LD

Bug bounty [2025] EN

A LiveDune program that offers rewards to people who find and responsibly report bugs and security
vulnerabilities in our software or systems.

Structure

Scope

1. Where to look for vulnerabilities?→
2. What vulnerabilities to look for?→
3. What not to submit?→
4. What is not eligible for a reward?→

Rewards

1. Payout table →
2. Vulnerability severity assessment →

Participation rules

1. General rules →
2. Testing rules →
3. Vulnerability disclosure policy →
4. RCE testing policy →
5. SQL injection testing policy →
6. File upload & file read testing policy →

3. Notes →

FAQ

1. How to write a good report? →
2. Do you provide test credentials? →
3. How and when are reports reviewed? →
4. How do you handle duplicates? →

Scope

Where to look for vulnerabilities?

You may look for vulnerabilities on our resources, specifically: ***.livedune.com**.

However, maximum payouts are provided only for issues found in services listed as **in-scope core services**. For all other resources, rewards may be reduced (in some cases, down to zero).

What vulnerabilities to look for?

We are primarily interested in finding critical vulnerabilities on our servers and services, but we also welcome other types of security issues. If you are unsure whether you should report an issue, check whether it is listed under **“What not to submit?”** If it is not listed there, please submit a report with a detailed description.

Instructions on how to report a vulnerability effectively can be found in the **FAQ** section.

- Remote Code Execution (RCE)
- Injection issues (e.g., SQL injection or XML injection)
- LFR/LFI/RFI

- SSRF
- Memory disclosures/leaks
- Business logic vulnerabilities
- IDOR
- Access control vulnerabilities
- Sensitive information disclosure
- Account takeover
- Authentication/authorization weaknesses
- XSS and CSRF that affect sensitive data

What not to submit?

- Issues not related to security
- Reports about vulnerabilities in services not related to LiveDune
- Reports about possible DDoS attacks
- Information about IP addresses, DNS records, and open ports
- Reports from vulnerability scanners and other automated tools
- Reports about publicly accessible login panels

What is not eligible for a reward?

- Self-XSS; XSS in unpopular or outdated browsers; Flash-based XSS
- CSRF and XSS without impact on sensitive data
- Absence of recommended protection mechanisms* (e.g., security HTTP headers, cookie security flags, CSRF protection)
- CORS misconfigurations*
- Use of outdated or potentially vulnerable third-party software*
- Fraud- or theft-related attacks
- Unlimited SMS/email sending

- DoS attacks*
- Insecurely configured TLS or SSL*
- User/email/phone enumeration (disclosing the existence of a username, email, or phone number in the system)
- Full Path Disclosure
- Disclosure of technical or non-sensitive information* (e.g., product/software versions, stack traces)
- Open Redirect without an additional attack vector (e.g., auth token theft)
- Content spoofing/page defacement
- Tabnabbing
- Vulnerabilities requiring complex or unlikely user interaction
- Vulnerabilities that require malware, root access, or a jailbreak to exploit
- Attacks requiring MITM of someone else's connection or physical proximity to someone else's device (e.g., NFC/Bluetooth/Wi-Fi attacks, shoulder surfing)
- Unlikely or purely theoretical attacks without proof of feasibility

> 💡 (*) Items marked with an asterisk require a detailed attack vector description and evidence of potential damage/harm; otherwise, they are not eligible for a reward.

Participation rules

By participating in our bug bounty program, you confirm that you have read and agreed to these **Participation Rules**. Violation of any rule may result in disqualification and loss of reward eligibility.

General rules

- If you encounter issues that are not security-related, please contact customer support. The bug bounty program covers technical security vulnerabilities only.

- If you find a 0-day or 1-day vulnerability whose patch was released less than a week ago, the reward will be considered by the LiveDune security team on a case-by-case basis.

Testing rules

- Use only your own accounts or accounts of users who have explicitly consented. Do not attempt to access other users' accounts or any confidential information.
- Avoid compromising the confidentiality, integrity, or availability of information in our services.
- Any activity that may harm LiveDune applications, infrastructure, customers, or partners is prohibited (e.g., social engineering, phishing, denial-of-service attacks, physical attacks).
- Use the minimal PoC required to demonstrate the vulnerability. If your PoC could affect other users or system stability, contact us for permission. Further exploitation is strictly prohibited.
- Automated scanning must be limited to **3 requests per second**.

