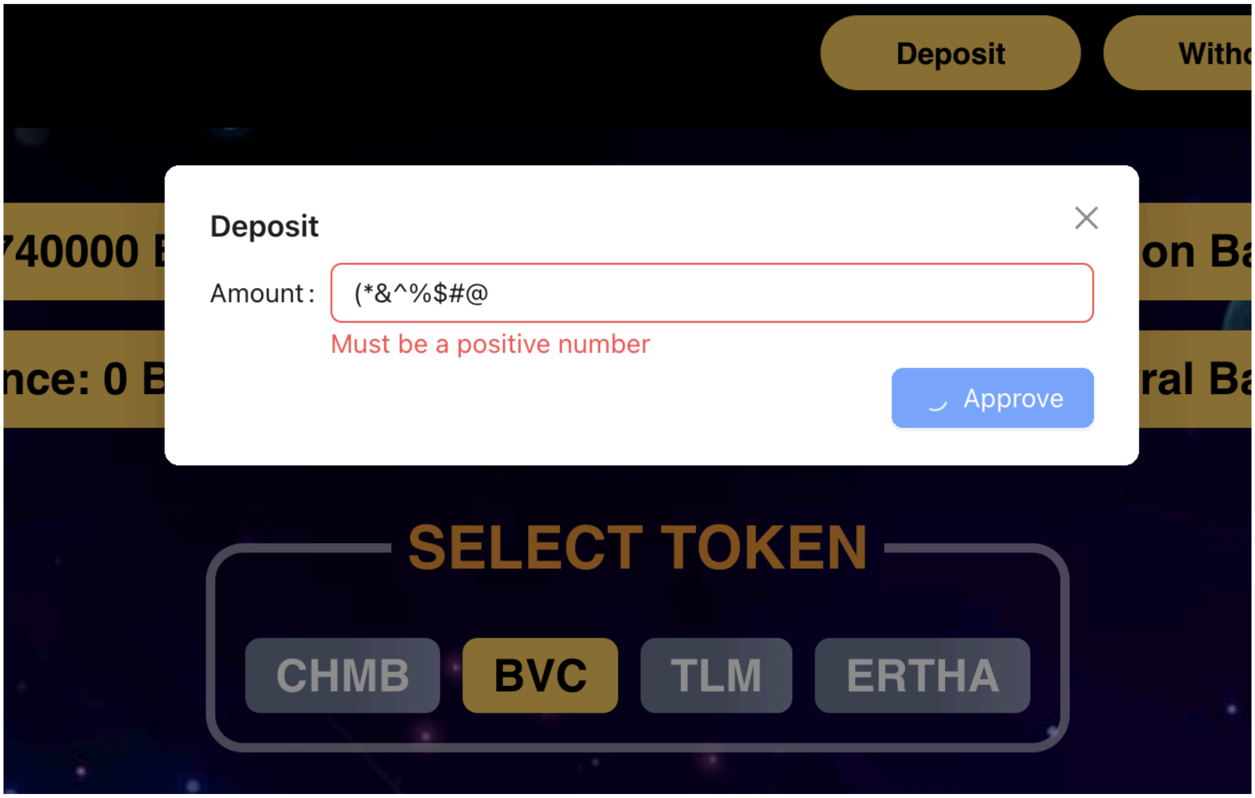
Medium : CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:H/A:L

Affected Hosts

hopium.virtual.tech

Proof of Concept

When a user is depositing assets, the amount field is not properly sanitized to accept only numeric values. Although there is a warning message stating that the input should be a positive number, a user can still input any value (or leave it empty), triggering a zero-value transaction that only consumes gas. Moreover this transaction is occurring without Starting a game which is required before depositing an amount.

