InfoSec Prep: OSCP VM

VM Link: <https://www.vulnhub.com/entry/infosec-prep-oscp,508/>

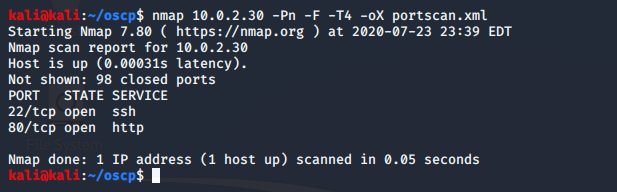
Writeup

To start, the VM displays its IP address on startup, saving us a discovery scan. 

IP Address: 10.0.2.30

We run an nmap scan against the host to discover running services.

nmap 10.0.2.30 -Pn -F -T4 -oX portscan.xml

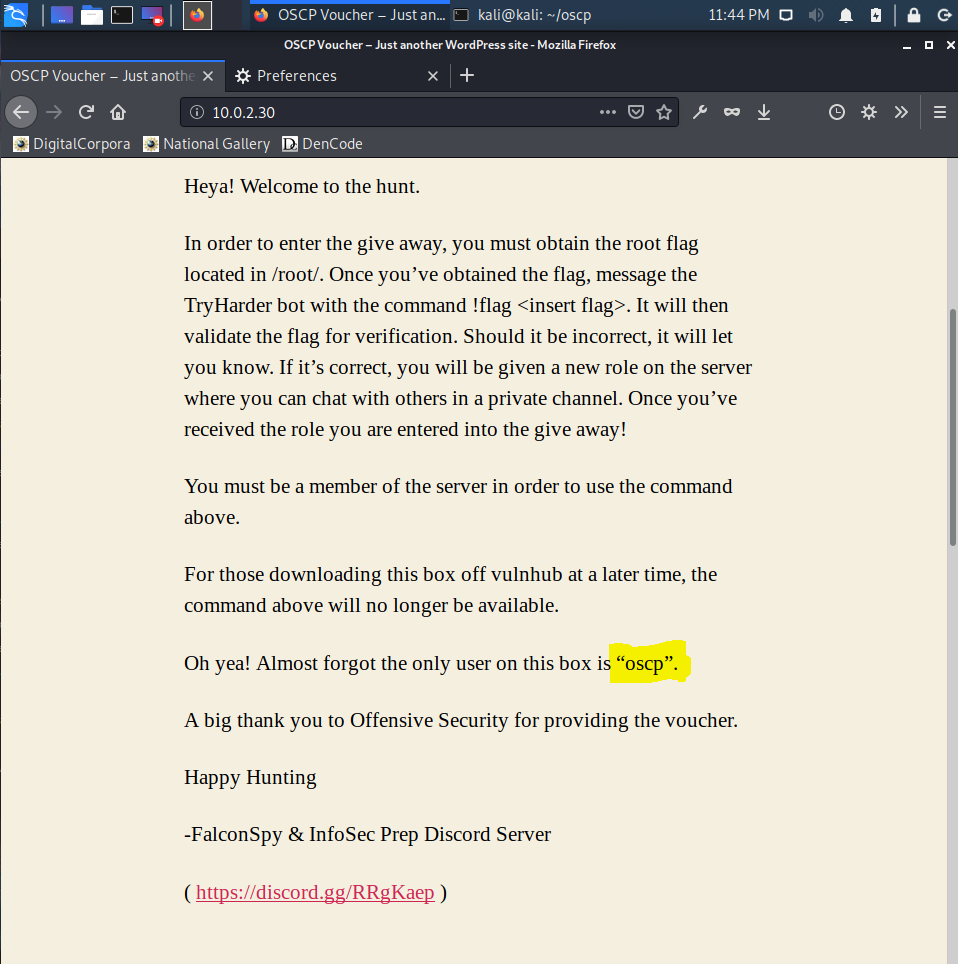


Running services:

22/tcp ssh

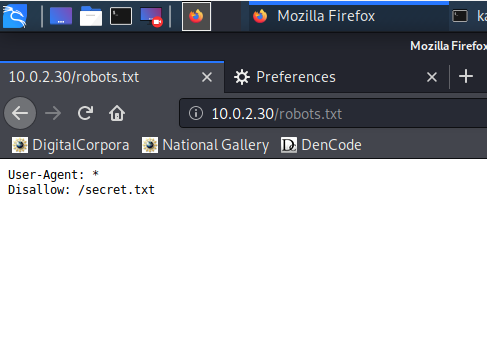
80/tcp http

Visiting the page, we find a username written in the latest post.

