Exercicis de comandes de Linux relacionades amb el Nivell 4 del model TCP/IP

1.- a) ¿Què mostra la comanda *ss -tun*? (la pots executar a la teva màquina real).

> NOTA: Fes servir sempre els paràmetre *-t* i *-u*; el primer mostra només els ports TCP i el segon els UDP. Si no es posen, *ss* mostrarà també els sockets de tipus Unix, que són locals i no ens interessen per res, embrutant la sortida.

aII) Obre un navegador i torna a observar la sortida d'aquesta comanda. ¿Quins canvis hi aprecies respecte el que apareixia a l'apartat anterior? ¿A qui correspon les noves adreces IP que surten a la columna "Peer Address"? ¿Per què hi ha connexions als ports 80 i 443?

aIII) Tanca el navegador i torna a observar la sortida de *ss -tun*. ¿Què hi veus ara? ¿Què significa l'estat "CLOSE_WAIT"?

b) ¿Quina diferència hi ha entre afegir el paràmetre *-l*, afegir el paràmetre *-a* o no afegir-ne cap a la comanda *ss -tun*?

c) ¿Per a què serveix afegir el paràmetre *-p* a la comanda *ss -tun* i/o a la comanda *ss -tunl*?

> NOTA: Aquest paràmetre només funciona executant *ss* com a "root". L'hauries de provar en una màquina virtual (amb la tarja en mode NAT o adaptador pont)

d) Utilitza la comanda *watch* juntament amb *ss -tun* per tal d'obtenir en temps real l'estat de les connexions. Fes servir el paràmetre *-n* de *watch* per a què actualitzi la pantalla cada 3 segons en comptes de cada 2 segons (consulta el manual)

e) Sabent que la sintaxi és *ss -tna state estatConnexio [state unAltreEstat ...]*, mostra només les connexions que tinguin l'estat "established" i les que tinguin l'estat "listening"

> NOTA: Altres estats possibles són: syn-sent, syn-recv, fin-wait-1, fin-wait-2, time-wait, closed, close-wait, last-ack, listening, closing. També està connected (que és igual a tots excepte closed i listening)

eII) Sabent que la sintaxi és *ss -tna dst di.recc.ió.IP[:nºport]*, mostra només les connexions que tinguin com a destí la direcció IP 192.168.12.10. ¿I com a destí la direcció IP de xarxa 192.168.12.0/24?

> NOTA: També existeix la possibilitat de filtrar per direcció IP d'origen amb la sintaxi *ss -tna src direccioIP[:nº]*
> NOTA: També existeix la possibilitat de filtrar només per ports, ja siguin de destí (amb el filtre *dport :nº*) o d'origen (amb el filtre *sport :nº*) . Fins i tot, en aquest cas, es poden fer servir els operadors *gt, ge, lt* i *le* per indicar ports majors, majors o igual, menors i menors o iguals que l'indicat, respectivament (per exemple: *dport gt :10*)
> NOTA: Es poden pot posar més d'un filtre a la vegada; en aquest cas s'entendrà que la connexió cal que compleixi tots els filtres per ser mostrada (és a dir, per defecte s'usa un operador AND impícit). Si es vol fer ser l'operador OR, cal encerclar els filtres implicats entre cometes i parèntesis i unir-los amb la paraula *or*. Per exemple, la comanda *ss -tna state established dst 192.168.12.10 "(dport = :80 or sport = :443)"* mostrarà només les connexions establertes amb els ports 80 o 443 de l'ordinador remot 192.168.12.10

f) ¿Quina informació conté l'arxiu "/etc/services" ? (pots consultar l'arxiu que hi ha a la teva màquina real i la seva pàgina del manual, si cal)

2.-Encén una màquina virtual qualsevol que tingui com a mínim una tarja de xarxa (considerarem que és "enp0s3"), en mode "adaptador pont" i que no tingui cap configuració prèvia que puguis haver deixat dels exercicis anteriors (a "/etc/netplan", "/etc/systemd/network" o "/etc/NetworkManager/system-connections") per tal de què simplement obtingui la seva configuració del servidor DHCP del centre, com qualsevol altra ordinador de l'aula. Instal.la en aquesta màquina virtual els paquets "vnstat" i "vnstati" (a Ubuntu) o "vnstat" (a Fedora) i engega el servei corresponent (*sudo systemctl start vnstat*). Més endavant veurem per a què serveix, però de moment mantingues-lo engegat.

> NOTA: Existeixen més programes similars a *vnstat*, com per exemple ***darkstat***, però no el veurem

3.-a) Instal·la a la màquina virtual usada a l'exercici anterior el paquet "ncat" (a Ubuntu) o "nmap-ncat" (a Fedora). A partir d'aquí, esbrina per a què serveixen els paràmetre *-l, -p, -k, -v* i *-u* de la comanda *ncat* (consultant els apunts o bé el seu manual)

b) Explica per a què poden servir els següents usos de *ncat* (els pots provar fent servir el client i el servidor a la mateixa màquina o bé fent servir el client *nc* a la màquina real (on ja ve instal·lat "de sèrie") per connectar al servidor *ncat* funcionant a la màquina virtual):

> NOTA: ¡Atenció!, a sistemes Fedora hauràs de desactivar prèviament el tallafocs (el qual ve activat de sèrie) per tal de què els clients puguin connectar amb el servidor sense problemes. Això simplement es fa executant la comanda *sudo systemctl stop firewalld*

Servidor: *ncat -l -p 5588* <---> Client: *ncat ipservidor 5588*
Servidor: *ncat -k -l -p 5588* <---> 2 clients: *ncat ipservidor 5588*
Servidor: *ncat --broker -l -p 5588* <---> 2 clients: *ncat ipservidor 5588*
Servidor: *ncat -l -p 5588 <arxiu* <---> Client: *ncat ipservidor 5588 > arxiu*
Servidor: *ncat -l -p 5588* <---> Client: *echo "HOLA" | ncat ipservidor 5588*
NOTA: També es poden enviar dades binàries, així: *echo -ne "\x80\x45\x71" | ncat ipservidor 5588*
Servidor: *ncat -l -p 5588 | dd of=arxiu.iso* <---> Client: *dd if=/dev/sdb | ncat ipServidor 5588*

c) Posa en marxa un servidor *ncat* en qualsevol port. Utilitza la comanda *ss* per confirmar que, efectivament, estiguis escoltant en aquest port. Conecta-t'hi llavors (o demana a algun company que s'hi connecti) amb un client *nc/ncat* i torna a utilitzar la comanda *ss* per confirmar que, efectivament, hi ha una connexió establerta amb un client del qual pots esbrinar la seva direcció IP. Si, finalment, es tanca aquesta connexió (interrompint el Netcat client, per exemple), ¿què es veurà al ss?

d) Crea un script anomenat "hola.sh" amb el següent contingut (i assigna-li permisos d'execució!!):

```
#!/bin/bash
echo "Digue'm el teu nom"
read nom
echo "El teu nom és $nom"
```

Executa ara en un terminal la comanda *ncat -k -l -p 5000 -e "./hola.sh"* i en un altre *ncat 127.0.0.1 5000*. ¿Què passa? Per què? Consulta el manual si necessites saber per a què serveix el paràmetre *-e*.

dII) Si en un terminal tens executant-se la comanda *ncat -k -l -p 5000 -e "/bin/date"* i en una altra la comanda *ncat --recv-only 127.0.0.1 5000* , ¿què estem fent?

dIII) I si tens executant la comanda *ncat -k -l -p 5000 -e "/bin/bash"*, ¿què passa quan ara t'hi connectes fent *ncat 127.0.0.1 5000*?

e) ¿Per a què serveixen els paràmetres *--allow/--allowfile*, *--deny/--denyfile* i *--max-conns* de *ncat*? (consulta el seu manual)

f) Executa en un terminal les comandes encanonades *cat /dev/urandom | tr -dc [:print:] | ncat -l -p 8080* i en un altre terminal la comanda *ncat 127.0.0.1 8080* ¿Què passa? ¿Per què? Consulta el manual de *tr* si necessites saber per a què serveixen els seus paràmetres *-d* i *-c* i *[:print:]* i la pàgina *man urandom* per saber en què consisteix el dispositiu "/dev/urandom" A partir d'aquí, ¿quina utilitat creus que tindria (no ho facis!) substituir el client utilitzat en aquest exemple per aquest altre: *ncat 127.0.0.1 8080 | sudo dd of=/dev/sdb*

> NOTA: Usos concrets de la comanda *tr* , més enllà del cas concret d'aquest apartat, poden ser els següents:
> *echo "hola" | tr oa ei* --> Mostrarà "heli"
> *echo "hola" | tr [:lower:] [:upper:]* --> Mostrarà "HOLA"
> *echo "hoooolaaaa" | tr -s oa* --> Mostrarà "hola"
> *echo "hoooolaaaa" | tr -d oa* --> Mostrarà "hl"
> *echo "hoooolaaaa" | tr -cd "oa"* --> Mostrarà "ooooaaaa"

4.-a) Instal.la a la màquina virtual el paquet "nload" i posa en marxa el programa corresponent (així: *nload enp0s3*) ¿Què veus? Per a què serveix la tecla **F2**? I la tecla **q**?

NOTA: Existeixen molts altres programes similars com per exemple ***bwm-ng, ifstat, slurm, bmon*** o ***nicstat***

b) Instal.la el paquet "tcptrack" i posa en marxa el programa corresponent (així: *tcptrack -i enp0s3*). Fixa't que la diferència entre aquest programa i els mencionats a l'apartat anterior és que aquest discrimina l'ample de banda utilitzat a nivell de connexions en comptes de a nivell de tarjes de xarxa. ¿Per a què serveixen les tecles **p**, **s** i **q** dins del mode interactiu del programa? ¿Per a què serveix el paràmetre *-d* a l'hora d'executar-lo (consulta el manual)? ¿Què fa la comanda *tcptrack -i enp0s3 src or dst 10.45.165.2*? ¿I la comanda *tcptrack -i enp0s3 dst port 80* ?

NOTA: Existeixen més programes similars a *tcptrack*, com per exemple ***iftop*** o ***trafshow*** (molt coneguts també)

c) Instal.la el paquet "nethogs" i posa en marxa el programa corresponent (així: *nethogs enp0s3*). Fixa't que la diferència entre aquest programa i els mencionats a l'apartat anterior és que aquest discrimina l'ample de banda utilitzat a nivell d'aplicacions en comptes dei a nivell de conexions o tarjes de xarxa. ¿Per a què serveixen les tecles **r**, **s** i **q** dins del mode interactiu del programa? ¿Per a què serveix el paràmetre *-d* a l'hora d'executar-lo (consulta el manual)?

d) Instal.la el paquet "tmux" i executa la comanda *tmux* Seguidament, pulsa les combinacions de tecles **CTRL+b** i després **%** . Seguidament pulsa **CTRL+b** i després **"**. Torna al terminal de la dreta amb les tecles **CTRL + b Cursor_adient** i torna a pulsar *CTRL+b i "* . Posa en marxa en un dels quatre terminals que hauràs d'haver obtingut el programa *nload*, en un altre el *tcptrack* i en un altre el *nethogs*, quedant-te un quart terminal per treballar. ¿Què veus? Fes una captura del resultat.

dII) Amb la sessió tmux encara oberta de l'apartat anterior, escriu ara **CTRL+b d** .¿Què passa? Escriu seguidament al terminal la comanda *tmux attach.* ¿Què passa ara? ¿Quins usos li trobes a aquesta funcionalitat?

NOTA: Per sortir definitivament de tmux, escriu **exit** (o pulsa **CTRL+d**) a totes les finestres per anar-les tancant fins arribar a sortir del tot.

2BIS.-Ja fa estona que vas posar en marxa el servei vnstat a la màquina virtual; ara és l'hora de recollir les dades obtingudes sobre l'activitat de la xarxa que aquest servei ha detectat

NOTA: Aquestes dades estan guardades per defecte dins de la carpeta "/var/lib/vnstat", en un fitxer diferent per cada tarja de xarxa detectada al sistema (excepte la loopback) anomenat igual que la tarja corresponent. Això es podria canviar editant el fitxer de configuració del programa, "/etc/vnstat.conf", el qual, de fet, té tres seccions: una és específica pel servidor *vnstatd*, una altra pel programa *vnstati* (que de seguida veurem) i una altra comú als tres programes (*vnstatd*, *vnstati* i comanda *vnstat*).

b) Executa la comanda *vnstat -i enp0s3*. ¿Què veus?

c) ¿Què passa si afegeixes a la comanda anterior el paràmetre *-d*? ¿O el *-m*? ¿O el *-h*? ¿O el *-l* (i esperes 10s)?

d) ¿Per a què creus que aniria bé afegir el paràmetre *--json*?

