Atacs ARP amb Nping

Tècniques de "Host discovery" a una LAN

Abans de passar a realitzar un atac de tipus "ARP Spoofing", cal saber primer quines són les nostres potencials "víctimes" (és a dir, els sistemes presents actualment a la LAN i, més específicament, la porta d'enllaç per defecte). La manera més habitual de realitzar un escaneig de xarxa és mitjançant l'enviament de peticions ARP a tota ella i esperar-ne rebre les respostes corresponent per identificar els equips encesos en aquell moment.

> NOTA: Aquest tipus d'escaneig s'anomena "actiu" perquè injecta tràfic provinent de l'atacant a la xarxa de les víctimes. Existeix un altre tipus d'escaneig de tipus "passiu" que consisteix simplement en parar-se a escoltar el tràfic existent a la xarxa sense participar-hi, de forma que en pugui deduïr igualment els equips presents. Els escanejos passius són més lents i poden no obtenir un resultat complet però a canvi són totalment imperceptibles.

Una manera fàcil de fer un escaneig mitjançant ARP és fent servir l'eina *nping*, proporcionada dins del paquet "nmap" (https://nmap.org). Aquesta eina ens permet generar "a la carta" (i enviar a un destí determinat) diferents tipus de paquets (IP, ICMP, TCP, UDP, etc) especificant valors concrets per camps de les capçaleres desitjades i entre els tipus de paquets que es poden generar (i enviar) hi ha justament els de tipus "ARP-request", que són els que ens interessen ara mateix (i també els de tipus "ARP-reply" també, que serà els que ens interessaran per fer l'atac de l'"ARP-Spoofing"). Així doncs, farem servir la comanda Nping (com a "root", però) per realitzar escanejos ARP massius a la nostra xarxa local

> NOTA: Existeixen altre eines que també permeten realitzar un "escombrat" ARP -o d'altres tipus!- per descobrir els sistemes accessibles en una xarxa, com ara "**arp-scan**" (https://github.com/royhills/arp-scan) o la pròpia "**nmap**" però amb *nping* ja en tindrem prou

Més en concret: per enviar una petició ARP (broadcast, per definició) es pot executar simplement la comanda *sudo nping --arp --arp-type arp-request ip.a.tro.bar* , on "ip.a.tro.bar" pot ser la IP d'un host concret (per saber si està encès o no), la IP d'una xarxa sencera (per saber quins hosts d'aquesta xarxa estan disponibles; aquesta és l'opció que ens interessarà més....en tot cas per a què funcioni caldrà indicar llavors la màscara en format CIDR), una IP amb el comodí * indicat en alguna part (per indicar un conjunt de hosts), un possible rang indicat amb guions -així: *ip.des.ti1-ip.des.ti2* - o bé diverses IPs individuals indicades separades per comes -així: *ip.des.ti1,ip.des.ti2,...-*). La idea, doncs, per descobrir tots els ordinadors presents, per exemple, a la LAN 192.168.12.0/24, seria fer *sudo nping --arp --arp-type arp-request 192.168.12.0/24* : aquesta comanda envia una petició ARP (en realitat més, veure paràmetre *-c* de la nota següent) a cadascun dels sistemes encesos a la xarxa per tal d'esbrinar la MAC associada a la IP 192.168.12.1, una altra petició ARP, de nou a tots els sistemes de la xarxa, per esbrinar la MAC associada a la IP 192.168.12.2, una altra petició ARP de nou a tothom per esbrinar la MAC associada a la IP 192.168.12.3, i així, fins arribar a la darrera IP de la xarxa (en aquest cas, 192.168.12.254). D'aquesta forma, en algun moment, cada màquina de la xarxa haurà respost amb la seva MAC a la petició ARP que li correspongui.

