Bug Bounty Program

Introduction

Cloudsmith aims to keep its Service safe, and data security is of utmost priority. Suppose you are a security researcher and have discovered a security vulnerability in the Service. In that case, we appreciate your help in disclosing it to us privately and letting us fix it before publishing technical details.

Cloudsmith will engage with security researchers when vulnerabilities are reported to us, as described here. We will validate, respond to, and fix vulnerabilities to support our commitment to security and privacy. We won't take legal action against, suspend, or terminate access to the Service of those who responsibly discover and report security vulnerabilities. Cloudsmith reserves all of its legal rights in the event of any noncompliance.

We are pleased to offer thanks and/or a bounty reward for vulnerability information that helps us protect our customers thanks to the security researchers who choose to participate in our bug bounty program. This will range from the public thanks to a variable monetary amount or loot (depending on our budget and the impact).

1. Rules and Process

- If you identify a verified security vulnerability in compliance with this Security Disclosure Policy (i.e. Bug Bounty Program), you can submit this to our bug bounty program with the following process:
- You should determine if an exploit is viable (i.e., practically possible).
- You should also check to ensure it's not explicitly out-of-scope (see below).
- Please feel free to report the exploit to us using the relevant information (see below).
- We will let you know we received the disclosure and triage impact and then schedule a review.
- The urgency for review will be determined based on the type and impact of the exploit.
- It may take up to one month to triage, depending on how busy the team is (we're not huge!)
- Timeframes are typically hours for critical/high, days for medium and (up to) weeks for low.
- Please do NOT send continual "any update?" messages; we get reminded automatically.
- Any disrespectful communications to our staff including but not limited to frequent checkups on the status of a report will make the report ineligible for any award.
- We will let you know if the exploit is viable and whether we accept it as a bug bounty (or not).
- If viable and fixed, we will publicly thank you on our Hall of Fame.
- You may also be issued with an award in addition to being placed in the Hall of Fame.

N.B. if the same exploit is reported by multiple reporters prior to remediation, then the first valid report that documents the exploit will receive the credit.

1.1. Unauthorized Cross-Account Access

Please note that in absolutely no event are you permitted to access, download, or modify data residing in any other Account or one that is not registered to you unless permission is expressly provided by the Account Owner.

1.2. Monetary Award Tax

If a monetary award is provided as a bounty, you are responsible for paying any taxes associated with the reward. We are obliged by law to report any such awards to the tax authorities, so please make sure that you handle this appropriately.

1.3. In Scope

The bug bounty scope applies to:

- the Cloudsmith Classic App (the "old" app) served at https://cloudsmith.io
- the Cloudsmith Web App (the "new" app) served at https://app.cloudsmith.com
- the Cloudsmith Core API served at https://api.cloudsmith.io
- the Cloudsmith Corporate Site served at https://cloudsmith.com

1.3.1. Vulnerability Classifications

The vulnerability classifications that are considered in-scope are:

- Access Control Bypass;
- API Misuse Issues;
- Authentication (Broken or Bypassed);
- Business Logic Issues;
- Cross-Site Scripting (XSS);
- Cross-Site Request Forgery (CSRF);
- Complexity Bomb;
- Decompression Bomb;
- Directory Traversal;
- Improper TLS protection;
- Open URL Redirection;
- Privilege Escalation;
- Provisioning Errors;
- Remote Code Execution (RCE);
- Sensitive/Private Data Leaks;
- Session Fixation;
- Session Management (Broken or Bypassed);
- Subdomain/Domain Takeover;
- SQL Injection.

If a vulnerability isn't explicitly listed here as either in-scope or explicitly mentioned in the next section as out-of-scope, then it might be applicable. Still, we reserve the right to determine this in communication with the security researcher.

1.4 Out Of Scope

Application bugs that are explicitly NOT considered to be in-scope:

- Any standard (i.e., non-security) application or API bugs (e.g., display issues, ordering issues, search issues, input/output issues, etc.), unless they involve privilege escalation or cross-account access.
- Any UI-based injection bugs in the Classic App (e.g., XSS or CORS issues); however, these are now applicable within the new Web App, which was released for preview in November 2024.
- Actions/areas that are explicitly NOT considered to be in-scope:
-
- Executing or attempting to execute any Denial of Service (DoS) attack; or
- Knowingly posting, transmitting, uploading, linking to, sending, or storing any Malicious Software; or
- Attempting to social engineer support or other staff; or
- Testing in a manner that would result in the sending of unsolicited or unauthorized junk mail, spam, pyramid schemes, or other forms of duplicative or unsolicited messages; or
- Testing in a manner that would degrade the operation of the Service; or
- Testing or otherwise accessing or using the Service from any jurisdiction that is Prohibited Jurisdiction; or
- Testing third-party applications, websites, or services that integrate with or link to the Service.

Vulnerabilities that are explicitly NOT considered to be in-scope:

- Clickjacking, CSRF, and content spoofing issues without demonstrable security impact.
- Publicly known vulnerable libraries without a working Proof of Concept.
- Self-XSS or having a user paste JavaScript into the browser console.
- Issues relating to non-Cloudsmith products.
- Open ports scanning, banner grabbing, and software version disclosure issues.
- Issues that affect only outdated user agents or unsupported platforms.