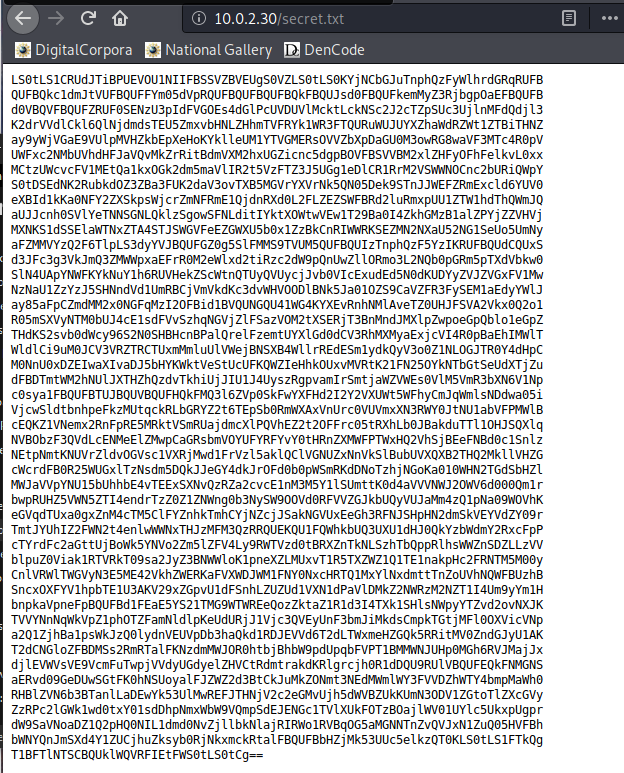


Username: oscp

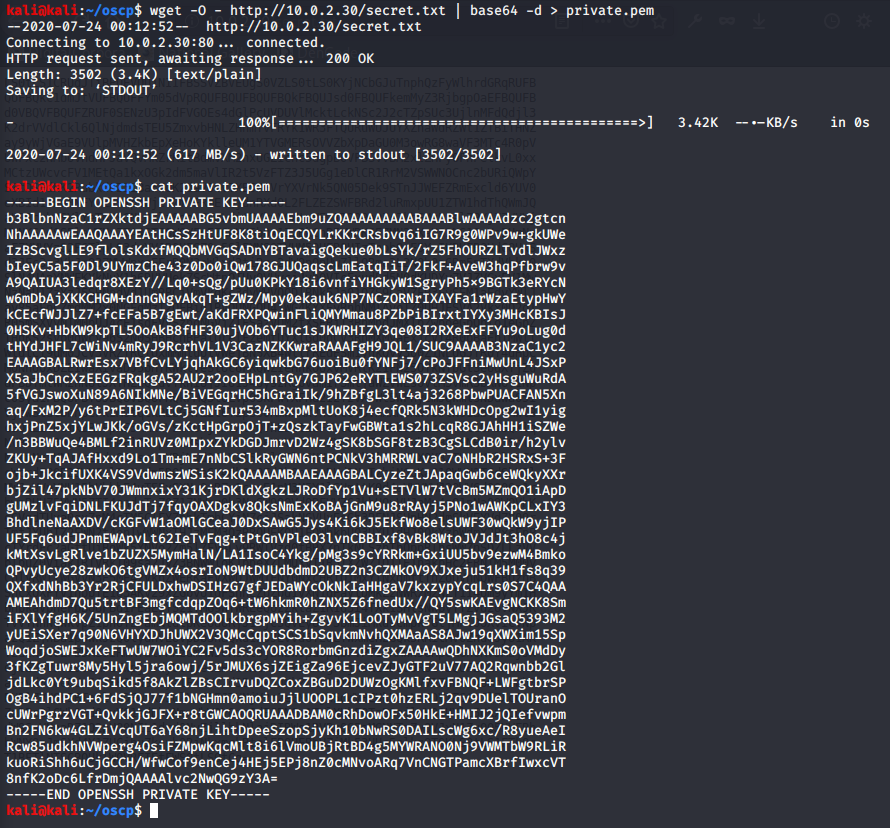
Next, we move on to robots.txt and discover a file..



