Something Awesome: CTF

Rohan Maloney

My Idea

For my 'Something Awesome' project I set out to build my own CTF (capture the flag) exercise. I wanted to incorporate a wide range of exploits, including the techniques we had learned in lectures as well as other techniques that I would research myself. I originally planned a set list of exploits and their flags, including basic SQLi and XSS to second-order SQLi and DOM-based XSS. Yet after starting to write my CTF I discovered many other types of vulnerabilities that stemmed from XSS and SQLi.

What I achieved

In my project I managed to implement a CTF with 9 flags which are obtainable through various techniques of varying complexity. Unfortunately, I couldn't work out how to deliver the 2 XSS based flags (as I need another user to 'visit' the vulnerable page), but all other flags become available once the user has demonstrated the exploit. The most important thing I achieved, was that I learned about many common exploits and how to secure against them. Some examples are: I discovered how vulnerable deserialising data can be especially in the examples of YAML or pickle in python[1]; I found that where there's an XSS vulnerability there may also be a Server-Side Template Injection (SSTI) vulnerability, and how this can enable attackers to execute dangerous code such as opening pipes[2]; I also discovered how easy it is to crack MD5 hashes of some passwords with one google search[3]. I made flags for each of these (and 6 more exploits) yet there were many more different kinds of exploits and vulnerabilities that I discovered in my research for this project.

Reflections

In doing this project I found myself focussing more on making the website secure (from unintended methods of obtaining flags) than I ever thought I would. I implemented this project in python with flask as I am already familiar with this framework – I didn't want to spend my time learning another language/framework. I believe that this was the right choice, but I also believe that with ant new framework I learn, it would be incredibly beneficial to do a similar project as it really helped me understand the security issues common in that framework – many of which I was guilty of.

1. https://xerosecurity.com/wordpress/exploiting-python-deserialization-vulnerabilities/
2. https://medium.com/@nyomanpradipta120/ssti-in-flask-jinja2-20b068fdaeee
3. https://crackstation.net/