A few of my findings

From applying my methodology for years I've managed to find quite a few interesting finds with huge impact. Sadly I can't disclose information about all of the companies I have worked with but I can tell you bug information and how I went about finding these bugs to hopefully give you an idea of how I apply my methodology. One thing to note is that I don't just test on private programs.

- 30+ open redirects found, leaking a users' unique token. Broke patch multiple times

I found that the site in question wasn't filtering their redirects so I found lots of open url redirects from just simple dorking. From discovering so many so quickly I instantly thought.. "This is going to be fun!". I checked how the login flow worked normally & noticed auth tokens being exchanged via a redirect. I tested and noticed they whitelisted *.theirdomain.com so armed with lots of open url redirects I tested redirecting to my website. I managed to leak the auth token **but upon the first test I couldn't work out how to actually use it.** A quick google for the parameter name and I found a wiki page on their developer subdomain which detailed the token is used in a header for API calls. The PoC I created proved I could easily grab a users' token after they login with my URL and then view their personal information via API calls. **The company fixed the open url redirect**, but didn't change the login flow. I managed to make this bug work multiple times from multiple open redirects before they made significant changes.

- **Stored XSS via their mobile app on heavily tested program**

I mentioned this briefly earlier. This was on a heavily tested public bug bounty program that has thousands of resolved reports. I simply installed their mobile app and the *very first* request made generated a GDPR page which asked me to consent to cookies. Upon re-opening the application the request was **not** made again. I noticed in this request I could control the "returnurl" parameter which allowed me to