> NOTA: A la comanda anterior es poden afegir el següents paràmetres genèrics interessants:
> *-c nº* : indica el nombre de paquets (en aquest cas, peticions ARP) que es vol enviar (per defecte és 5)
> *--delay nº* : indica el temps (per defecte, en segons) que Nping s'esperarà entre enviament i enviament. Es poden indicar els sufixos "ms","m" o "h" (milisegons, minuts i hores), i el valor numèric pot ser decimal
> *--rate nº* : indica el nombre d'enviaments que Nping farà per segon. És la inversa de l'opció *–delay* (és a dir, *--rate 20* és el mateix que *--delay 0.05*); si ambdues opcions s'indiquen, només es tindrà en compte la darrera que s'escrigui a la línia de comandes
> *-H* : indica que no es vol veure per pantalla els paquets enviats (marcats per línies començant per "**SENT**") sinó només els paquets rebuts (marcats per línies començant per "**RCVD**")

0.-Arrenca una màquina virtual VirtualBox (pot ser de tipus Server) que tingui la seva tarja de xarxa en mode "adaptador pont" i instal.la-hi els paquets "nmap" i "wireshark-cli" (si és Fedora) o "tshark" (si és Ubuntu).

> NOTA: Per a què la comanda *tshark* funcioni sense haver de ser "root", hagis d'executar primer *sudo usermod -a -G wireshark elteuusuari* i reiniciar la sessió del teu usuari

1.-a) Observa (amb la comanda *ip -c link show*, per exemple) quina és l'adreça MAC de la tarja de xarxa d'aquesta màquina virtual i, tot seguit executa la comanda *tshark -Y "eth.src == MA:Cd:el:at:ar:ja and arp"* (on l'adreça MAC representa la de la tarja de la màquina virtual) ¿Què és el que es pretén veure amb la comanda anterior? (recorda si cal la sintaxi dels filtres de pantalla del Wireshark/Tshark)

aII) Mentre mantens la comanda *tshark* iniciada de l'apartat anterior executa ara en un altre terminal (recorda, en una màquina VirtualBox pots pulsar *CTRLdret+Fnº* per anar canviant de terminal) la comanda *sudo nping -c 1 -H --arp --arp-type arp-request 192.168.12.10* (on la IP indicada pot ser qualsevol adreça d'un ordinador de la teva xarxa local) ¿Què veus, efectivament, a la sortida del Tshark?

> NOTA: En el cas de voler visualitzar la mateixa informació que la que es mostra al panell de "detalls" de la finestra del Wireshark però a la sortida de la comanda *tshark*, només caldria afegir-li el paràmetre ***-V*** o, si només es volen veure només els detalls d'un determinat protocol que forma part del paquet (com per exemple "*frame*" -on hi són les dades genèriques del paquet, com la seva mida, etc- , "*eth*", "*arp*", "*ip*", *"icmp"*, "*tcp*", "*udp*","*tls*", "*data*" -que representen les dades en sí transportades-, etc) caldria afegir-li en canvi el paràmetre **-O protocol**). D'aquesta manera, si afegim per exemple *-O arp*, podríem comprovar els valors dels camps "Sender MAC Address", "Sender IP Address", "Target MAC Address" i "Target IP Address" de les peticions ARP mostrades, per exemple. Això també és vàlid pels apartats c) i d) . Si el que es vol, en canvi, és observar la representació hexadecimal del paquet, (és a dir, el mateix que mostra el panell de sota de la finestra del Wireshark) caldrà afegir llavors el paràmetre ***-x***

b) ¿Quina diferència veus a la sortida del terminal de la comanda *nping* si repeteixes la mateixa comanda que la de l'apartat anterior però sense indicar el paràmetre *-H*? ¿I si no indiques ara el paràmetre *-c 1*?