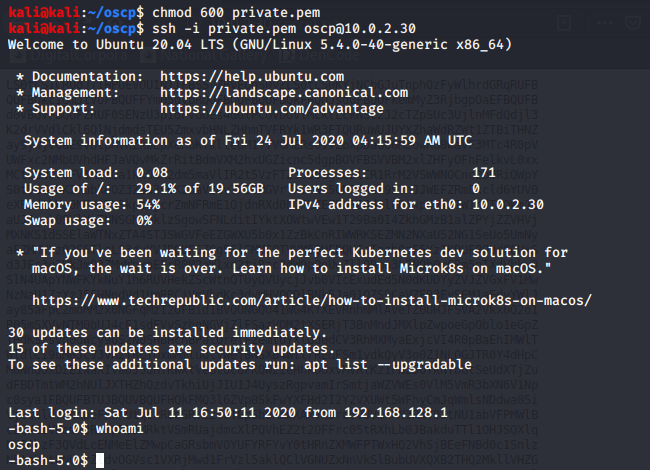
File: /secret.txt



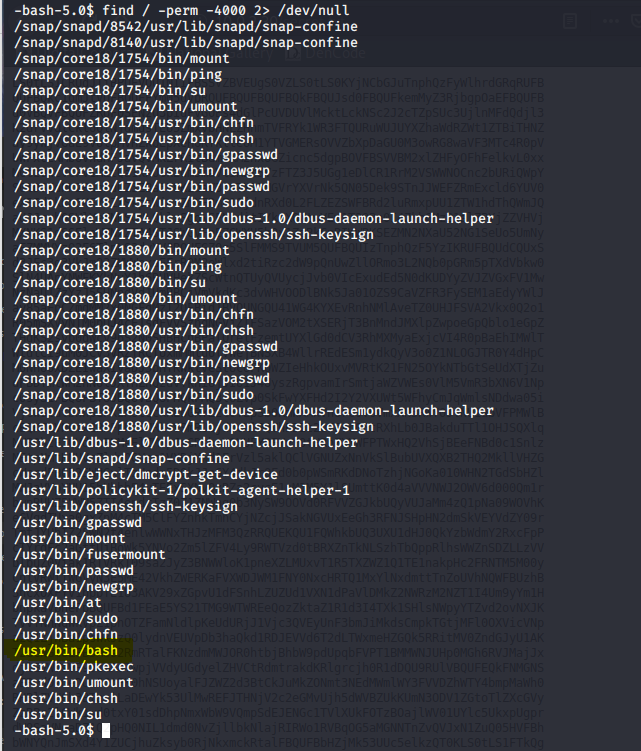
The padding at the end of the file suggests this is base64 encoded, so we download and decode it in the command line. We see this file contains a private key and that it is used for openssh. We save this as private.pem.



We now have both a username (oscp) and a private key associated with ssh. So we try them in combination and successfully log in to the server. Remember ssh requires keys to have specific permissions. We achieve this with ‘chmod 600 private.pem’.



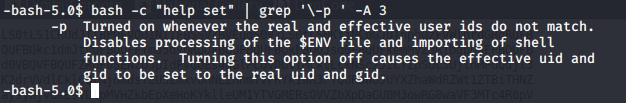
We start enumerating the system. We use ‘find / -perm -4000 2> /dev/null’ to find any binaries with the SetUID attribute set. We find several but the one of interest in this case is /usr/bin/bash.



To explain this vulnerability, we use ‘ls l /usr/bin/bash’ to show permissions of the binary. We can see the SUID flag is set, which sets the effective uid to that of whoever owns the file; in this case root.



Bash has a flag ‘-p’ which when enabled will cause the session to run with the $ENV file and shell functions of the effective user.



Because of this we can easily spawn a shell as root and obtain the flag.

