**Introduction**

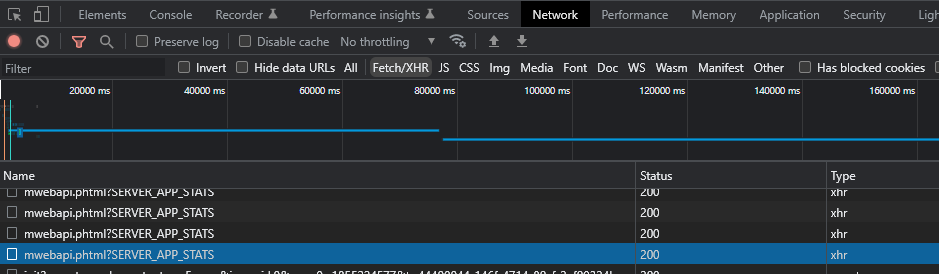
Bumble is a popular dating app that allows users to connect with potential matches. The app was launched in 2014 and is available for both iOS and Android devices. Unlike other dating apps, Bumble requires women to make the first move, giving them a greater sense of control over their dating experience. Bumble also offers a unique feature called Bumble BFF, which allows users to find friends rather than romantic partners. In addition, the app also includes Bumble Bizz, which is a networking and career-oriented feature. Bumble has become a popular choice for those looking to connect with new people, whether it be for dating, friendship, or professional networking.

A few months ago, as I navigated the dating landscape, I found myself searching for a solution to a common conundrum: how to discover which of my potential matches had expressed interest in me. Like many dating apps on the market today, Bumble presented me with the option to purchase a subscription, which would then allow me access to the information of users who had liked my profile.

**The Research**My main goal while researching is to discover which of my potential matches has expressed interest in me without having to pay money. I wanted to understand how the client knows while I’m swipe through my options who liked me and how I can find out before if I have match or no.

**Bumble site**When I set out to conduct my research on the Bumble app, I spent some time exploring Bumble's website, learning about the website features and how it operates.  
After setting up the profile on the site, I waited for likes. Right after I received a few individual likes, I went to the page that should show me who liked me, but unfortunately after a short check I found that the images I see are blurry.

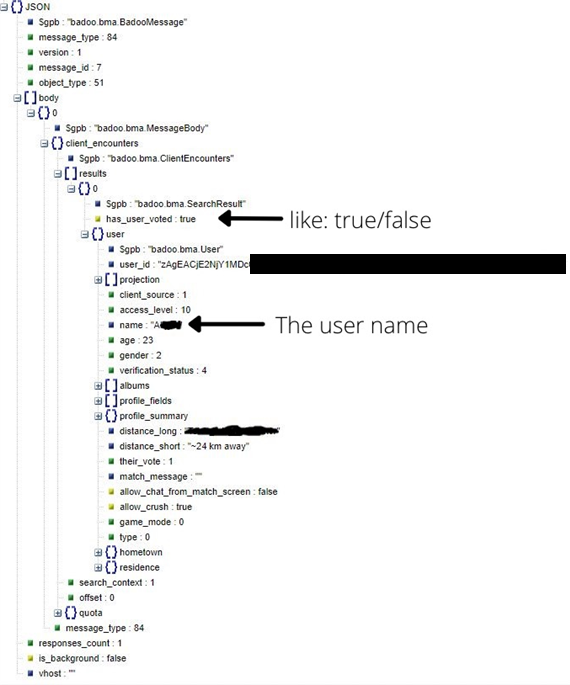
**Browsers Developer Tools**Next, I focused on understanding the configuration of the site and how the client receives the information. I used the “network” tab to see what information was being loaded. This is how I discovered that the server sends the information using “phtml” files.



**The Data**  
Then, because there were a lot of "phtml" files being loaded, I decided to look for the largest file. When I found the file started diving into its information.

After I copied the information that the client received from the server in a file ending with "**mwebapi.phtml?SERVER\_GET\_ENCOUNTERS**" to JSON Viewer to see the information more conveniently. I discovered that this is how the client loads all the information about the users that are displayed.

While analyzing the information, I found that the data is not encrypted, and each user has a key called "**has\_user\_voted**" which is a Boolean.  
This is how the data is presented:



Now I had a pretty good idea of how the server was sending the data to the client and how the client knows if someone "like" me. I was able to use this information to create Python script using Selenium that allowed me to see who liked me before swiping.

**Conclusion**One of the key vulnerabilities I took advantage of in this project is the fact that the information that should provide whether the user has liked or not comes together with the rest of the information about the user. In addition, the server sends unencrypted information. This means you don't have to pay to see who liked you and you can see before if someone liked or not.

**Disclaimer**It is important to note that this is a proof of concept and **I do not** take any responsibility for any damage caused using this script, or for security issues or bans.