VULNERABILITY
REWARD PROGRAM
ODUWACOIN

REWARD PROGRAM

1. Avoid using web application scanners for automatic vulnerability searching which generates massive traffic

2. Make every effort not to damage or restrict the availability of products, services, or infrastructure

3. Avoid compromising any personal data, interruption, or degradation of any service

4 Do not access or modify other user data, localize all tests to your accounts

5. Perform testing only within the scope

6. Do not exploit any DoS/DDoS vulnerabilities, social engineering attacks or spam

7. Do not spam forms or account creation flows using automated scanners

8.In case you find chain vulnerabilities we will pay only for vulnerability with the highest severity.

9.Do not break any law and stay in the defined scope

10.Any details of found vulnerabilities must not be communicated to anyone who is not a Oduwacoin Team or an authorized developer without appropriate permission

11.Give us enough time to fix the vulnerability before distributing information about it in any way.

12.During testing, make the necessary efforts to avoid causing any damage to the exchange and its users.

13.Provide the most comprehensive information about the vulnerability so that we can quickly fix it.

14. Do not share any vulnerability or make bug information public

Reward ranges from

$50 to $1000

IN-SCOPE VULNERABILITIES

We are interested in the following vulnerabilities:

Business logic issues

Payment's manipulation

Remote code execution (RCE)

Database vulnerability, SQLi

File inclusions (Local & Remote)

Access Control Issues (IDOR, Privilege Escalation, etc)

Leakage of sensitive information

Server-Side Request Forgery (SSRF)

LFI/LFI/SQLi

Balance manipulation

Partial authentication bypass

Theft of privileged information

Other vulnerability with clear potential for financial or data loss

In general, do not correspond to the severity threshold:

Ddos

Self xss

Spam

Social engineering

IN-SCOPE VULNERABILITIES

OUT OF SCOPE - WEB

Vulnerabilities found in out-of-scope resources are unlikely to be rewarded unless they present a serious business risk (at our sole discretion). In general, the following vulnerabilities do not correspond to the severity threshold:

Vulnerabilities in third-party applications

Best practices concerns

Recently (less than 30 days) disclosed 0day vulnerabilities

Vulnerabilities affecting users of outdated browsers or platforms

Social engineering, phishing, physical, or other fraud activities

Publicly accessible login panels without proof of exploitation

Reports that state that software is out of date/vulnerable without a proof of concept

Vulnerabilities involving active content such as web browser add-ons

Most brute-forcing issues without clear impact

Denial of service

Theoretical issues

Moderately Sensitive Information Disclosure

Spam (sms, email, etc)

Missing HTTP security headers

Infrastructure vulnerabilities, including:

Certificates/TLS/SSL related issues

DNS issues (i.e. MX records, SPF records, etc.)

Server configuration issues (i.e., open ports, TLS, etc.)

Open redirects

User account enumeration

Clickjacking/Tap jacking and issues only exploitable through clickjacking/tap jacking

Descriptive error messages (e.g. Stack Traces, application or server errors)

Self-XSS that cannot be used to exploit other users

Login & Logout CSRF

Weak Captcha/Captcha Bypass

Lack of Secure and HTTP Only cookie flags

Username/email enumeration via Login/Forgot Password Page error messages

CSRF in forms that are available to anonymous users (e.g., the contact form)

OPTIONS/TRACE HTTP method enabled

Host header issues without proof-of-concept demonstrating the vulnerability

Content spoofing and text injection issues without showing an attack vector without being able to modify HTML/CSS

Content Spoofing without embedded links/HTML

Reflected File Download (RFD)

Mixed HTTP Content

HTTPS Mixed Content Scripts

DoS/DDoS issues

Manipulation with Password Reset Token

MitM and local attacks

OUT OF SCOPE - MOBILE

Attacks requiring physical access to a user's device

Vulnerabilities requiring extensive user interaction

Exposure of non-sensitive data on the device

Reports from static analysis of the binary without PoC that impacts business logic

Lack of obfuscation/binary protection/root(jailbreak) detection

Bypass certificate pinning on rooted devices

Lack of Exploit mitigations i.e., PIE, ARC, or Stack Canaries

Sensitive data in URLs/request bodies when protected by TLS

Path disclosure in the binary

OAuth & app secret hard-coded/recoverable in IPA, APK

Sensitive information retained as plaintext in the device's memory

Crashes due to malformed URL Schemes or Intents sent to exported Activity/Service Broadcast Receiver (exploiting these for sensitive data leakage is commonly in scope)

Runtime hacking exploits using tools like but not limited to Frida/ Appmon (exploits only possible in a jailbroken environment)

Shared links leaked through the system clipboard

Any URIs leaked because a malicious app has permission to view URIs opened.

Exposure of API keys with no security impact (Google Maps API keys etc.)

If you don't have the time to peruse through the report, below are some of its key findings:

- 58% of bug bounty hackers are self-taught
- 37% of white-hat hackers say they hack as a hobby in their spare time (not their primary job).
- About 12% of hackers on HackerOne make $20,000 or more annually from bug bounties.
- Over 3% o bug hunters are making more than $100,000 per year.
- 13.7% say bounties earned represent 90-100% of their annual income.
- India (23%) and the United States (20%) are the top two countries represented on the HackerOne platform, followed by Russia (6%), Pakistan (4%), and the United Kingdom (4%)
- Nearly 1 in 4 hackers have not reported a vulnerability that they found because the company didn't have a channel to disclose it.
- US companies have paid over $15 million to bug hunters via HackerOne in 2017
- US bug hunters racked over $4.1 million in bug rewards, while Indian white-hat hackers earned over $3 million.
- "Websites" was the overwhelming winner to the question of "What is Your Favorite Kind of Platform or Product to Hack?" with a 70.8% score
- "Money" was not the primary motivation for getting into bug hunting. It ranked only fourth.
- XSS was the favorite vulnerability white-hat hackers liked to search for.
- Almost 30% of respondents said they use Burp Suite for hunting bugs. Other ranked tools include: