[redacted]

2FA Bypass Techniques

Clickjacking on 2FA Disable Feature

1. Try to Iframe the page where the application allows a user to disable 2FA
2. If Iframe is successful, try to perform a social engineering attack to manipulate victim to fall in your trap.

Response Manipulation

1. Check Response of the 2FA Request.
2. If you Observe "Success":false
3. Change this to "Success":true and see if it bypass the 2FA

You can also use Burp Match & Replace Rules for this.

Status Code Manipulation

1. If the Response Status Code is 4XX like 401, 402, etc.
2. Change the Response Status Code to "200 OK" and see if it bypass the 2FA

2FA Code Reusability

1. Request a 2FA code and use it.
2. Now, Re-use the 2FA code and if it is used successfully that's an issue.
3. Also, try requesting multiple 2FA codes and see if previously requested Codes expire or not when a new code is requested.
4. Also, try to re-use the previously used code after long time duration say 1 day or more. That will be an potential issue as 1 day is enough duration to crack and guess a 6-digit 2FA code.

CSRF on 2FA Disable Feature

1. Navigate to 2FA Page and Click on Disable and capture this request with Burp Suite & Generate a CSRF PoC
2. Send this PoC to the victim user and check if CSRF happens successfully and removes the 2FA from victim account.
3. Also check if there is any authentication confirmation such as password or 2FA code required before disabling 2FA

Backup Code Abuse

Apply same techniques used on 2FA such as Response/Status Code Manipulation, Brute-force, etc. to bypass Backup Codes and disable/reset 2FA

Enabling 2FA Doesn't Expire Previous Session

1. Login to the application in two different browsers and enable 2FA from 1st session.
2. Use 2nd session and if it is not expired, it could be an issue if there is an insufficient session expiration issue. In this scenario if an attacker hijacks an active session before 2FA, it is possible to carry out all functions without a need for 2FA

2FA Code Leakage in Response

1. At 2FA Code Triggering Request, such as Send OTP functionality, capture the Request.
2. See the Response of this request and analyze if the 2FA Code is leaked.

JS File Analysis

While triggering the 2FA Code Request, Analyze all the JS Files that are referred in the Response to see if any JS file contain information that can help bypass 2FA code.

Lack of Brute-Force Protection

This involves all sort of issues which comes under security misconfiguration such as lack of rate limit, no brute-force protection, etc.

1. Request 2FA code and capture this request.
2. Repeat this request for 100-200 times and if there is no limitation set, that's a rate limit issue.
3. At 2FA Code Verification page, try to brute-force for valid 2FA and see if there is any success.
4. You can also try to initiate, requesting OTPs at one side and brute-forcing at another side. Somewhere the OTP will match in middle and may give you a quick result.

Password Reset/Email Change - 2FA Disable

1. Assuming that you are able to perform email change or password reset for the victim user or make victim user do it by any means possible.
2. 2FA is disabled after the email is changed or password is reset. This could be an issue for some organizations. However, depends on case by case basis.

Missing 2FA Code Integrity Validation

1. Request a 2FA code from Attacker Account.
2. Use this valid 2FA code in the victim 2FA Request and see if it bypass the 2FA Protection.

Direct Request

1. Directly Navigate to the page which comes after 2FA or any other authenticated page of the application.
2. See if this bypasses the 2FA restrictions.

2FA Refer Check Bypass

1. Directly Navigate to the page which comes after 2FA or any other authenticated page of the application.
2. If there is no success, change the refer header to the 2FA page URL. This may fool application to pretend as if the request came after satisfying 2FA Condition.