Chaining Multiple Requests to Achieve Rate Limiting Vulnerabilities

Ahmad Halabi Nov 30, 2020 · 4 min read

Hello,

I want to share with you a new methodology about finding rate limit vulnerabilities and even bypassing rate limit protections.

For those who don’t know me, my name is Ahmad Halabi and I am a part time bug bounty hunter.

Overview ::

A lot of programs and companies implement Rate Limiting protections on sensitive endpoints that requires authentication and important functionalities like Login and creating posts as an example. Protections can vary a lot, and since there are multiple types and ways how protections are implemented, there are also methods to bypass some of these protections.

Chaining Multiple Requests to achieve Rate Limiting vulnerability which was Sending Unlimited Collaboration Invites

The program generally contains Algorithm section in the page where you can add collaborator to work with you on your project.
When you add a collaborator, a notification is sent to his email telling him to join you as a collaborator.
You can invite a user just once, unless you removed him from collaborator and re-invited him.

Through the above feature, I found a bug by chaining three requests `add_collaborator` , `normal request` and `remove_collaborator` I was able to create a thing in burp called Macro that lead me send notifications to a target user unlimited times by performing the below steps:

- Perform add collaborator request.
- Send normal request to www.target.com.
- Perform remove collaborator request.

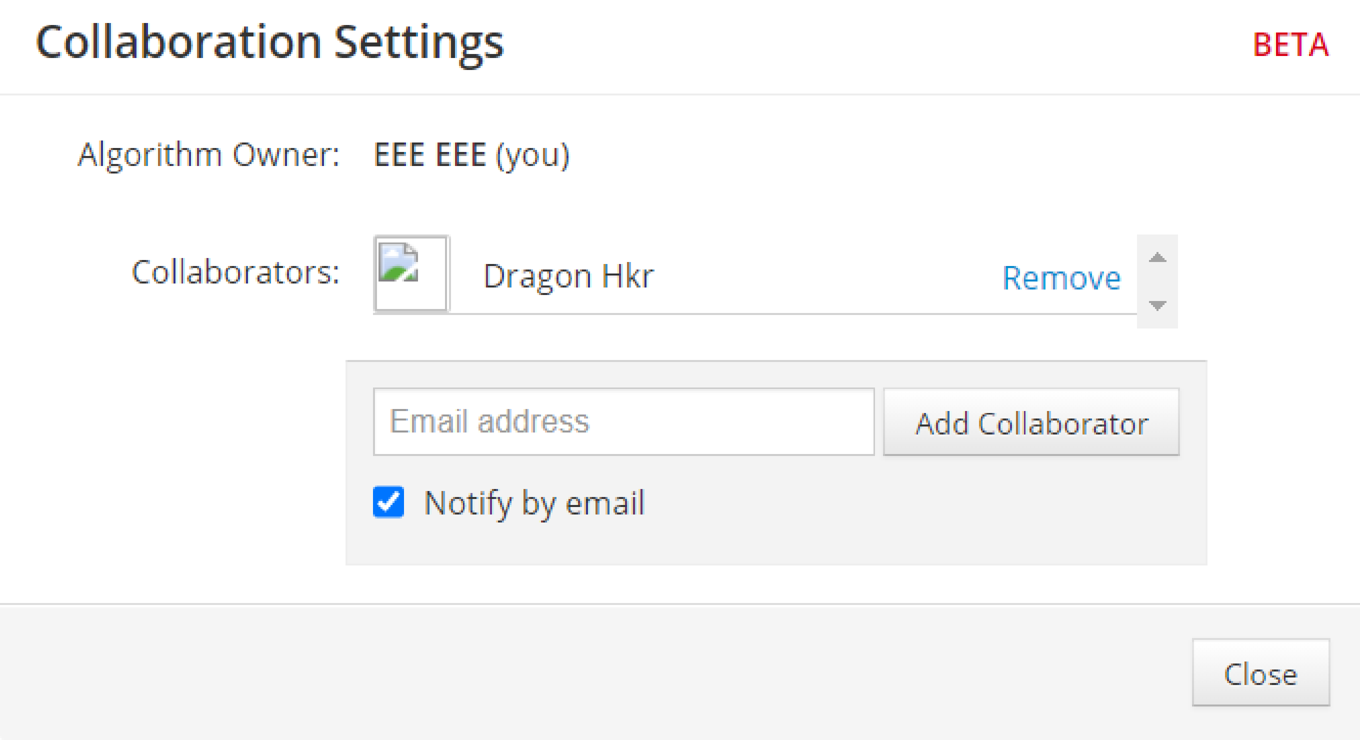
And repeating the above three steps in an automated way will result in bombing victim's mailing system with collaboration invites.

