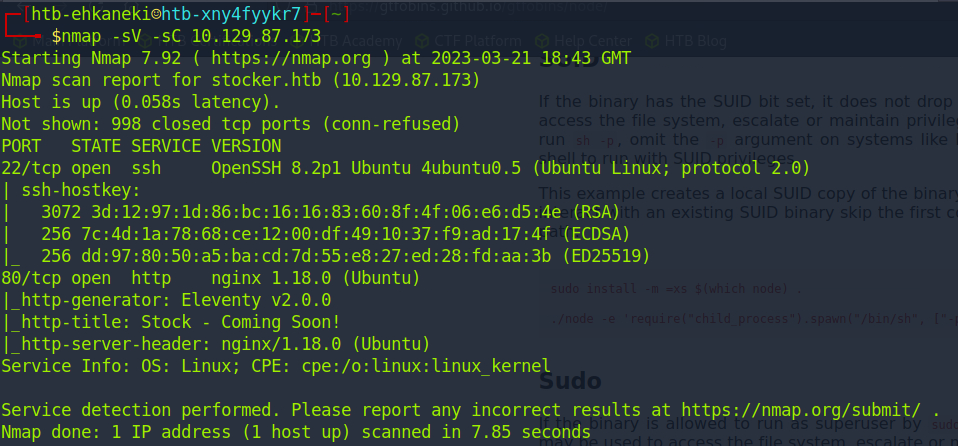
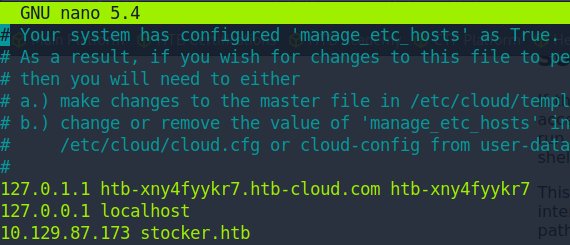
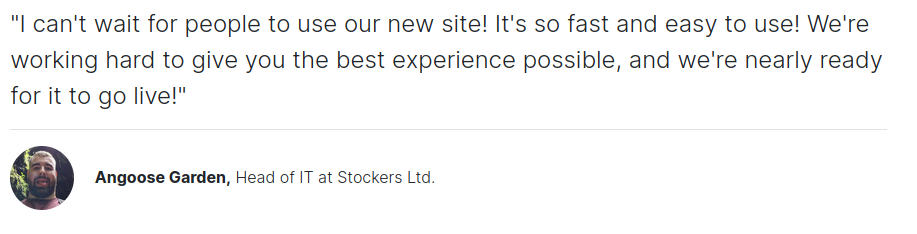
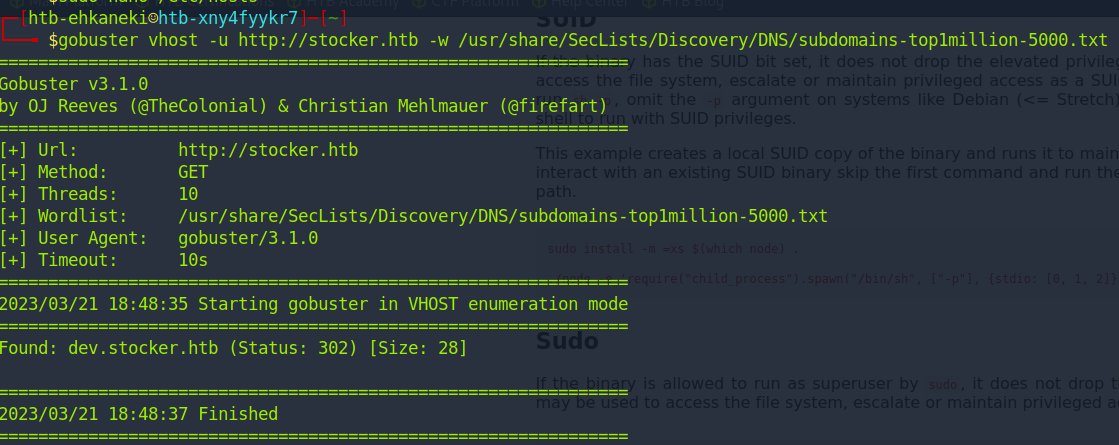
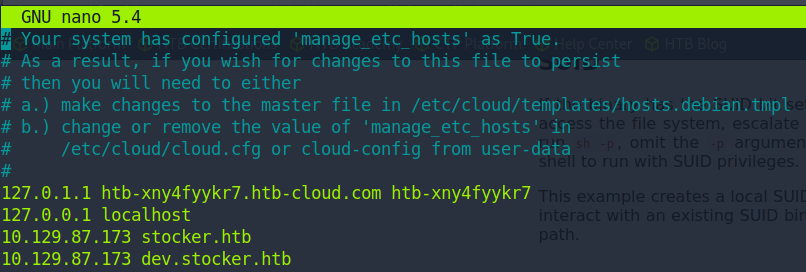
Stocker HTB Writeup

