How I Might Have Hacked Any Microsoft Account

Laxman Muthiyah

Security Researcher

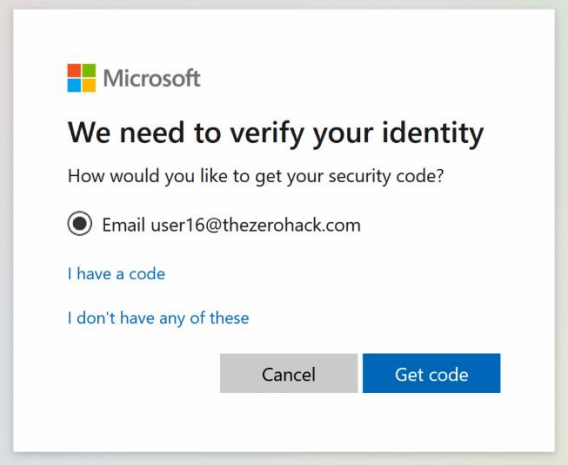
May 27, 2021

Abstract- This paper is about how I found a vulnerability on Microsoft online services that allowed me to hack any Microsoft account using a race condition vulnerability in forgot passcode validation endpoint. Microsoft security team fixed the issue and rewarded me $50000 as a part of their identity bounty program.

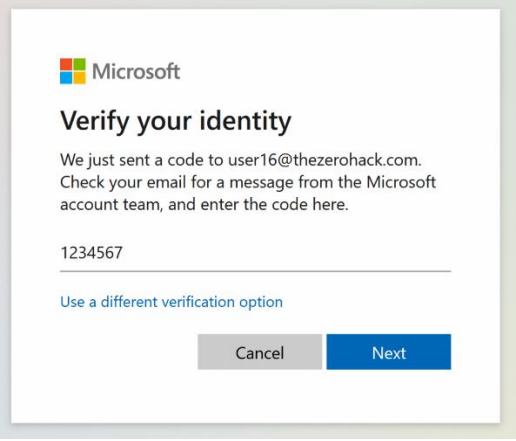
Index Terms- Microsoft account takeover vulnerability, Bug Bounty Program, Race condition / hazard, Rate limit bypass, Brute forcing

I. INTRODUCTION

Microsoft password reset mechanism is using a 7 digit passcode to reset a user's password. To reset a Microsoft account's password, we need to enter our email address or phone number in their forgot password page, after that we will be asked to select the email or mobile number that can be used to receive security code.