Vulnerability disclosure policy

- You must not disclose vulnerabilities or share any details without written permission from the LiveDune security team.
- We reserve the right to deny any request for public disclosure.

RCE testing policy

Testing vulnerabilities that may lead to Remote Code Execution (RCE) must follow this policy.

During testing, the only allowed server actions are:

- Execute commands: `ifconfig` (`ipconfig`), `hostname`, `whoami`
- Read files: `/etc/passwd` and `/proc/sys/kernel/hostname` (Windows equivalents: `drive:/boot.ini`, `drive:/install.ini`)
- Create an empty file in the current user's directory

Any other actions must be pre-approved by the LiveDune security team.

SQL injection testing policy

Testing vulnerabilities that may lead to SQL command injection must follow this policy. During testing, the only allowed actions are:

- Retrieve metadata about the current database (e.g., `SELECT database()`), version (`SELECT @@version`), current user (`SELECT user()`, `SELECT system_user()`), or host name (`SELECT @@hostname`).
- Retrieve schema information (e.g., `table_schema`), list tables (`table_name`), and column names (`column_name`).
- Run mathematical, conversion, or logical queries (including using `SLEEP`) without extracting data (except the metadata listed above).

Any other actions must be pre-approved by the LiveDune security team.

File upload & file read testing policy

Testing vulnerabilities that may allow arbitrary file reads or file uploads must follow this policy.

Prohibited actions when uploading files:

- Modify, delete, replace, or tamper with any files on the server (including system files), except files associated with your own account or a user who has explicitly consented.
- Upload files that may cause denial of service (e.g., very large files).
- Upload malicious files (e.g., malware or spyware).

If you gain the ability to read arbitrary files on the server, you may only read `/etc/passwd` and `/proc/sys/kernel/hostname` (Windows equivalents: `drive:/boot.ini`, `drive:/install.ini`). Any other actions must be pre-approved.

Rewards

We pay rewards to external security researchers only for previously unknown issues and only if all Participation Rules are followed.

All reports are evaluated individually, taking into account the criticality of the affected system and the criticality of the vulnerability itself. The table below shows maximum rewards.

Payout table

Criticality	**Critical**	**High**	**Medium**	**Low**
Maximum payout	Up to $1,000	Up to $500	Up to $250	Up to $100

How we determine severity

We reserve the right to make the final decision regarding the severity of a reported vulnerability. After receiving a report, we conduct an internal investigation and determine severity, considering factors including:

- Privilege level required to carry out the attack
- Difficulty of discovery and exploitation
- Whether user interaction is required
- Impact on confidentiality, integrity, and availability of affected data
- Business and reputational impact
- Number of affected users

Notes

- Contact email: `bug-hunters@livedune.com`
- Use the subject: **“bug-hunter”**
- A reward is paid only if the LiveDune security team determines that all participation conditions were met and the vulnerability is significant.
- The reward amount is final and not subject to discussion.

FAQ

How to write a good report?

One report should describe one vulnerability. An exception is when vulnerabilities are linked together or can be chained.

A good vulnerability report should include:

- Vulnerability description
- Steps to reproduce
- Severity analysis
- Remediation recommendations

The report should also contain:

- Vulnerable application URL
- Vulnerability type
- Screenshots or a video showing the reproduction steps
- An example of a formatted request from Burp Suite (or any other PoC)
- Code snippets (when applicable)

Failure to meet minimum requirements may result in a reduced reward. If there is not enough information to verify the vulnerability, no reward will be paid.

All information about the vulnerability (including attachments) should be stored only within the report you submit. Do not publish it on external resources.

Do you provide test credentials?

We do not provide additional access or credentials (including test accounts). Please use your own accounts for testing.

How and when are reports reviewed?

Vulnerability reports are reviewed by our internal security team. Response time depends on workload, but we aim to process requests within **two weeks**.

How do you handle duplicates?

We pay a reward only for the first received report (provided it contains all necessary information to reproduce the issue). Subsequent reports addressing the same vulnerability are treated as duplicates and are **not** rewarded. Similar attack

vectors may also be treated as duplicates if a single report contains enough information to fix all related vectors.