How I Could Have Hacked Any Instagram Account

Laxman Muthiyah

Security Researcher

May 27, 2021

Abstract- This paper is about how I found a vulnerability on Instagram that allowed me to hack any Instagram account with just mobile number using a race condition vulnerability in 6 digit forgot passcode validation endpoint. Facebook and Instagram security team fixed the issue and rewarded me $30000 as a part of their bounty program.

Index Terms- Instagram account takeover vulnerability, Bug Bounty Program, Race condition / hazard, Rate limit bypass, Brute force

I. Introduction

Facebook is working constantly to improve its security controls on all of their platforms. As a part of it, they recently increased bug bounty reward payouts for all critical vulnerabilities including account takeovers. Instagram is also a part of Facebook bug bounty program. So, I decided to try my luck on Facebook and Instagram. Fortunately, I was able to find one on Instagram.

II. Instagram password reset functionality

Instagram forgot password endpoint is the first thing that came to my mind while looking for an account takeover vulnerability. I tried to reset my password on the Instagram web interface. They have a link-based password reset mechanism which is pretty strong and I couldn't find any bugs after a few minutes of testing.

Then switched to their mobile recovery flow, where I was able to find a susceptible behavior. When a user enters his/her mobile number, they will be sent a six-digit passcode to their mobile number. They have to enter it to change their password. Therefore, if we are able to try all the one million codes on the verify-code endpoint, we would be able to change the password of any account. But I was pretty sure that there must be some rate limiting against such brute-force attacks. I decided to test it.

III. Rate limits and their bypasses

My tests did show the presence of rate limiting. I sent around 1000 requests, 250 of them went through and the rest 750 requests were rate limited. Tried another 1000, now many of them got rate limited. So, their systems are validating and rate limiting the requests properly.

Two things that struck mind was the number of requests and the absence of blacklisting. I was able to send requests continuously without getting blocked even though the number of requests I can send in a fraction of time is limited.

After a few days of continuous testing, I found two things that allowed me to bypass their rate limiting mechanism.

1. Race Hazard
2. IP rotation

For those who are unaware of race condition, please read it here. Sending concurrent requests using multiple IPs allowed me to send a large number of requests without getting limited. The number of requests we can send is dependent on concurrency of requests and the number of IPs we use. Also, I realized that the code expires in 10 minutes, it makes the attack even harder, therefore we need 1000s of IPs to perform the attack.

IV. Proof of concept

The following HTTP requests are sent by Instagram mobile app.

Requesting passcode: -

POST /api/v1/users/lookup/ HTTP/1.1
User-Agent: Instagram 92.0.0.11.114 Android (27/8.1.0; 440dpi; 1080×2150; Xiaomi/xiaomi; Redmi Note 6 Pro; tulip; qcom; en_IN; 152830654)
Accept-Language: en-IN, en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept-Encoding: gzip, deflate
Host: i.instagram.com
Connection: keep-alive

q=target_mobile_number&device_id=android-device-id-here

The target user will receive a passcode to their mobile number and it will expire in 10 minutes.

Verify passcode: -

POST /api/v1/accounts/account_recovery_code_verify/ HTTP/1.1
User-Agent: Instagram 92.0.0.11.114 Android (27/8.1.0; 440dpi; 1080×2150; Xiaomi/xiaomi; Redmi Note 6 Pro; tulip; qcom; en_IN; 152830654)
Accept-Language: en-IN, en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept-Encoding: gzip, deflate
Host: i.instagram.com
Connection: keep-alive

recover_code=123456&device_id=android-device-id-here

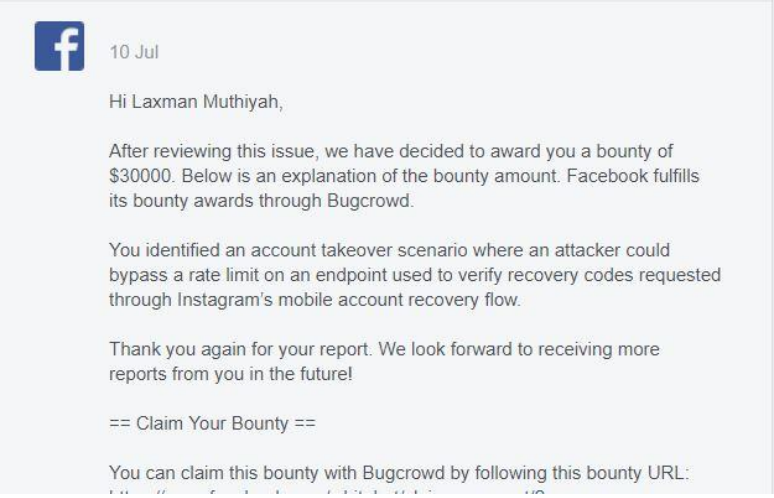
Now we need to brute-force this endpoint using multiple IPs. Roughly, I was able to send 200 requests from a single IP without hitting rate limit.

I have used 1000 different machines (to achieve concurrency easily) and IPs to send 200k requests (that's 20 percent of total one million probability) in my tests.

In a real attack scenario, the hacker needs 5000 IPs to hack an account. It sounds big but that's actually easy if you use a cloud service provider like Amazon or Google. It would cost around 150 dollars to perform the complete attack of one million codes.

I reported this vulnerability with clear reproduction steps to Facebook security team along with a video of sending 200k valid requests. They were quick in addressing and fixing the issue.

After the patch, Facebook rewarded me $30,000 for responsible disclosing this vulnerability as a part of their bug bounty program.



10 Jul

Hi Laxman Muthiyah,

After reviewing this issue, we have decided to award you a bounty of $30000. Below is an explanation of the bounty amount. Facebook fulfills its bounty awards through Bugcrowd.

You identified an account takeover scenario where an attacker could bypass a rate limit on an endpoint used to verify recovery codes requested through Instagram's mobile account recovery flow.

Thank you again for your report. We look forward to receiving more reports from you in the future!

== Claim Your Bounty ==

You can claim this bounty with Bugcrowd by following this bounty URL:

REFERENCES

[1] Paleari, Roberto, et al. "On race vulnerabilities in web applications." International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Berlin, Heidelberg, 2008.

[2] Petrov, Boris, et al. "Race detection for web applications." ACM SIGPLAN Notices 47.6 (2012): 251-262.

[3] Hubbard, John, Ken Weimer, and Yu Chen. "A study of SSL proxy attacks on Android and iOS mobile applications." 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC). IEEE, 2014.