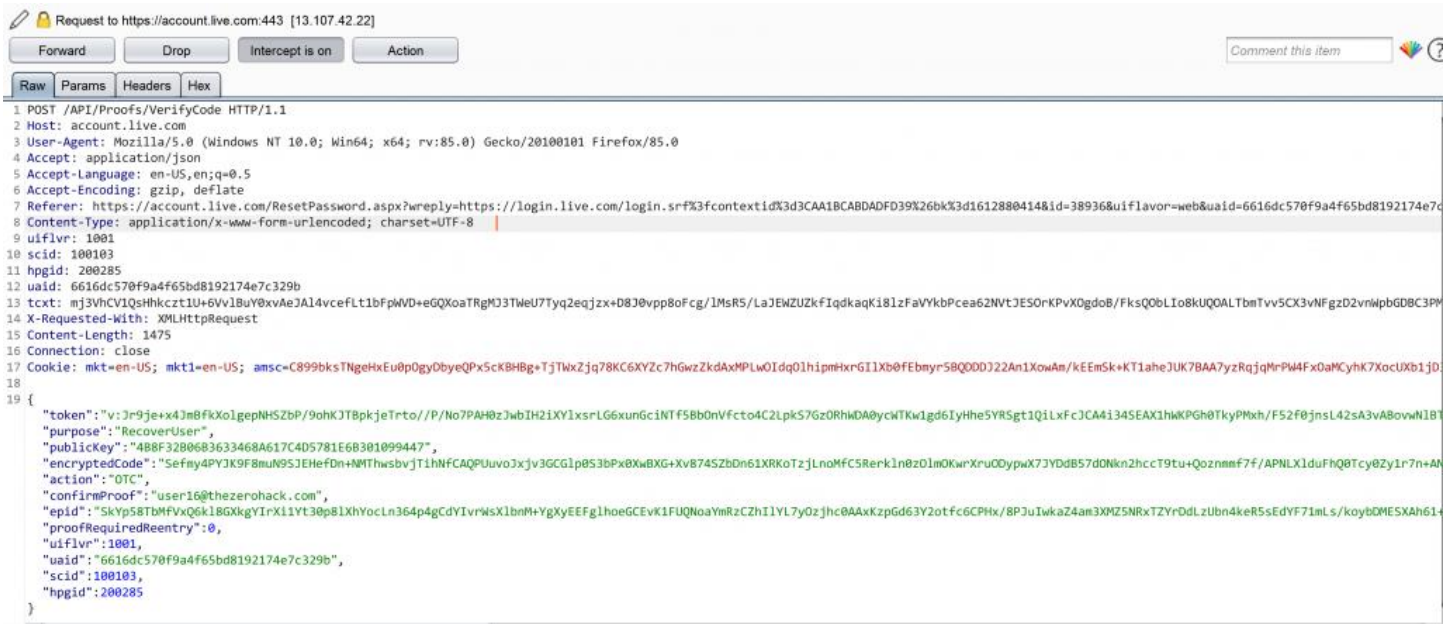


Once we receive the 7-digit security code, we will have to enter it to reset the password. Here, if we can brute force all the combination of 7-digit code (that will be 10^7 = 10 million codes), we will be able to reset any user's password without permission. But obviously, there will be some rate limits that will prevent us from making large number of attempts.



II. CLIENT-SIDE ENCRYPTION

Intercepting the HTTP POST request made to code validation endpoint looked like this



If you look at the screenshot above, the code 1234567 we entered was nowhere present in the request. It was encrypted and then sent for validation. I guess they are doing this to prevent automated brute force tools from exploiting their system. So, we cannot automate testing multiple codes using tools like Burp Intruder since they won't do the encryption part.

After some time, I figured out the encryption technique and was able to automate the entire process from encrypting the code to sending multiple concurrent requests.

III. RATE LIMITS AND THEIR BYPASSES

My initial test showed the presence of rate limits as expected. Out of 1000 codes sent, only 122 of them got through, others are limited with 1211 error code and they are blocking the respective user account from sending further attempts if we continuously send requests.

Then, I tried sending simultaneous / concurrent requests like I did for Instagram, that allowed me to send large number of requests without getting blocked but I was still unable to get the successful response while injecting the correct 7-digit security code. I thought

they have some controls in place to prevent this type of attack. Although I am getting an error while sending the right code, there was still no evidence of blocking the user like we saw in the initial test. So, I was still hoping that there would be something.

After some days, I realized that they are blacklisting the IP address if all the requests we send don't hit the server at the same time, even a few milliseconds delay between the requests allowed the server to detect the attack and block it. Then I tweaked my code to handle this scenario and tested it again.

Surprisingly, it worked and I was able to get the successful response this time!

I sent around 1000 seven-digit codes including the right one and was able to get the next step to change the password.

The above process is valid only for those who do not have two factor authentication enabled, if a user has enabled 2FA, we will have to bypass two factor code authentication as well, to change the password.

I tested an account with 2FA and found both are same endpoint that are vulnerable to this type of attack. At first, user will be prompted to enter a 6-digit code generated by authenticator app, only then they will be asked to enter 7-digit code sent to their email or phone number. Then, they can change the password.