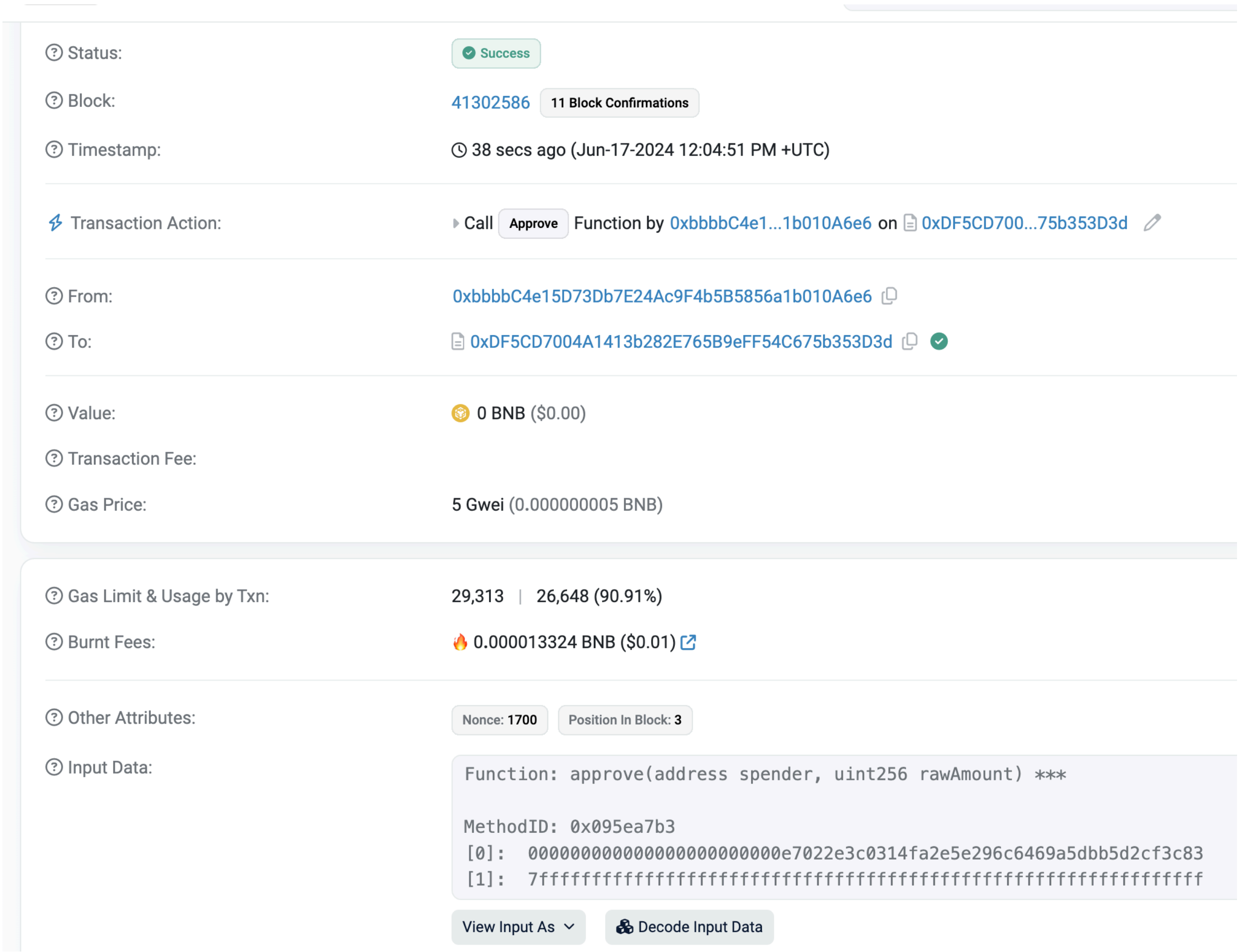


Deposit

Amount : (*&^%$#@

Must be a positive number

In the image: The current implementation allows a user to introduce special characters or an empty string, and the approve button will still process a transaction with a zero value, consuming only gas.



Status: Success

Block: 41302586 11 Block Confirmations

Timestamp: 38 secs ago (Jun-17-2024 12:04:51 PM +UTC)

Transaction Action: Call Approve Function by 0xbbbbC4e1...1b010A6e6 on 0xDF5CD700...75b353D3d

From: 0xbbbbC4e15D73Db7E24Ac9F4b5B5856a1b010A6e6

To: 0xDF5CD7004A1413b282E765B9eFF54C675b353D3d

Value: 0 BNB ($0.00)

Transaction Fee:

Gas Price: 5 Gwei (0.000000005 BNB)

Gas Limit & Usage by Txn: 29,313 | 26,648 (90.91%)

Burnt Fees: 🔥 0.000013324 BNB ($0.01)

Other Attributes: Nonce: 1700 Position In Block: 3

Input Data:

```
Function: approve(address spender, uint256 rawAmount) ***

MethodID: 0x095ea7b3
[0]:  000000000000000000000000e7022e3c0314fa2e5e296c6469a5dbb5d2cf3c83
[1]:  7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
```

In the image: Details of the 0 value transaction from above.