Steps To Reproduce ::

To prove the existence of the bug in the basic way, all you need is to add a collaborator, remove it, and then add it again and remove it. You will receive two notifications to that

If you want to do it in advanced way, you can create a script that automate the add, remove collaborator process and repeats itself every time. Or you can use burp as well. I used burp Macro feature.

- Keep burp proxy running and Perform add collaborator and remove collaborator and navigate to your account.



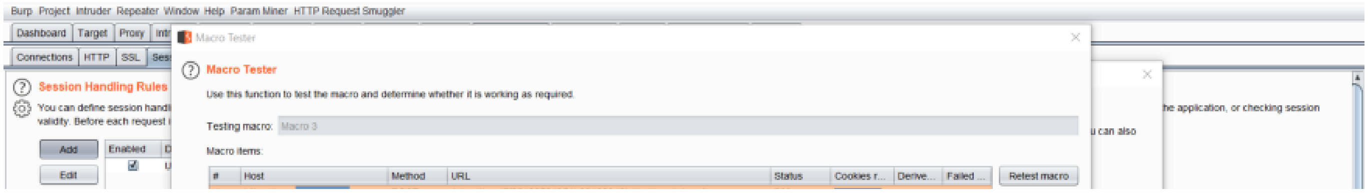
Collaboration Settings

Algorithm Owner: EEE EEE (you)

Collaborators: Dragon Hkr

Collaboration Settings

- In burp, navigate to Project Options -> Under Session Handling Rules click Add -> In Rule Actions click Add then choose Run a Macro.
- Under Select Macro click Add -> Burp requests history will open, now choose the three requests in order: Add collaborator — Request to profile account — Remove collaborator. Click Ok and then click Test macro and see that a notification is sent to the target email inbox.



Session Handling Rules

You can define session handl… validity. Before each request i…

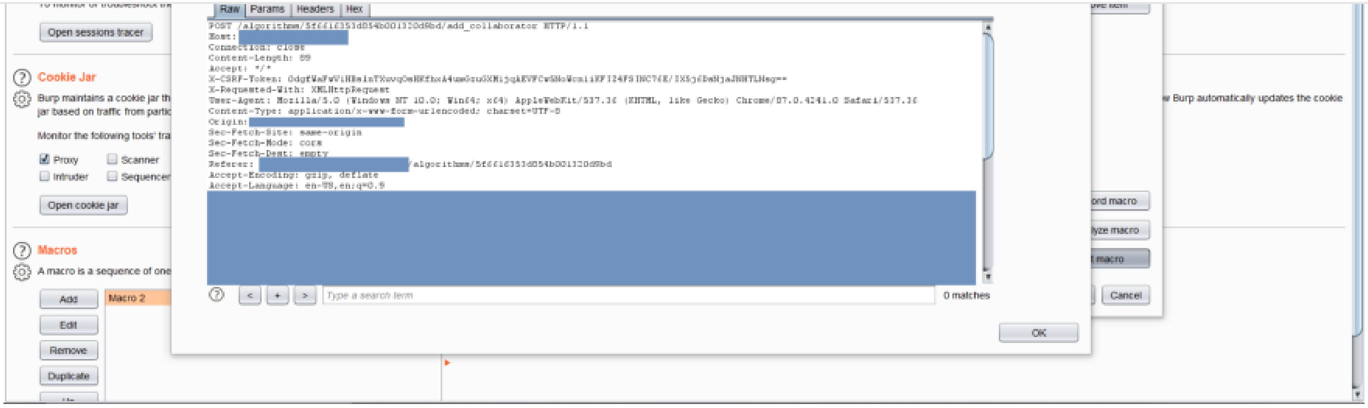
Macro Tester

Use this function to test the macro and determine whether it is working as required.