> NOTA: En teoria, un cop rebudes les respostes ARP, aquestes haurien de guardar-se a la catxé ARP del sistema (i, per tant, mostrar-se mitjançant la comanda *ip -c neigh show*) però nping no utilitza l'"stack" del sistema (va per lliure), amb la qual cosa la catxé ARP, al contrari del que semblaria el normal, no es veu alterada per les seves accions.

c) Mentre mantens la comanda *tshark* iniciada del primer apartat executa ara la comanda *sudo nping -c 1 -H --arp --arp-type arp-request 192.168.12.0/24* (on la IP indicada representa la IP de la teva xarxa local) ¿Què veus ara a la sortida del Tshark?

d) Atura el Tshark i torna'l a executar indicant ara com a filtre de pantalla aquest: *"eth.dst == MA:Cd:el:at:ar:ja and arp"* (on l'adreça MAC representa la de la tarja de la màquina virtual) mentre tornes a executar la mateixa comanda *nping* de l'apartat anterior ¿Quins paquets veus ara a la sortida del Tshark? ¿Quadra aquesta sortida amb el que veus a la sortida del Nping?

> NOTA: La comanda *nping* usada als apartats anteriors realitza un escaneig "actiu" però també hi ha la possibilitat de fer un escaneig "passiu". Una manera d'aconseguir-ho és simplement obrint el Wireshark i anant a l'opció Statistics --> "IPv4 statistics" --> "Source and destination addresses": al quadre mostrat apareixeran els ordinadors amb els quals s'ha intercanviat algun tipus de tràfic IP (això vol dir, però, que els ordinadors que hagin sigut descoberts via ARP però amb els què no s'hagi intercanviat encara cap paquet IP no apareixen; compte). Una manera similar d'obtenir aquesta informació amb Tshark seria amb la comanda ***tshark -q -z endpoints,ip***

Atac "ARP Flooding" (ACTUALMENT JA NO FUNCIONA)

Un atac de tipus "ARP Flooding" pot ser útil si l'atac "ARP Spoofing" no és efectiu (degut a què les màquines "víctimes" tinguin una taula ARP estàtica -cosa, per altra banda, molt poc habitual-). De totes formes, l'atac "ARP Flooding" avui dia precisament **és difícil que funcioni** perquè molts switchos actuals tenen mecanismes de protecció adients.

Tot i que ja es va estudiar a la teoria, a continuació es mostren uns petits apunts de recordatori sobre com funciona un atac "ARP Flooding". En tot cas, es pot consultar més informació sobre aquest tipus d'atacs a https://0xbharath.github.io/art-of-packet-crafting-with-scapy/network_attacks/cam_overflow/index.html :

> *El primer que fa un switch per saber a quina boca ha d'(re)enviar una determinada trama que li hagi arribat per una altra és buscar la direcció MAC de destí de la trama en qüestió dins de la seva taula CAM. Si aquesta apareix, enviarà la trama per la boca que indiqui la CAM. Si no, pot ser que hagi expirat l'entrada corresponent a la CAM o què encara no se li hagi enviat res (o no hagi enviat res) aquest equip desti en concret; en aquest cas, el switch enviarà la trama a totes les boques (excepte per on va entrar). D'aquesta manera, tots els equips rebran la trama; aquell equip la direcció MAC de la

qual coincideixi amb la MAC de destí de la trama contestarà i això permetrà al switch registrar una nova entrada a la CAM guardant l'associació d'aquesta MAC amb la boca per on s'ha rebut la resposta. Gràcies a això, el switch ja no haurà de tornar a "inundar" totes les boques amb futures trames dirigides a aquest equip.

*Si s'omple la CAM, tal com s'explica a teoria, el que pot passar és que les trames que tinguin una direcció MAC de destí no emmagatzemada a la CAM es retransmetran per totes les boques, comportant-se llavors el switch com un "hub". Això pot passar als switchos de gama baixa però com hem dit, els de gama mitja/alta inclouen mecanismes per mitigar l'atac (encara que, atenció, normalment no venen configurats per defecte!). L'efecte de convertir un switch en un "hub", tal com ja sabem, és que l'atacant pugui connectar-se a qualsevol boca del switch i **capturar tràfic que no rebria en circumstàncies normals** (a més de provocar també un possible "DoS").

1.- Instal·la a la mateixa màquina virtual de l'exercici anterior el paquet "dsniff" (dins del qual ens trobarem la comanda "macof"). Obre el Wireshark i comença a escoltar per la tarja enp0s3; tot seguit executa la comanda següent: *sudo macof -i enp0s3 -n 10000*. ¿Què veus a la finestra del Wireshark? ¿Quines són les adreces IP d'origen i de destí dels paquets generats i per què? ¿Com podries esbrinar, mitjançant el mateix Wireshark, si aquest atac ha funcionat (no ho sol fer)?

Atac "ARP Spoofing"

Per començar, a la màquina atacant hem d'habilitar una opció del kernel anomenada "ip-forward", la qual permet que el sistema pugui reenviar paquets IP entre les seves diverses tarjes de xarxa (de manera que el tràfic que es rebi per una es pugui transmetre a l'altra per sortir-ne i viceversa). Per defecte aquesta opció està deshabilitada perquè per en condicions normals un sistema Linux no necessita funcionar com a reenviador ("router").

Necessitem tenir l'"IP-Forwarding" activat a la màquina atacant per poder fer passar el tràfic de la víctima a la porta d'enllaç i viceversa (i així aconseguir que la víctima pugui continuar navegar per Internet sense notar res); si no l'habilitéssim, la connexió a Internet es tallaria perquè el tràfic originat a la víctima no podria "rebotar" i arribar a la porta d'enllaç legítima i viceversa, ja que es "quedaria" sempre bloquejat a la màquina atacant.

1.-a) Tria alguna de les següents formes diferents d'habilitar l'"IP-Forwarding" i implementa-la:

NOTA: En qualsevol cas, el valor "1" a tots els exemples següents equival a "activar" però es podria fer el contrari, és a dir, "desactivar" l'"IP-Forwarding" executant exactament el mateix però canviant el valor en qüestió per "0".

* De forma no permanent (fins el següent reinici del sistema), amb comandes genèriques:
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
* De forma no permanent (fins el següent reinici del sistema), amb una comanda específica:
sudo sysctl net.ipv4.ip_forward=1

NOTA: La comanda *sysctl* té altres opcions interessants, com ara consultar el valor actual d'alguna opció, així: *sysctl net.ipv4.ip_forward* o *sysctl -n net.ipv4.ip_forward* , o consultar-ne el de totes les opcions existents, així: *sysctl -a* , etc

* De forma permanent: cal especificar el valor 1 a la línia "*net.ipv4.ip_forward=*" de l'arxiu "*/etc/sysctl.conf*" (o, depenent de la distribució, "*/etc/sysctl.d/00-sysctl.conf*") i reiniciar el sistema per començar a aplicar el canvi (o bé executar la comanda *sudo sysctl -p* per aplicar-lo immediatament sense haver de reiniciar).

NOTA: L'"ip-forward" permanent es pot implementar, alternativament, afegint la línia *IPForward=yes* a l'arxiu "*.network" corresponent a qualsevol de les tarjes de xarxa gestionades per systemd-networkd (amb una sola que s'indiqui ja funciona per totes) En realitat, el que fa aquesta línia és executar la comanda *echo 1 > /proc/sys/net/ipv4/ip_forward* automàticament cada cop que la tarja en qüestió s'activi
NOTA: En el cas d'habilitar l'"ip-forward" per IPv6, l'opció a habilitar és "net.ipv6.conf.all.forwarding"

A partir d'ara suposarem la configuració següent en les tres màquines involucrades en l'atac "ARP Spoofing" que implementarem al proper exercici (és a dir, "router", màquina atacant i màquina víctima):

* La porta d'enllaç per defecte de la nostra LAN té la IP **192.168.12.10** (sempre podrem comprovar quina és observant la sortida de la comanda *ip -c route show*) i la MAC **00:ab:cd:ef:12:10** (sempre podrem comprovar quina és a partir de saber la seva IP i observant llavors la sortida de la comanda *ip -c neigh show*)

* La màquina virtual utilitzada als exercicis anteriors (i que utilitzarem ara com a sistema "atacant") té la IP **192.168.12.111** (sempre podrem comprovar quina és observant la sortida de la comanda *ip -c address show* executada en el propi sistema) i la MAC **00:ab:cd:ef:12:11** (sempre podrem comprovar quina és amb la mateixa comanda anterior o bé, com ja hem fet al primer exercici d'aquest document, observant la sortida de la comanda *ip -c link show*)

* La màquina real (que serà el sistema "víctima") té la IP **192.168.12.222** (aquesta dada l'hem de triar nosaltres segons la màquina concreta que haguem decidit atacar, a partir, per exemple, d'haver fet un escaneig previ amb *arp-scan*, *nmap* o *nping*, tal com s'ha descrit als exercicis anteriors) i la MAC **00:ab:cd:ef:12:22** (que podrem esbrinar, a partir de saber la seva IP i d'haver-nos comunicat prèviament d'alguna manera amb ella -amb un simple "ping", per exemple- amb la comanda *ip -c neigh show*)

Un cop tenim clares les dades anteriors, passarem a implementar l'atac "ARP Spoofing" (és a dir, l'"enverinament de les catxés ARP tant de l'enrutador com de la màquina "víctima") pas a pas amb la comanda *nping* Per fer-ho, cal generar respostes ARP (sense que hi hagi cap petició prèvia!), i això es fa indicant el valor *arp-reply* al seu paràmetre *--arp-type*. En aquest cas, la IP indicada al final de la comanda *nping* haurà de ser la IP concreta a la què volguem enviar la resposta en qüestió (és a dir, la víctima); recordeu que les respostes ARP són "unicast" (a diferència de les peticions ARP, que són "broadcast")

NOTA: Indicar el valor de cadena "arp-request" al paràmetre *--arp-type* de la comanda *nping* és equivalent a indicar el valor numèric 1 (l'estàndard oficial) i indicar el valor de cadena "arp-reply" és equivalent a indicar el valor numèric 2.

Però aquestes respostes ARP que s'enviaran (tant a la màquina víctima com al "router") han d'estar contenir uns valors determinats per a què l'atac tingui èxit. En concret, cal saber que els paquets ARP contenen, entre altres, quatre camps importants a la seva capçalera, el valor dels quals els haurem d'establir "a mà" mitjançant els següents paràmetres de la comanda *nping*:

--arp-sender-mac di:rM:AC	Estableix la MAC del remitent del paquet (en una resposta ARP, aquest valor és justament la dada demanada en la petició) Aquest valor l'haurem de definir "a mà" en un atac d'"spoofing"
--arp-sender-ip dir.IP	Estableix la IP del remitent del paquet Aquest valor l'haurem de definir "a mà" en un atac d'"spoofing"
--arp-target-mac di:rM:AC	En una petició ARP, aquest valor és sempre *ff:ff:ff:ff:ff:ff* En una resposta ARP aquest valor és la MAC de qui va fer la petició Aquest valor és irrellevant en un atac d'"spoofing"
--arp-target-ip dir.IP	En una petició ARP, és la IP de la qual es vol saber la MAC associada En una resposta ARP, és la IP de qui va fer la petició Aquest valor és irrellevant en un atac d'"spoofing"

1BIS.-a) Procedirem ara a realitzar l'atac pròpiament dit, enganyant primer a la víctima (l'ordre tant és) per fer-li creure que nosaltres som la seva porta d'enllaç. Per aconseguir això, li hem d'enviar contínuament (per a què la seva taula ARP estigui permanentment "enverinada") respostes ARP (no sol.licitades) contenint la informació enganyosa pertinent. Concretament, la resposta ARP a enviar ha de contenir els següents valors:

IP de destí:	"192.168.12.222"	(IP de la víctima -és a dir,la màquina real-; és on va dirigida la resposta)
MAC de destí:	"00:ab:cd:ef:12:22"	(MAC de la víctima; és on va dirigida la resposta)
IP d'origen:	"192.168.12.10"	(**IP de la porta d'enllaç**; ens estem fent passar per ella)
MAC d'origen:	"00:ab:cd:ef:12:11"	(**MAC de l'atacant**; la discrepància amb la IP anterior és l'atac en sí)

