1. What vulnerability did you exploit? How does it work?

I exploited the csrf vulnerability of the website. I made an html file called csrf\_nsfw that executes a img that loads a script to a link http://3.80.200.34/xvwa/vulnerabilities/csrf/?passwd=freemoney&confirm=freemoney&submit=submit. The image doesnt actually load into anything so the victim wouldnt know what actually happens.

This js link call by use of image will load that exact link and will fill in password as freemoney and confirm as freemoney and also submit to model how an actual person who decided to change password manually will do. But because the victim is logged in and the page does not redirect to the actual page, the victim will unknowingly get his password changed by the attacker.

2. Is your application subject to this particular vulnerability? Why/why not?

Our application parkazes is a mobile app and is not subject to link redirect attacks that fills in password values on click. Even if it was form request attack which I tried but for some reason would not actually fill in the form for the xvwa website, our app is not vulnerable to request of this type as since webform attacks are different in the csrf variant in mobile apps. Thus our app parkaze is not vulnerable to the attack.

3. How will you protect against this vulnerability in your application?

We do not handle request forgery attacks via passwords in this application. That is already built upon by the parkaze group prior to us entering the group. It is a mobile app and we handle front end and backend for new/upgraded functionality of

them app.

My public ip:

2601:182:cd01:9a83:d1a2:5fb5:876f:ce8c

2nd time for the screen shots:

2601:182:cd01:9a83:e5b0:fbfe:fc50:90b6