Started with a Nmap scan 

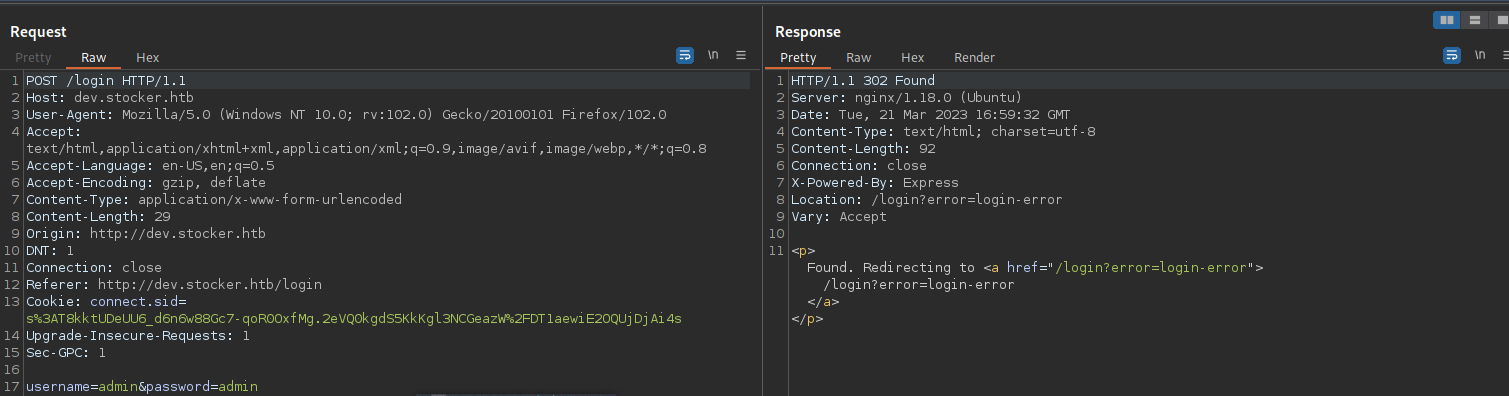
We get the basic two ports open SSH and a HTTP server. I first checked to see if there was any vulnerabilities with the services running first but nothing jumped out yet. So I went to enumerate the http server running. But first added the IP to my hosts file. 

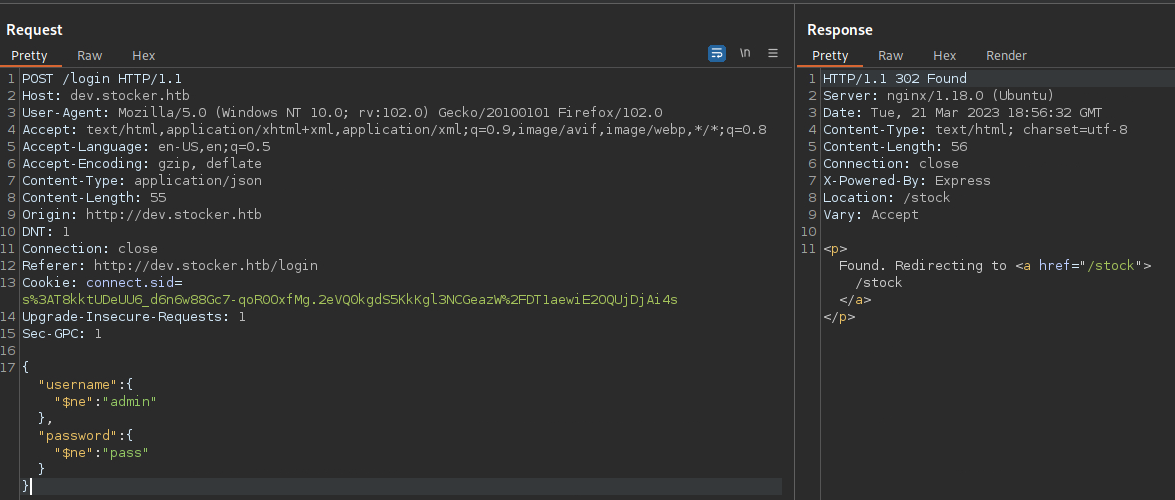
After exploring the website the only interesting thing I was able to find was the name of the head of the IT department Angoose which maybe can come in handy later on in the machine.

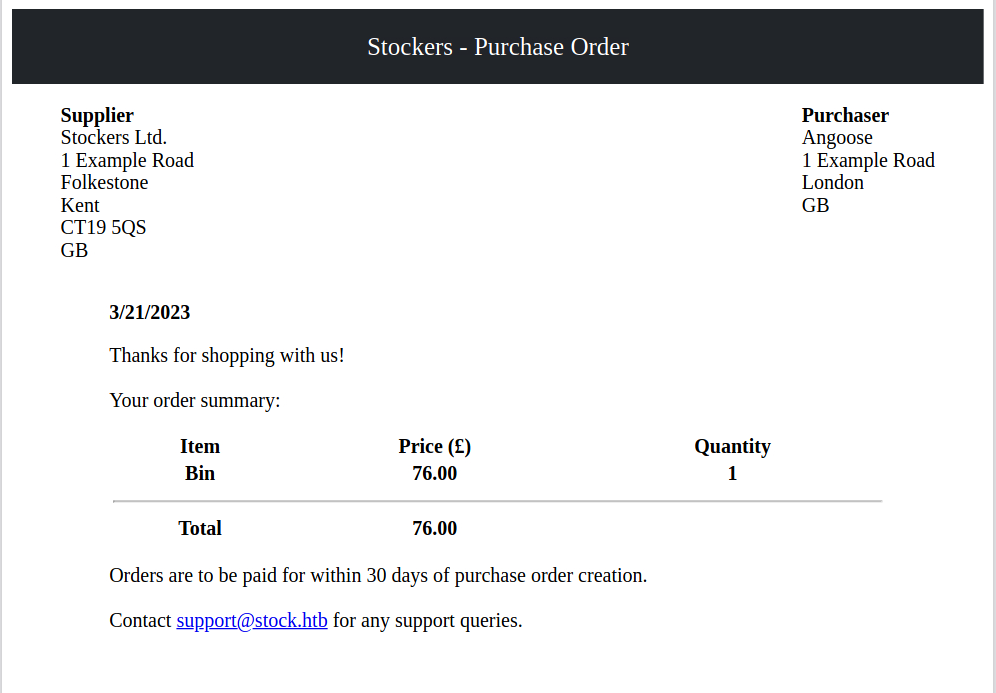
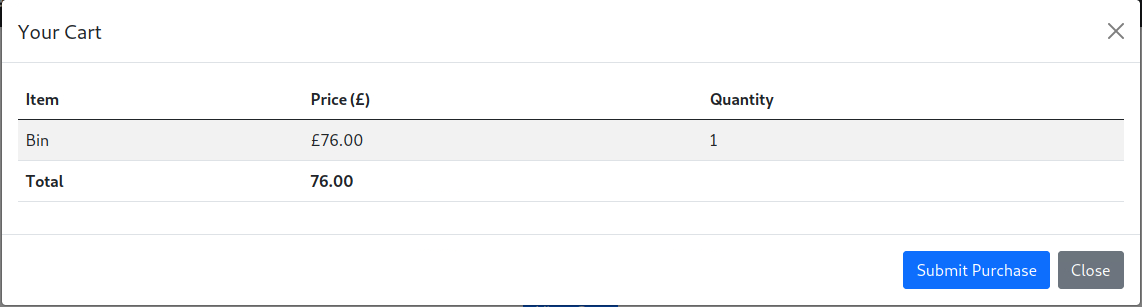
I then started the enumeration process for subdomains. Thus finding dev.stocker.htb

