60 MIN

2FA/OTP Bypass

DAY ##

Bypassing Two-factor Authentication

DIRECT BYPASS

- To bypass 2FA, just **try to access the next endpoint directly** (you need to know the path of the next endpoint). If this doesn't work, try to change the **Referrer header** as if you came from the 2FA page.

REUSING THE TOKEN

- Maybe you can reuse a previously used token inside the account to authenticate.

SHARING UNUSED TOKEN

- Check if you can get the token from your account and try to use it to bypass the 2FA in a different account.

LEAKED TOKEN

- Is the token leaked on a response from the web application?

PASSWORD RESET FUNCTION

- Using In almost all web applications the **password reset function automatically logs the user into the application** after the reset procedure is completed. Check if a **mail **is sent with a **link **to **reset the password** and if you can **reuse** that **link **to reset the password as **many times as you want** (even if the victim changes his email address).

OAUTH

- If you can compromise the account of the user in a trusted **OAuth** platform (Google, Facebook...)

Brute Force

Lack of rate limit

- Is there any limit on the number of codes that you can try, so you can just brute force it? Be careful with a possible "silent" rate limit, always try several codes and then the real one to confirm the vulnerability.

Brute Force

FLOW RATE BUT NO RATE LIMIT

- In this case, there is a flow rate limit (you have to brute force it very slowly: 1 thread and some sleep before 2 tries) but no rate limit. So with enough time, you can be able to find the valid code.

Brute Force

RE-SEND CODE AND RESET THE LIMIT

- There is a rate limit but when you "resend the code" the same code is sent and the rate limit is reset. Then, you can brute force the code while you resend it so the rate limit is never reached.

Rate Limit Bypass (In-Depth)

USING SIMILAR ENDPOINTS

- If you are attacking the /api/v3/sign-up endpoint try to perform bruteforce to /Sing-up, /SignUp, /singup...
- Also try appending to the original endpoint bytes like %00, %0d%0a, %0d, %0a, %09, %0C, %20

Rate Limit Bypass (In-Depth)

BLANK CHARS IN CODE/PARAMS

- Try adding some blank byte like %00, %0d%0a, %0d, %0a, %09, %0C, %20 to the code and/or params. For example code=1234%0a or if you are requesting a code for an email and you only have 5 tries, use the 5 tries for example@email.com, then for example@email.com%0a, then for example@email.com%0a%0a, and continue...

Rate Limit Bypass (In-Depth)

CHANGING IP ORIGIN USING HEADERS

- If they are limiting to 10 tries per IP, every 10 tries change the IP inside the header.

```
X-Originating-IP: 127.0.0.1
X-Forwarded-For: 127.0.0.1
X-Remote-IP: 127.0.0.1
X-Remote-Addr: 127.0.0.1
X-Client-IP: 127.0.0.1
X-Host: 127.0.0.1
X-Forwared-Host: 127.0.0.1


#or use double X-Forwared-For header
X-Forwarded-For:
X-Forwarded-For: 127.0.0.1
```

Rate Limit Bypass (In-Depth)

CHANGE OTHER HEADERS

- Try changing the user-agent, the cookies... anything that could be able to identify you.

Rate Limit Bypass (In-Depth)

ADDING EXTRA PARAMS TO THE PATHS

- Try If the limit in in the path /resetpwd, try BFing that path, and once the rate limit is reached try /resetpwd?someparam=1

Rate Limit Bypass (In-Depth)

LOGIN TO YOUR ACCOUNT BEFORE EACH ATTEMPT

- Maybe if you **login into your account before each attempt** (or each set of X tries), the rate limit is restarted. If you are attacking a login functionality, you can do this in burp using a Pitchfork attack in **setting your credentials every X tries** (and marking follow redirects).

Brute Force

Lack of rate limit in the user's account

- Sometimes you can configure the 2FA for some actions inside your account (change mail, password...). However, even in cases where there is a rate limit when you tried to log in, there isn't any rate limit to protect actions inside the account.

Brute Force

LACK OF RATE LIMIT
RE-SENDING THE CODE VIA SMS

- You won't be able to bypass the 2FA but you will be able to waste the company's money.

Brute Force

INFINITE OTP REGENERATION

- If you can generate a new OTP infinite times, the** OTP is simple enough** (4 numbers), and you can try up to 4 or 5 tokens per generated OTP, you can just try the same 4 or 5 tokens every time and generate OTPs until it matches the ones you are using.

CSRF/CLICKJACKING

- Check if there is a Cross Site Request Forgery (CSRF) or a Clickjacking vulnerability to disable the 2FA.

Remember Me Functionality

GUESSABLE COOKIES

- If the "remember me" functionality uses a new cookie with a guessable code, try to guess it.

Remember Me Functionality

IP ADDRESS

- If the "remember me" functionality is attached to your IP address, you can try to figure out the IP address of the victim and impersonate it using the **X-Forwarded-For** header.

Older Versions

SUBDOMAINS

- If you can find some "testing" subdomains with the login functionality, they could be using old versions that don't support 2FA (so it is directly bypassed) or those endpoints could support a vulnerable version of the 2FA.

Older Versions

APIs

- If you find that the 2FA is using an API located under a /v*/ directory (like "/v3/"), this probably means that there are older API endpoints that could be vulnerable to some kind of 2FA bypass.

PREVIOUS VERSIONS

- If When the 2FA is enabled, previous sessions created should be ended. This is because when a client has his account compromised he could want to protect it by activating the 2FA, but if the previous sessions aren't ended, this won't protect him.

IMPROPER ACCESS CONTROL TO BACKUP CODES

- Backup codes are generated immediately after 2FA is enabled and are available on a single request. After each subsequent call to the request, the codes can be regenerated or remain unchanged (static codes). If there are CORS misconfigurations/XSS vulnerabilities and other bugs that allow you to “pull” backup codes from the response request of the backup code endpoint, then the attacker could steal the codes and bypass 2FA if the username and password are known.

INFORMATION DISCLOSURE

- If you notice some confidential information appear on the 2FA page that you didn't know previously (like the phone number), then this can be considered an information disclosure vulnerability.

PASSWORD-RESET == 2FA

Create an Account and Turn On 2FA.

- Logout from that account.
- Now, Go to forget Password-Reset page.
- Change your password.
- Now try to log in.
- If you are not asked to enter a 2FA code, You can report.

REFERENCES

- https://medium.com/@iSecMax/two-factor-authentication-security-testing-and-possible-bypasses-f65650412b35
- https://azwi.medium.com/2-factor-authentication-bypass-3b2bbd907718