Per tant, executa (millor en un bucle infinit dins d'un shell script) la següent comanda:

sudo nping -c 9999 --arp --arp-type arp-reply --arp-sender-ip 192.168.12.10 --arp-sender-mac 00:ab:cd:ef:12:11 192.168.12.222

b) A continuació hem d'enganyar la porta d'enllaç legítima fent-li creure que nosaltres som la màquina víctima. Per això, li hem d'enviar contínuament (per a què la seva taula ARP estigui permanentment "enverinada") respostes ARP (no sol.licitades) contenint la informació enganyosa pertinent. Concretament, la resposta ARP a enviar ha de contenir els següents valors:

IP de destí:	"192.168.12.10"	(IP de la porta d'enllaç; és on va dirigida la resposta)
MAC de destí:	"00:ab:cd:ef:12:10"	(MAC de la porta d'enllaç; és on va dirigida la resposta)
IP d'origen:	"192.168.12.222"	(**IP de la víctima**; ens estem fent passar per ella)
MAC d'origen:	"00:ab:cd:ef:12:11"	(**MAC de l'atacant**; la discrepància amb la IP és l'atac en sí)

Per tant, executa (millor en un bucle infinit dins d'un shell script, en un altre terminal diferent de l'usat a l'apartat anterior, o bé executant ambdós en segon pla) la següent comanda:

sudo nping -c 9999 --arp --arp-type arp-reply --arp-sender-ip 192.168.12.222 --arp-sender-mac 00:ab:cd:ef:12:11 192.168.12.10

c) Per comprovar si, efectivament, l'atac està tenint èxit, executa la comanda *ip -c neigh show* a la màquina víctima (és a dir, la màquina real) i comprova que aparegui l'associació "enverinada" 192.168.12.10<->00:ab:cd:ef:12:11 (a la porta d'enllaç també hi hauria d'haver l'associació "enverinada" corresponent però això no ho podem comprovar)

La manera més "pràctica", però, de comprovar si l'atac està funcionant és observar si, efectivament, el tràfic originat en qualsevol dels dos extrems travessa el sistema atacant i pot ser inspeccionat per aquest (si no està xifrat). Això pots fer-ho, per exemple, enviant des de la màquina víctima algun "ping" a alguna adreça IP d'Internet qualsevol, o també realitzant alguna consulta DNS (així, per exemple *dig www.marca.com*)...o generant qualsevol altre tipus de tràfic no xifrat (en aquest sentit, si, en canvi, navegues per Internet, el tràfic observat no tindrà gaire sentit aparentment perquè serà HTTPS -tot i que aquest "inconvenient" el solucionarem en propers PDFs quan veiem el concepte de certificats TLS-). En tot cas, s'ha d'observar que la màquina víctima obté la resposta desitjada (i per tant, l'atac està passant desapercebut) i que, al mateix temps, la màquina atacant ha pogut capturar els paquets implicats en la conversa (si s'ha fet un "ping", això vol dir tant els paquets "echo-request" com els "echo-reply" i si s'ha fet una petició DNS -no xifrada, que és avui dia el més normal-, tant aquesta com la seva resposta, etc).

d) Executa a la màquina de l'atacant (en un altre terminal virtual diferent mentre les instàncies Nping estan funcionant) la comanda *tshark -Y "icmp"* (per inspeccionar el tràfic ICMP, generat per exemple per la comanda *ping*) o bé *tshark -Y "dns"* (per inspeccionar el tràfic DNS, generat per exemple per la comanda *dig*) o bé tshark -Y "http" (per inspeccionar el tràfic HTTP sense xifrar, generat per exemple per la comanda *curl*), entre qualsevol altre de possibles exemples de tràfic, i observa com aquest tipus de tràfic en qüestió es pot visualitzar a la sortida del Tshark (hauràs de generar-lo a la víctima per veure-ho, és clar)