Adding the new sub domain to the hosts file 

The new sub domain led to a login page that I began to inspect the page to see if there was anything interesting on the front end. To figure out how to login page was working I submitted some random credentials into the login page to see what the web traffic looked like going to the server.



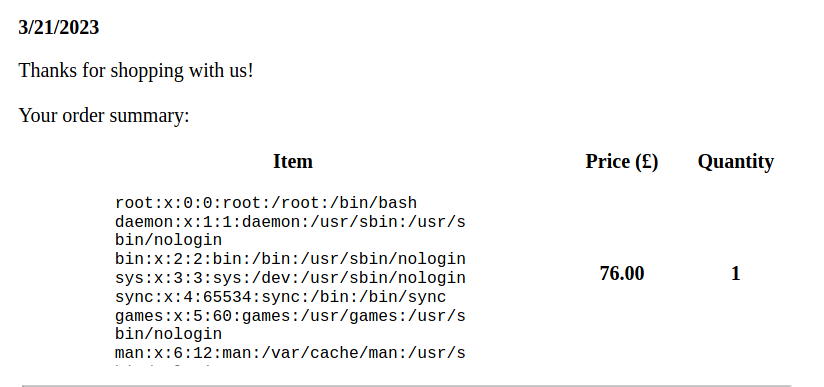
Since I saw the response was looking for something in JS I changed the content type in the request to ‘application/json’ and the login parameters to attempt a NoSQL injection using default credentials ‘{"username":{"$ne":"admin"}, "password":{"$ne":"pass"}}’ to bypass the login and to see if we get a different response from the server. 

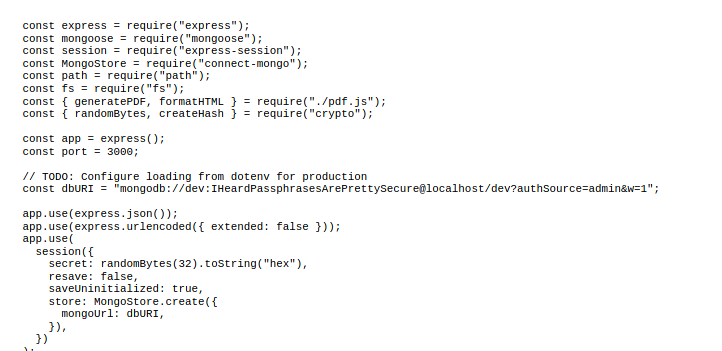
We get a response that the request is trying to redirect to a new directory of the domain called ‘/stock’. Bringing us to a inventory purchasing page. Which allows us to add items to a cart submitting the purchase and getting a PDF returned to us documenting our purchase we had made. 