Testing macro: Macro 3

Macro items:

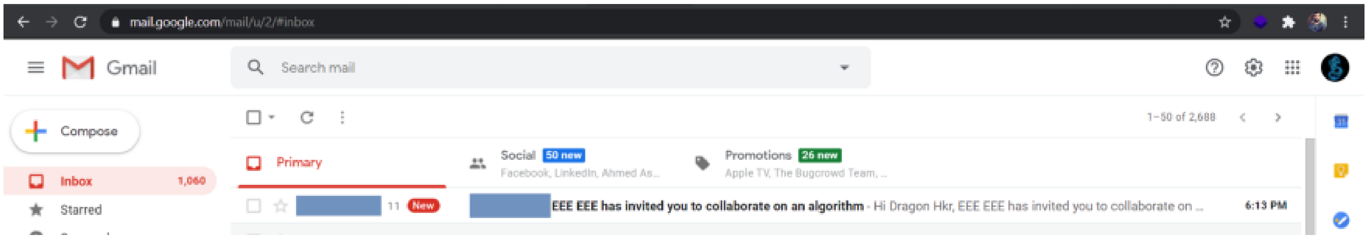
Burp Project Intruder Repeater Window Help Param Miner HTTP Request Smuggler
Dashboard Target Proxy Intr
Connections HTTP SSL Ses
Add
Edit
Enabled
# Host Method URL Status Cookies r… Derive… Failed …
Retest macro



```
POST /algorithms/5f6616353d054b001320d9bd/add_collaborator HTTP/1.1
Host: [redacted]
Connection: close
Content-Length: 59
Accept: */*
X-CSRF-Token: OdgfWaFwViHBwinTXuvqOmHKfhxA4umGzuGXHijqkEVFCwSNoWcmiiRF124FSINC7dE/IX5j6DmNjaJNHTLHeg==
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4241.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: [redacted]
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]/algorithms/5f6616353d054b001320d9bd
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
[redacted]
```

Launching Macro Attack

- You can try `Test macro` many times and every time you try it, a notification will be sent to the mail inbox.



EEE EEE has invited you to collaborate on an algorithm - Hi Dragon Hkr, EEE EEE has invited you to collaborate on ... 6:13 PM

Notifications sent to victim inbox

Automating this attack will lead to huge mass mailing victim inbox.

Impact ::

Abuse sending collaboration invites to any user, large scaled attack will blow up users mailing system with huge number of invites.

Why this Vulnerability Arises ::

Simply because the program didn't expect chaining three requests to achieve rate limiting on collaboration invites.

Applying rate limit protection for the Add Collaborator and Remove Collaborator requests.

Lesson Learned from this bug ::

If you depend on multiple requests to perform your certain action, don't just rely on them as a protection, you should also implement sort of rate limiting protection because multiple requests can be chained to exploit rate limiting.

Report Timeline ::

19 Sep, 2020 : Initial Report.

2 Nov, 2020 : Report Triaged.

24 Nov, 2020: Bounty Awarded ($1,000).

24 Nov, 2020: Report Resolved.

For those who didn't read my article yet about how I started bug bounty hunting, how I ranked 1st at U.S. Dept Of Defense (2019) and how I reached top 100 hackers on hackerone, You can find it below.

My Bug Bounty Journey & Ranking 1st in U.S. DoD & Achieving top 100 hackers in 1 year

I am sharing some of my methodology, recourses, tips and advices to become a better bug bounty hunter.

ahmdhalabi.medium.com

Good luck :)

Ahmad Halabi

Security Researcher

ahmadhalabi.net

Bug Bounty Hunting Security Penetration Testing Hackerone Bug Bounty Tips