Unauthenticated Password Reset Endpoint Lacks Rate Limiting, Enabling Email Flooding

Moderate **mjashanks** published **GHSA-277c-prw2-rqgh** yesterday

Package

Budibase App (SaaS) (Web Application / SaaS)

Affected versions	Patched versions
https://account.budibase.app/auth/forgot	Not yet patched

Description

Summary

A business logic vulnerability exists in Budibase's password reset functionality due to the absence of rate limiting, CAPTCHA, or abuse prevention mechanisms on the "Forgot Password" endpoint. An unauthenticated attacker can repeatedly trigger password reset requests for the same email address, resulting in hundreds of password reset emails being sent in a short time window. This enables large-scale email flooding, user harassment, denial of service (DoS) against user inboxes, and potential financial and reputational impact for Budibase.

Details

Budibase provides a "Forgot Password" feature that allows users to request a password reset link by submitting their email address. However, the password reset API endpoint does not enforce any form of abuse protection.

Specifically, the following protections are missing:

No rate limiting per IP address
No rate limiting per email / user
No time-based cooldown
No CAPTCHA or bot protection
No request throttling on repeated requests

Because of this, an attacker can automate password reset requests at scale using standard tools such as Burp Suite Intruder or other automated scripts. Each request successfully triggers a password reset email, even when sent repeatedly for the same email address from the same IP.

This is a business logic flaw affecting a security-sensitive endpoint and can be abused without authentication.

Proof of Concept (PoC)
Step 1: Account Setup

Navigate to https://account.budibase.app

Sign up for a new account (or use any existing account)

Log out

Step 2: Trigger Forgot Password

Go to the Forgot Password page:

https://account.budibase.app/auth/forgot

Enter a test email address (e.g. hasinotester@gmail.com)

Submit the request

Step 3: Capture the Request in Burp Suite

With Burp Suite running, capture the password reset request.
The request appears as follows:

PUT /api/v2/auth/password HTTP/1.1
Host: account.budibase.app
Content-Type: application/json
Origin: https://account.budibase.app
Referer: https://account.budibase.app/auth/forgot

{"email":"hasinotester@gmail.com"}

Step 4: Send the Request to Intruder
Right-click the request → Send to Intruder
In Intruder → Positions, clear all positions
Highlight only the email value and mark it as payload:

{"email":"§hasinotester@gmail.com§"}

Set attack type to Sniper

Step 5: Prepare Payloads

Create a text file containing the same email repeated multiple times 50 or more if you want, for example:

hasinotester@gmail.com
hasinotester@gmail.com
hasinotester@gmail.com
...

Load this file in Intruder → Payloads

Start the attack

Step 6: Observe Results

Each request returns a successful response (e.g. 201 Created)
No error, block, CAPTCHA, or cooldown is triggered
Hundreds of password reset emails are delivered to the target inbox

During testing, I successfully sent 300–500 password reset requests using Burp Suite Community Edition (which is intentionally slow). A real attacker using faster automation tools or Turbo Intruder could send significantly more requests in a much shorter time.

Impact

This vulnerability allows an unauthenticated attacker to:
Flood a user's inbox with hundreds of password reset emails
Harass or deny service to legitimate users
Abuse Budibase's email infrastructure at scale
Potentially cause financial impact due to transactional email costs
Risk domain reputation damage or email provider throttling
Perform large-scale automated abuse with minimal effort
Because the attack requires no authentication and no special privileges, it can be exploited by anyone on the internet.

Financial Impact (Cost of Abuse)

Budibase uses Amazon Simple Email Service (SES), which is a paid, pay-per-email service, to send transactional emails such as password reset messages.
Because the password reset endpoint lacks CAPTCHA, rate-limiting, and abuse protection, an attacker can continuously trigger password reset emails at scale, causing direct and ongoing financial loss to Budibase.

Amazon SES Cost Model (Relevant to This Issue)

$0.10 per 1,000 emails sent
$0.12 per GB of outbound data
Additional costs may apply for:

Dedicated IPs
Bounce and complaint handling
Reputation recovery if emails are flagged as spam

Abuse Scenario
An attacker can:
Automate password reset requests for a single or multiple email addresses
Send hundreds or thousands of emails per hour
Generate continuous SES charges with no limitation

Example:
100,000 automated reset emails = direct recurring cost
Sustained abuse over days or weeks = non-trivial financial impact
This cost is incurred even if the attacker does not control the victim's email inbox, making the attack low-effort and high-impact.

Severity Assessment

Severity: High
Priority: P2

Video PoC

I attempted to upload a video demonstrating the privilege escalation PoC, but the file size exceeds GitHub's 10 MB limit. Therefore, please check the video through this link:

(https://drive.google.com/file/d/13hOJBUe7-8rUeIFTrZUB6UFXzF46ePcJ/view?usp=sharing)

Responsible Disclosure

I will also report this vulnerability to Budibase via their security contact email:

[community@budibase.com]

Justification:
The issue affects a security-critical endpoint, is fully automatable, lacks all abuse protections, and was demonstrated at scale with hundreds of successful requests. This results in a real-world denial-of-service and abuse scenario rather than a theoretical issue.

OWASP Classification

OWASP Top 10 – A04:2021: Insecure Design
OWASP API Security Top 10 – API4:2023: Unrestricted Resource Consumption

Recommendations / Mitigation

Budibase should implement the following mitigations:
Enforce strict rate limiting on password reset requests (per IP and per user)
Introduce CAPTCHA or bot protection after repeated requests
Apply time-based cooldowns for password reset requests
Monitor and alert on abnormal password reset activity
Implement server-side abuse detection for security-sensitive endpoints

Reported by:
Ismail Hassan
Email: manhasino@gmail.com

Severity

Moderate 5.3 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	Low

Learn more about base metrics

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

CVE ID

CVE-2026-25043

Weaknesses

No CWEs

Credits

Hasinohacker Reporter