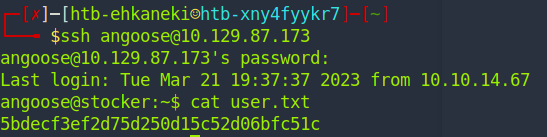
I then checked the request/response in BurpSuite to check what was happening when clicking on submit purchase within the window. 

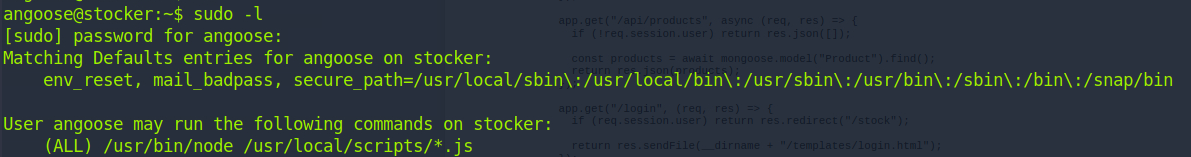
After doing some research on behind the software that was executing the PDF file I find that it is vulnerable to XSS and allows us to exploit it by making it run a script to read a local file on the web sever running.

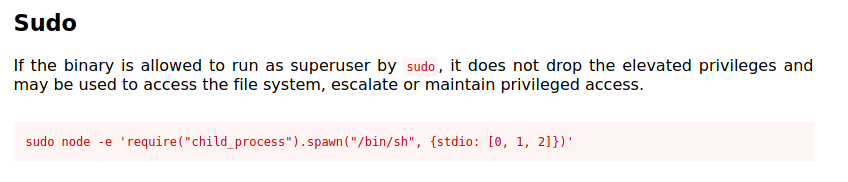
<https://book.hacktricks.xyz/pentesting-web/xss-cross-site-scripting/server-side-xss-dynamic-pdf>

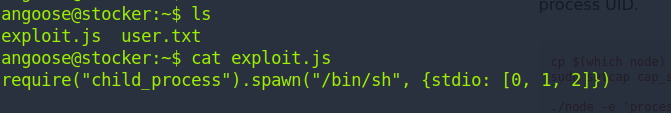
Reading some of the information on the hacktricks website we see that if we insert a iframe or embed tag within the title field it will print the results within the PDF after executing it. Using ‘<embed src=file:///etc/passwd></embed>’ it prints us the results of the passwd file. I noticed it wasn’t showing all of it properly so I changed the script to change the size as well to display it properly ‘<embed height=1000px width=1000px src=file:///etc/passwd></embed>’ 

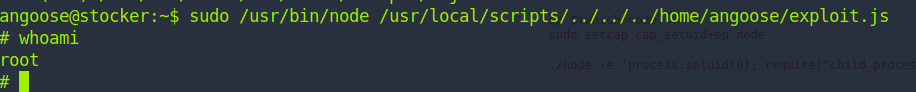
After messing with the file paths I checked the local path that the webapp was being hosted to display the main JS file that was being presented within the directory. Using "<embed height=1000px width=1000px src=file:///var/www/dev/index.js></embed>" it shows some interesting things. 

We see that we have a password for a user on the system we can start using to try to SSH into one of the users behind the web application. Since we saw very early on the “Angoose” was the head of the IT department and one of the users that showed after displaying the ‘passwd’ file we can try his username first with the password we found. 

Thus finding our first flag. Now to continue enumerating to get the root flag of the machine with ‘sudo -l’ 

We now see we can run ‘/usr/bin/node /usr/local/scripts/\*.js’ as root which we can start exploring options to exploit as well. Since we can run any Java Script file that we make within that path we can start making a JS file containing a exploit to become as the user root. I then went to <https://gtfobins.github.io/> to see what we can do within the node binary to spawn as root. 

We now have the JS code to implement the exploit so lets go create the JS file to run. 

After running ‘sudo /usr/bin/node /usr/local/scripts/../../../home/angoose/exploit.js’ we spawn as the root user 

Thus giving us the root flag of the machine. 