21

(Possible) staff account takeover via reset token bruteforce at helpdesk.bistudio.com

Share:

State	Resolved (Closed)
Disclosed	**September 19, 2018 8:12pm +0530**
Reported To	BOHEMIA INTERACTIVE a.s.
Asset	Domain listed in the policy scope (Other)
Weakness	Weak Password Recovery Mechanism for Forgotten Password
Bounty	$200
Severity	Critical (9 ~ 10)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE · EXPORT

europa submitted a report to **BOHEMIA INTERACTIVE a.s.**. Apr 3rd (10 months ago)

As stated in a brief exchange with @rvn in my other report ##312433, I might have found a logic flaw in the way https://helpdesk.bistudio.com handles the reset flow and tokens.
I've asked if it was possible to obtain a test account, but I fully understand that it's something that cannot be done; as such I'll submit a "blind" report based on my black-box analysis and wait for your team to verify it. Also note that this flaw seems to also be present in the "Set out of office email response" flow, albeit less critical.

Flow

The **SYSTEM PASSWORD RESET** flow is a 3-steps process:

1. the staff member requests a SMS TOKEN using the first form
2. the 6-digits SMS TOKEN is used in the second form
3. the staff member can now set a new SYSTEM PASSWORD in the third form

Analysis and logic

I was able to go through the process even after providing non-existing usernames and tokens by intercepting the **response** in BurpSuite and changing the status code from **400 Bad Request** to **200 OK** and the body from `"status":"error"` to `"status":"ok"`, allowing the AngularJS applet to follow through.
I then noticed that the API endpoint for verifying the SMS TOKEN and changing the password where open and free of rate-limiting measures, allowing for a quick bruteforce of the 000000-999999 space.
It should be therefore possible to perform an account takeover on any staff member, provided the SMS TOKEN really is a 6-digits code

Theoretical POC

1. adversary starts the SYSTEM PASSWORD RESET process for the target victim using a POST request to `/api/system/verification-codes` (ie: `{"username":"admin"}`). The backend generates a SMS TOKEN and sends it to the victim's phone. Meanwhile,
2. adversary obtains the **securityCode** value for the victim by bruteforcing `/api/system/verification-codes/[0-9]{6}` before the victim can cancel the flow (threat scenario places the attack durin night time)
3. adversary can now reset the SYSTEM PASSWORD by sending the complete POST request to `/api/system/email-account/password` (ie: `{"password":"<NEW PASSWORD>","code":"<BRUTEFORCED SMS TOKEN>","securityCode":"<RETRIEVED SECURITY CODE>"}`)

Step #1 offers a ReCAPTCHA anti-CSRF token but it's not used anywhere in the flow, making the attack possible

Step #2 is really a matter of resources. Being free of rate-limiting, the API endpoint will be quickly queried for all the possible token combinations in a matter of minutes using a multithreaded approach (ie: using BurpSuite's Intruder).

Albeit theoretical, the logic behind the threat scenario seems plausible. It might be worth investigating.

Recommended actions

Properly implement the ReCAPTCHA and a strict ratelimiting on the API endpoints

Impact

An adversary might be able to takeover staff accounts, or set their "out of office" email replies.

rvn posted a comment. Apr 4th (10 months ago)
Thanks for the report, we will check it out.

rvn changed the status to ○ **Triaged**. Apr 4th (10 months ago)
This has been confirmed as a legitimate issue.

BOHEMIA INTERACTIVE a.s. rewarded europa with a **$200** bounty. Apr 4th (10 months ago)
Bounty awarded. Thanks!

europa posted a comment. Apr 4th (10 months ago)
Glad I was able to blindly help!
Thanks for the reward, I remain available for further testing
Cheers!

rvn closed the report and changed the status to ○ **Resolved**. Apr 5th (10 months ago)
This should be fixed now. Thanks again!

europa posted a comment. Apr 5th (10 months ago)
First of all, thank you again for the highest reward!

I just tested and it's still "working":

- the ReCAPTCHA value isn't checked in the flow, the X-XSRF-TOKEN header can be omitted and the flow will still go through
- the `/api/system/email-account/password` API endpoint isn't rate-limited, allowing me to send as many requests as I want (I tried with 1,000) Has the fix been deployed in production already?

rvn posted a comment. Apr 5th (10 months ago)
Captcha isn't that necessary anymore as now you have a fix amount of tries before the account gets locked. Sending further requests is possible but will not have any effect. There is a side issue related to this that is being fixed right now but overall it should be resolved

europa posted a comment. Apr 5th (10 months ago)
Alrighty, should be good as long as it's not used to purposely lock staff accounts. Cheers!

europa requested to disclose this report. Sep 18th (5 months ago)
If you guys are okay with this one, I believe it could be disclosed as it highlights the honesty of the team and why hackers should work on your program regardless of its bounty table.
Needless to say, it is also why I sticked around. Great job!

freespirit agreed to disclose this report. Sep 19th (5 months ago)
Disclosed per reporter's request.

This report has been disclosed. Sep 19th (5 months ago)

rvn updated the severity from High to Critical. Sep 19th (5 months ago)