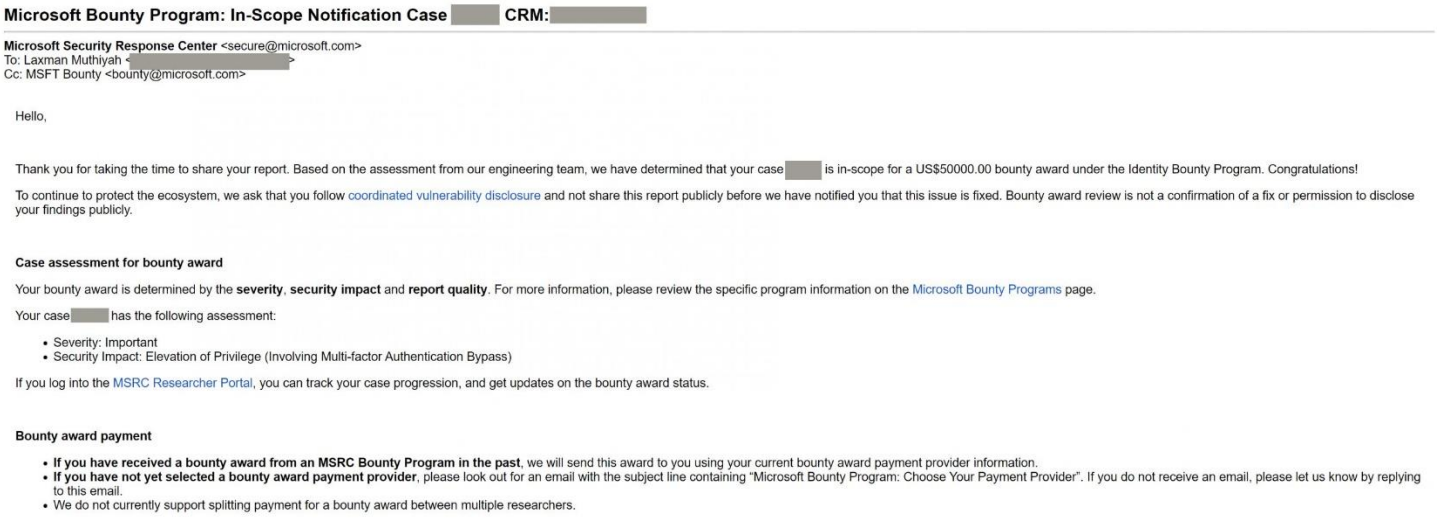
Putting all together, an attacker has to send all the possibilities of 6 and 7 digit security codes that would be around 11 million request attempts and it has to be sent concurrently to change the password of any Microsoft account (including those with 2FA enabled).

It is not at all a easy process to send such large number of concurrent requests, that would require a lot of computing resources as well as 1000s of IP address to complete the attack successfully.

Immediately, I recorded a video of all the bypasses and submitted it to Microsoft along with detailed steps to reproduce the vulnerability. They were quick in acknowledging the issue.

The issue was patched in November 2020 and my case was assigned to different security impact than the one expected. I asked them to reconsider the security impact explaining my attack. After a few back-and-forth emails, my case was assigned to Elevation of Privilege (Involving Multi-factor Authentication Bypass). Due to the complexity of the attack, bug severity was assigned as important instead of critical.

After the patch, MSRC rewarded me $50,000 for responsible disclosing this vulnerability as a part of their identity bounty program.



Microsoft Bounty Program: In-Scope Notification Case [redacted] CRM: [redacted]

Microsoft Security Response Center <secure@microsoft.com>
To: Laxman Muthiyah <[redacted]>
Cc: MSFT Bounty <bounty@microsoft.com>

Hello,

Thank you for taking the time to share your report. Based on the assessment from our engineering team, we have determined that your case [redacted] is in-scope for a US$50000.00 bounty award under the Identity Bounty Program. Congratulations!

To continue to protect the ecosystem, we ask that you follow coordinated vulnerability disclosure and not share this report publicly before we have notified you that this issue is fixed. Bounty award review is not a confirmation of a fix or permission to disclose your findings publicly.

Case assessment for bounty award

Your bounty award is determined by the **severity**, **security impact** and **report quality**. For more information, please review the specific program information on the Microsoft Bounty Programs page.

Your case [redacted] has the following assessment:

- Severity: Important
- Security Impact: Elevation of Privilege (Involving Multi-factor Authentication Bypass)

If you log into the MSRC Researcher Portal, you can track your case progression, and get updates on the bounty award status.

Bounty award payment

- **If you have received a bounty award from an MSRC Bounty Program in the past**, we will send this award to you using your current bounty award payment provider information.
- **If you have not yet selected a bounty award payment provider**, please look out for an email with the subject line containing "Microsoft Bounty Program: Choose Your Payment Provider". If you do not receive an email, please let us know by replying to this email.
- We do not currently support splitting payment for a bounty award between multiple researchers.

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] Reese, Ken, et al. "A usability study of five two-factor authentication methods." Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019). 2019

[4] Detering, Dennis, et al. "On the (in-) security of JavaScript Object Signing and Encryption." Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium. 2017.