Based on the vulnerable app you submitted for lab 2a, try and make it secure. Please submit the secured app and a short summary of the measures you took to secure it.

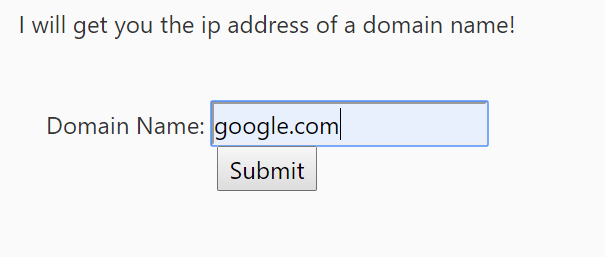
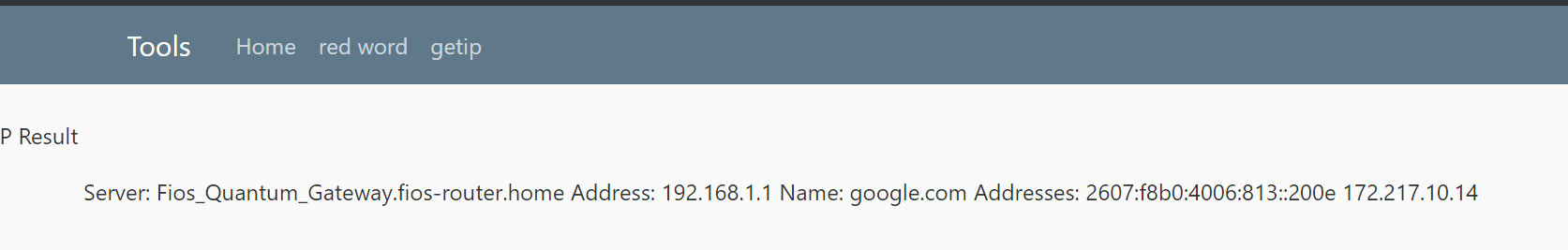
For assignment 2b, the task was to create our intentionally vulnerable web apps secure. From the last assignment my approach was to create intentionally vulnerabilities of the following:

* XSS
* Command Injection

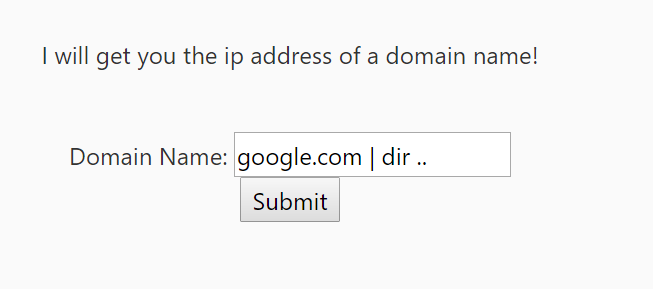
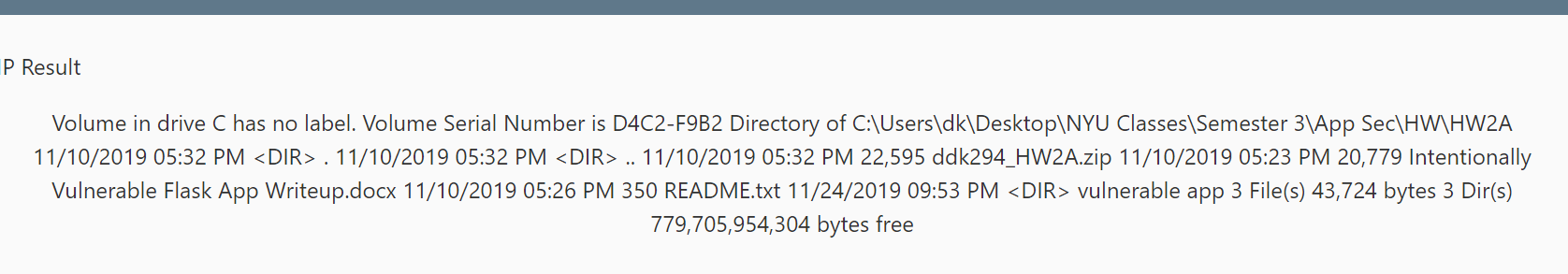
To make it vulnerable, I turned off autoescape via jinja2 when the output was being loaded into the DOM. To make it secure, I turned the autoescape back on for it to autoescape for me. I learned that utilizing libraries with autoescape is very powerful as input sanitation has been handled and that Jinja2 seems to be very powerful for both security and template inheritance in creating webapps.