Non-qualifying best practices that are explicitly NOT considered to be in-scope:

- Missing cookie flags on non-authentication cookies.
- Missing best practices in DNS configuration (e.g., DKIM/DMARC/SPF/TXT).
- Missing best practices in SSL/TLS configuration.
- Missing best practices in Content Security Policy (CSP) or lack of other security-related headers.
- Leakage of sensitive tokens (e.g., reset password token) to trusted third parties on a secure connection (HTTPS).

Non-qualifying business practices that are explicitly NOT considered to be in-scope:

- Institution access code enumeration or demonstrating access codes leaked in internet forums.
- Credential re-usage from public dumps.
- UUID enumeration of any kind.
- Ability to determine if a username or email has a Cloudsmith account.
- Signing up with multiple accounts to abuse referral code usage.
- Password length, complexity, and re-use requirements.
- Email verification feature.

Websites/services that are explicitly NOT considered to be in-scope:

- Any "third-party" website that we use to host a service but do not control uses their bug bounty program, e.g.:
- Help Website (https://help.cloudsmith.io) - this is hosted by https://readme.io.
- Status Website (https://status.cloudsmith.io) - this is hosted by https://statuspage.io.
- Support Website (https://support.cloudsmith.com) - this is hosted by https://zendesk.com.
- Amazon Web Services (https://aws.amazon.com) - this is our infrastructure provider.
- GRC / Trust Centre Website (https://trust.cloudsmith.com) - this is hosted by https://www.vanta.com/.
- Cloudsmith non-production systems; only production systems are in scope for the bug bounty program.

Any third-party widgets, such as:

- Chat widget (powered by Intercom, Zendesk or Qualified).
- Status widget (powered by Incident.io).

Any websites under the following domains:

- *.readme.io (incl. dash.readme.io and cloudsmith.readme.io)
- *.googleapis.com
- *.googletagmanager.com
- *.google-analytics.com
- *.doubleclick.net
- *.hubspot.com
- *.bing.com
- *.statuspage.io
- *.simpleanalytics.com
- *.qualified.com
- *.zendesk.com
- *.incident.io

1.5 Severity

When we issue bug bounties, we assign them a severity level. This represents the level at which we assess the impact of the disclosed security issue to be, and it influences the bug bounty award. Please remember, any decisions for assessed severity and impact are final, and we reserve the right to determine them.

SEVERITY LEVEL	HEADER
CRITICAL	Vulnerabilities that score in the critical range usually have most of the following characteristics: • Exploitation of the vulnerability likely results in root-level compromise of servers or infrastructure devices. • Exploitation is usually straightforward, in the sense that the attacker does not need any special authentication credentials or knowledge about individual victims, and does not need to persuade a target user, for example via social engineering, into performing any special functions.
HIGH	Vulnerabilities that score in the high range usually have some of the following characteristics: • The vulnerability is difficult to exploit. • Exploitation could result in elevated privileges. • Exploitation could result in significant data loss or downtime.
MEDIUM	Vulnerabilities that score in the medium range usually have some of the following characteristics: • Vulnerabilities that require the attacker to manipulate individual victims via social engineering tactics. • Denial of service vulnerabilities that are difficult to set up. • Vulnerabilities where exploitation provides only very limited access. • Vulnerabilities that require user privileges for successful exploitation.
LOW	Vulnerabilities in the low range typically have little impact on an organization's business.

We endeavor to try and fix security bugs within the following timelines:

SEVERITY LEVEL	TYPICAL MEAN TIME TO FIX (MTTF)
CRITICAL	Within 2 weeks of being verified
HIGH	Within 4 weeks of being verified
MEDIUM	Within 6 weeks of being verified
LOW	Within 26 weeks of being verified

1.6 Reporting

To start the reporting process, please share the details of any suspected vulnerabilities with the Cloudsmith Security Team by sending us the details to bug-bounties+nospam@cloudsmith.io (remove the +nospam to reach us). Please do not publicly disclose these details outside this process without explicit permission.

In reporting any suspected vulnerabilities, please include the following information:

ITEM	DESCRIPTION
Vulnerable URL	The URL where the vulnerability occurs
Vulnerable Parameter	If applicable, the parameter where the vulnerability occurs
Vulnerability Type	The type of vulnerability
Steps to Reproduce	Step-by-step information on how to reproduce the issue
Screenshots	A demonstration of the attack to aid description
Attack Scenario	An example attack scenario may help demonstrate the risk and resolve your issue faster

1.6 Reserved Rights

Cloudsmith reserves all rights to decide on scope, impact, and reward.

If another researcher has previously reported an exploit, only the first disclosure will be considered. However, you'll always have our appreciation, and we'll let you know if this happens. This often occurs due to the number of submissions that we get.

If we cannot replicate the issue, regardless of the timeframe between the report and our triage, we cannot award any bounties. We'll try our best to replicate this, but please know that we don't receive reports, so we fix them without acknowledgement.

Finally, regardless of the reason, we do not tolerate abuse or threats to our staff or company. We reserve the right to terminate and block all communication with such people. Everyone should acknowledge and respect that there is a human being on both sides of every communication. Thank you for helping us make the package management world a bit safer.