e) ¿Per a què serveix el paràmetre *--remove*? Fixa't que no podràs tornar a monitoritzar la tarja indicada a no ser que utilitzis el paràmetre *--add* (o bé, aturis i tornis a iniciar el servei vnstat)

f) Executa la comanda *vnstati -c 15 -ne -h -i enp0s3 -o hola.png.* (consulta en el manual per a què serveixen els paràmetre *-c* i *-ne*). Veuràs que ha generat una foto; obre-la amb un visor de fotos i digues què veus

NOTA: Si estàs treballant en un sistema sense entorn gràfic, pots instal.lar el paquet "fbida" que proporciona la comanda *fbi*, la qual és un visor de fotografies per terminal. Si estàs en Gnome, pots fer servir el visor "eog" (és el que ve x. defecte)

5.-a) Instal.la a la màquina virtual el paquet "nmap" i fes-lo servir per esbrinar (fent servir el paràmetre *-sn*) quines màquines estan enceses a la xarxa de l´aula. Observa el temps emprat.

b) Mira (fent servir els paràmetre *-p 1-10000* i *-Pn*) quins serveis té oberts una màquina remota qualsevol (del centre o d'Internet). ¿Obtens la mateixa informació si no indiques el paràmetre *-p 1-10000* ? ¿Per què?

c) Esbrina, fent servir els mateixos paràmetres que a l'apartat anterior però afegint-hi ara el paràmetre *-sV,* quin programa (i quina versió) té funcionant una màquina remota qualsevol darrera de cada port obert

d) Esbrina, fent servir els mateixos paràmetres que a l'apartat b) però afegint-hi ara el paràmetre *-O* i executant la comanda *nmap* com a "root", quin sistema operatiu té funcionant una màquina remota qualsevol.

> NOTA: A https://nmap.org/book/osdetect.html es pot llegir tota la teoria que hi ha al darrera del funcionament del paràmetre -O i, en general, de les tècniques d'"OS fingerprinting" que fa servir Nmap.

e) ¿Per a què serveix el paràmetre *-iL fitxer.txt*? ¿I el paràmetre *-oG fitxer.txt*?

6.-a) Executa la comanda *nmap --packet-trace -n -sn 192.168.12.10* ¿Quina tècnica ha emprat Nmap per descobrir si el destí indicat està present a la xarxa?

> NOTA: El paràmetre -n és per a què Nmap no faci cap consulta DNS (en aquest cas, inversa) als servidors DNS configurats al sistema, cosa sempre que fa per defecte (en el cas de voler fer servir uns altres servidors DNS diferents es podria afegir el paràmetre *–dns-servers x.x.x.x,y.y.y.y,...*). En qualsevol cas es pot comprovar aquest comportament a pantalla si s'esborra aquest paràmetre i es torna a provar la comanda.

b) Repeteix la mateixa comanda anterior però ara contra un ordinador de la teva LAN que sàpigues realment que no té el port 80 obert. ¿Com descobreix ara Nmap si el destí indicat està present a la xarxa (o no)?

c) Repeteix la mateixa comanda anterior però ara executant-la com a "root" (és igual el destí mentre sigui de la teva xarxa LAN). ¿Quina tècnica ha emprat ara Nmap per descobrir si el destí indicat està present a la xarxa? ¿I si ara hi afegeixes, a la comanda anterior, com a "root", el paràmetre *--send-ip*? ¿I si afegeixes a tot l'anterior el paràmetre *-PE*?

d) Repeteix la mateixa comanda del primer apartat però ara dirigida a un ordinador de fora de la teva LAN (com per exemple "www.hola.com"). ¿Quina tècnica ha emprat Nmap per descobrir si el destí indicat està present a la xarxa?

e) Repeteix la mateixa comanda anterior però ara executant-la com a "root". ¿Quina tècnica ha emprat ara Nmap per descobrir si el destí indicat està present a la xarxa? ¿I si afegeixes a tot l'anterior el paràmetre *-PE*?

f) ¿Per a què creus que serviria executar la comanda *watch -n 10 nmap -n -sn 192.168.12.0/24* ?