The next vulnerability for Command Injection, in order to make the webapp vulnerable, I directly returned the output of the nslookup OS command. This was insecure for 2 reasons:

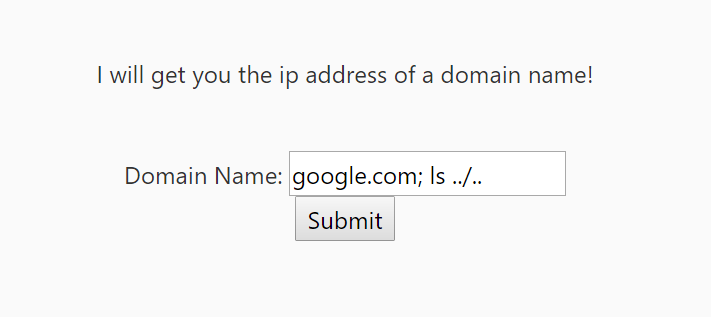
1. It showed the entire output of the nslookup command, and therefore more information could be gathered such as the domain name of the beginning DNS server, which my router was acting as. This in turn can lead to attackers knowing the sort of router the webserver was using. This in turn could lead to other sorts of exploits. See below:

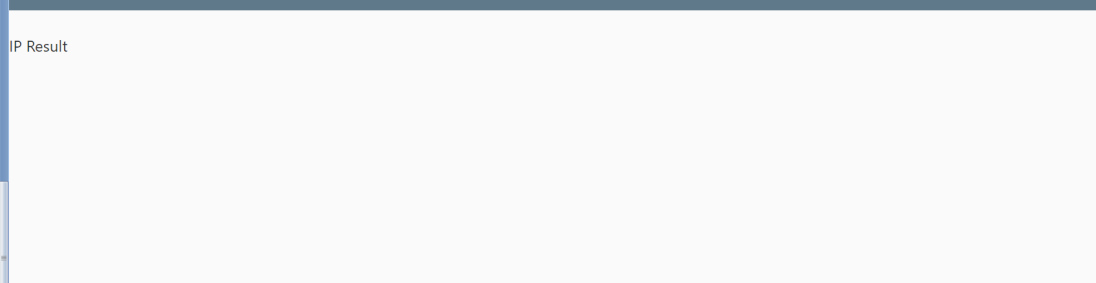
 🡪

1. Command Injection. The attacker could:
   1. Command Injections. See ScreenCaps Below:
      1. Literally do anything the OS allowed the attacker to do in terms of OS commands. Exfiltrate information, pipe inputs/scripts into files for malware or creating other webpages for the server, changing files, etc.
   2. determine based off of chaining commands whether it was a UNIX/LINUX based server or Windows server based on how chaining occurred. See ScreenCaps Below.
      1. Windows Command Chaining:

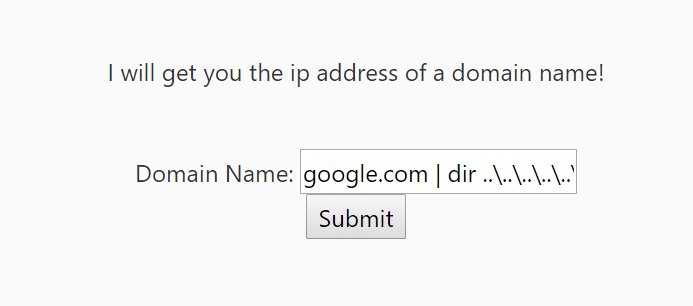
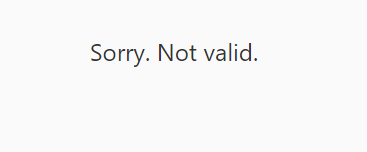
 🡪

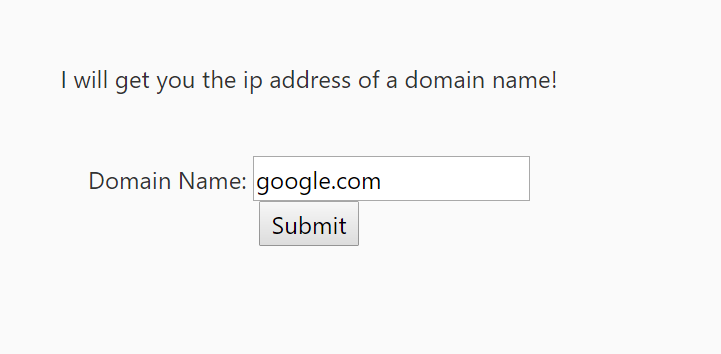
* 1. Linux/Unix command chaining:

🡪



I proceeded to solve this vulnerability by input validation by only accepting alphanumeric characters and splitting the strings at periods to be able to determine that it indeed was a domain name. I then only returned the IP address as well.

🡪 

🡪 