Although the user is required to approve this zero-value transaction, if the application warns the user to use only valid positive numbers, it should enforce this by accepting only valid input for approval.

Recommendation

Ensure that user input is restricted to only the expected values (numbers) before allowing the approval of any transaction.

Decimals are not required in the deposit function

Summary

When a user deposits an amount into any of the tokens, allowing input of decimal numbers is unnecessary since the wages are fixed in rounded numbers.

Classification

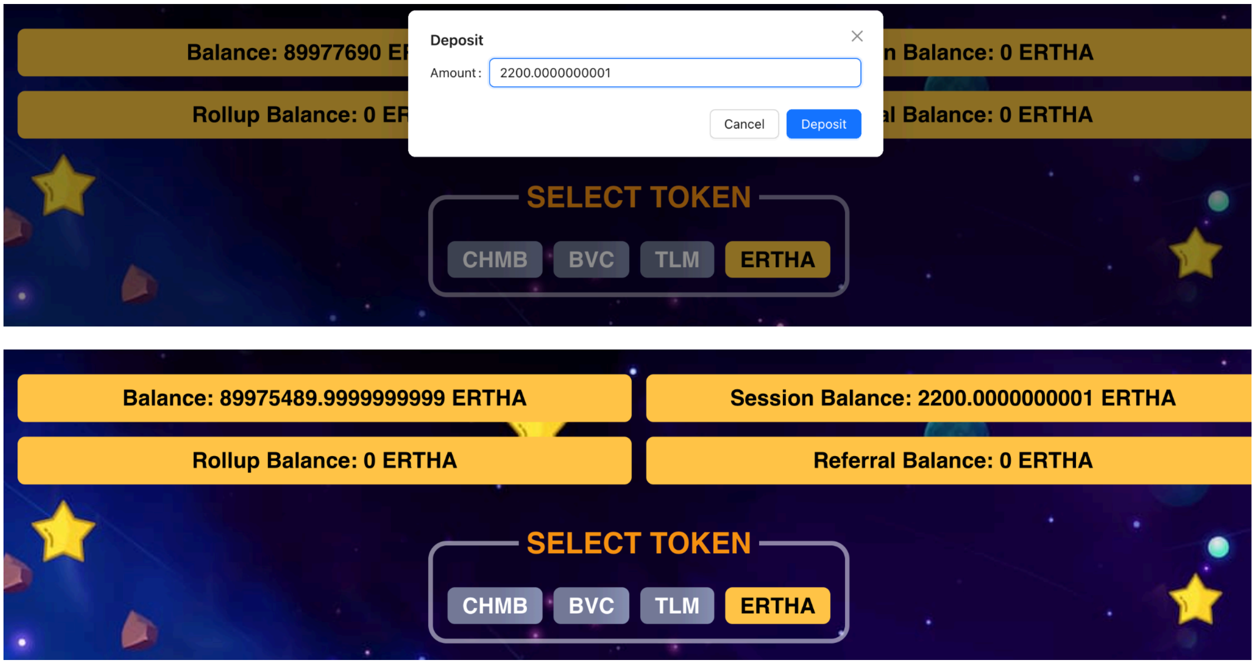
Medium: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Affected Hosts

hopium.virtual.tech

Proof of Concept

When a user initiates a game and proceeds to deposit, it is possible to enter amounts with decimals.



Balance: 89977690 EF

n Balance: 0 ERTHA

Rollup Balance: 0 EF

al Balance: 0 ERTHA

Deposit

Amount: 2200.0000000001

Balance: 89975489.9999999999 ERTHA

Session Balance: 2200.0000000001 ERTHA

Rollup Balance: 0 ERTHA

Referral Balance: 0 ERTHA

In the image: The application processes decimals in the Session Balance.

Although the application seems to manage calculations involving decimal numbers in the rollup/withdrawal process, allowing users to input decimal numbers is unnecessary. This is because the application enforces a fixed wager amount for each available token.

Leaving the decimals might cause rounding issues in the backend, in this way the input must be properly constrained.