7.-a) Executa la comanda *nmap --packet-trace --reason -n -Pn -p80 192.168.12.10* ¿Quina tècnica ha emprat Nmap per descobrir els ports que té oberts el destí indicat? ¿I si executes la mateixa comanda però indicant el port nº81? ¿Quina diferència hi veus, en general, si esborres el paràmetre *-Pn*?

b) ¿Quina diferència hi ha entre executar la primera comanda de l'apartat anterior com a usuari normal i com a "root"? (és a dir, fent per defecte l'escaneig *-sT* o fent el *-sS*, respectivament)? ¿I si executes la mateixa comanda però indicant el port nº81?

c) Repeteix les mateixes comandes del primer apartat (és a dir, una apuntant al port 80 i una altra al port 81, i totes amb *-Pn* per simplificar) però ara dirigida a un ordinador de fora de la teva LAN (com per exemple "www.hola.com"). ¿Quina tècnica ha emprat ara Nmap per descobrir els ports que té oberts el destí indicat? ¿I si les executes com a "root"?

d) Executa les següents comandes, una rera l'altra, i digues si veus alguna diferència en els resultats obtinguts (i en el cas que sí, per què creus que apareix la diferència observada):

nmap --packet-trace -n -Pn -p 80 192.168.12.10 (com a "root" i no)
nmap --packet-trace -n -Pn -p 80 -sF 192.168.12.10 (com a "root")
nmap --packet-trace -n -Pn -p 80 -sA 192.168.12.10 (com a "root")
nmap --packet-trace -n -Pn -p 80 -sN 192.168.12.10 (com a "root")
nmap --packet-trace -n -Pn -p 80 -sX 192.168.12.10 (com a "root")
nmap --packet-trace -n -Pn -p 80 -sU 192.168.12.10 (com a "root")

NOTA: A l'article https://resources.infosecinstitute.com/topic/port-scanning-using-scapy apareixen diversos diagrames que mostren l'intercanvi de paquets (i el que s'esperaria de resposta) quan es realitzen diferents tipus d'escaneig, com ara -sS, -sT, -sU, -sX, -sW, -sN, -sF, -sA, etc) juntament amb una explicació de perquè aquests diferents escaneigs són útils per esbrinar, de diferents formes més o menys fiables segons les circumstàncies, si el port investigat respon -és a dir, "està obert"- o no. També es mostren uns codis Python que permeten implementar aquests escanejos fent servir una llibreria anomenada Scapy que estudiarem properament (per ara no cal fer-li cas). Un altre recurs molt interessant sobre aquest aspecte és la pròpia pàgina del manual del Nmap o també https://nmap.org/book/man-port-scanning-techniques.html

10.-a) Fes una petició ARP (per tant, broadcast, recordem-ho) amb la comanda *nping* preguntant per la MAC d'una màquina qualsevol d'un company de classe, de la qual coneixes la IP.

NOTA: La IP obtinguda en la resposta, no obstant, degut al funcionament intern de *nping*, no es guardarà a la teva taula ARP del sistema (i, per tant, no apareixerà a la sortida de la comanda *ip neigh show*)

aII) Repeteix l'apartat anterior però ara indicant la IP d'una màquina que no pertanyi a la teva xarxa local (com per exemple, 1.1.1.1) ¿Què passa ara? ¿Per què?

aIII) ¿Què fa la comanda *nping --arp --arp-type ARP-reply 192.168.12.10* ?

b) Simula el comportament de la comanda *ping* (és a dir, un enviador de paquets ICMP de tipus "Echo-request") amb els paràmetres adients de *nping* contra la màquina "www.hola.com". ¿Quina resposta obtens? Si canvies el tipus del paquet ICMP enviat pel tipus 3, ¿reps ara alguna respossta?

c) Posa en marxa un servidor Netcat d'aquesta manera: *ncat -u -l -p 1111* i, en un altre terminal, executa la comanda *nping --udp -p 1111 --data-string "hola" -c 1 127.0.0.1* ¿Què passa? Atura finalment el servidor Netcat.

cII) Envia un paquet UDP amb un TTL molt baix a qualsevol lloc d'Internet. ¿Què vol dir el missatge "TTL=0 during transit (type=11/code=0)" que veuràs a la pantalla? ¿Què té a veure això amb la comanda *mtr*?

d) Posa en marxa un altre servidor Netcat però ara d'aquesta manera: *ncat -l -p 1111* . ¿Quina és la diferència amb el servidor Netcat implementat en un apartat anterior? Mentre està en marxa, ara en un altre terminal executa la comanda *nping --tcp -p 1111 --data-string "hola" -c 1 127.0.0.1* ¿Per què ara no es veu res a la pantalla del servidor Netcat? Pista: fixa't en els paquets que està enviant i rebent la comanda *nping*

NOTA: Existeix el parametre *–tcp-connect* de nping que completa el "3-way handshake" però no admet payloads

dII) ¿Què fa la comanda *nping -c 1 --tcp -p 80,443 192.168.12.105-109* ?