Recommendation

Consider restricting users from entering decimal amounts when making deposits.

Strict transport security not enforced.

Summary

The application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The sslstrip tool automates this process.

To exploit this vulnerability, an attacker must be suitably positioned to intercept and modify the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Classification

Informational

Affected Hosts

hopium.virtual.tech

Recommendation

The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.

Note that because HSTS is a "trust on first use" (TOFU) protocol, a user who has never accessed the application will never have seen the HSTS header, and will therefore still be vulnerable to SSL stripping attacks. To mitigate this risk, you can optionally add the 'preload' flag to the HSTS header, and submit the domain for review by browser vendors.

References

- HTTP Strict Transport Security

Reset function is redundant

Summary

The Reset function is redundant as it merely changes the selection from “Alien” to “Astroman,” a task that users can already perform through the existing user interface.

Classification

Informational

Affected Hosts

hopium.virtual.tech

Recommendation

If the Reset function only changes the selection, it can be removed to streamline and simplify the user interface.

Appendix A

Vulnerabilities Classification (based on Guidelines for Applying Security Patches)

Classification	Description
Catastrophic	This rating is assigned by the Vice President of Information Risk and Security. The rating will be rarely invoked, primarily for Known Exploited Vulnerabilities. Once a vulnerability is declared as Catastrophic, strict limits apply to the expected remediation timeline. Mitigations such as securing an asset behind a WAF or firewall and actively monitoring logs are the recommended immediate response.
Critical	This rating is given to flaws that could be easily exploited by a remote unauthenticated attacker and lead to system compromise (arbitrary code execution) without requiring user interaction. These are the types of vulnerabilities that can be exploited by worms. Flaws that require an authenticated remote user, a local user, or an unlikely configuration are not classed as critical impact.
Important	This rating is given to flaws that can easily compromise the confidentiality, integrity, or availability of resources. These are the types of vulnerabilities that allow local users to gain privileges, allow unauthenticated remote users to view resources that should otherwise be protected by authentication, allow authenticated remote users to execute arbitrary code, or allow local or remote users to cause a denial of service.
Medium	This rating is given to flaws that may be more difficult to exploit but could still lead to some compromise of the confidentiality, integrity, or availability of resources, under certain circumstances. These are the types of vulnerabilities that could have had a critical impact or important impact but are less easily exploited based on a technical evaluation of the flaw, or affect unlikely configurations.

Low	This rating is given to all other issues that have a security impact. These are the types of vulnerabilities that are believed to require unlikely circumstances to be able to be exploited, or where a successful exploit would give minimal consequences.

Risk rating scale calculation (based on NIST and CVSS 3.1)

Rating	Scale
Critical	9 – 10
High	7.0 - 8.9
Medium	4.0 - 6.9
Low	0 - 3.9

Risk score calculation

Risk Score is calculated using the following equation:

Risk = Likelihood * Business Impact,

where Likelihood is calculated by taking the highest CVSS rated issue and dividing the rating by two, and Business Impact is selected from range 1 to 5 based on the following table.

In this case 5.1 was the highest divided by 2 = 2.55 with BI of 2 = 5.1 of risk score which is Low.

Business Impact	Explanation
1	System is isolated, only Public data and no means of pivoting or reputational harm being done
2	Public data, and no reputation harm when system compromised
3	Internal data with limited impact, pivoting only to systems with BI 1 and almost no reputation harm when system compromised
4	Internal data with limited impact, pivoting only to system with BI 3, moderate reputation harm when system compromised
5	Restricted(+PII) or customer data, or system can be used for pivoting to get to system with high value data/resources, or can be used for great harm reputation

Qualitative Risk Score	Qualitative Risk Level	Rating Definition
25	Critical	Critical risk score indicates the target system or data is at a very high risk of being compromised, and immediate action is required.
15 - 24	High	High risk score indicates the target system or data is at a significant risk of being compromised, and prompt action is required.
8 - 14	Medium	Medium risk score indicates the target system or data is at a moderate risk of being compromised, and ongoing monitoring and improvement is recommended.
4 - 7	Low	Low risk score indicates the target system or data is at a low risk of being compromised, but ongoing monitoring and improvement is recommended.
1 - 3	Very Low	Very Low risk score indicates the target system or data is at a very low risk of being compromised and no immediate action is required.