> NOTA: Per finalitzar l'atac, normalment és suficient aturant els bucles posats en marxa a les instàncies Nping. No obstant, és recomanable re-establir explícitament els valors adients a la catxé ARP tant de la víctima com de la porta d'enllaç. Això es pot fer enviant una resposta ARP a cada extrem amb els valors vàlids.

e) ¿Com hauries de modificar l'atac "ARP Spoof" anterior per a què es pogués realitzar no contra una sola víctima només sinó contra totes les màquines detectades prèviament en un escaneig ARP com el realitzat al primer exercici d'aquest document?

Un atac que es podria realitzar un cop implementat l'"ARP Spoofing" és, per exemple, un de tipus **DoS** en el qual la víctima deixés de tenir Internet (ja que ara l'enrutador seria l'atacant i aquest podria habilitar i deshabilitar l'"ip-forward" a consciència)

2.-a) Torna a implementar l'atac "ARP Spoof" descrit a l'exercici anterior i comprova que la víctima té connexió a Internet (per exemple, navegant o fent un "ping" infinit a alguna IP pública d'Internet com per exemple 1.1.1.1 o 8.8.8.8) sense problema. Tot seguit, deshabilita temporalment l'"ip-forward" de la màquina atacant i comprova que la víctima deixa de tenir connexió. Finalment, habilita de nou l'"ip-forward" i comprova que la víctima tingui de nou connexió a Internet.

b) ¿Com creus que podries programar talls d'Internet periòdics (per exemple, cada 5 minuts)?

Un altre atac que es podria realitzar un cop implementat l'"ARP Spoofing" (i que desenvoluparem amb més detall en propers PDFs) és el "**DNS Spoofing**", el qual consisteix bàsicament en fer creure a la víctima que quan accedeix a un servidor remot mitjançant el seu nom DNS hi està accedint de la forma habitual però on, en realitat, està connectant amb un servidor "impostor". És a dir, consisteix en enverinar la resolució de noms (això és possible perquè a dia d'avui aquest tipus de tràfic no viatja xifrat ni signat!) de tal forma que el nom DNS desitjant no apunti a l'adreça IP legítima sinó a una altra impostora. Si aquest servidor "impostor" implementa, a més, el mateix tipus de servei que el legítim (un servidor web, per exemple, amb les pàgines clonades per a què semblin les mateixes), l'engany ja serà complet. Això és el que se sol anomenat "**phishing**", ja que l'engany "pesca" a la màquina víctima i d'aquesta forma, l'usuari enganyat pot començar a aportar dades personals -com el número de tarjeta de crèdit, contrasenyes, etc- de forma voluntària pensant-se que està en un lloc web (d'un banc, d'una botiga, etc) que no és el de veritat.

Per poder implementar un atac "DNS Spoofing" necessitem posar en marxa un servidor DNS que contesti "malament". El servidor DNS més senzill de fer funcionar és el dimoni "Systemd-resolved" (el qual, a més, ja ve "de sèrie" a la majoria de sistemes); només caldrà configurar-lo, això sí, per a què escolti peticions provinents de l'exterior i per a què actuï com a servidor autoritatiu fent servir com a "arxiu de zona" el contingut de l'arxiu "/etc/hosts". Concretament:

*Per a què escolti peticions de clients DNS remots (per defecte "Systemd-resolved" només ho fa de clients funcionant a la pròpia màquina) cal indicar a l'arxiu "/etc/systemd/resolved.conf" la línia (o línies, perquè n'hi pot haver més d'una): *DNSStubListenerExtra=ip.tarja:nºport* (on "ip.tarja" representa la IP de la tarja de xarxa de la màquina per on es voldran rebre peticions remotes -en aquest cas, estem parlant de la IP de la màquina de l'atacant, tot i que pot indicar-se genèricament "0.0.0.0"-; i "nºport és el port d'escolta -si no s'indica, per defecte serà el 53 tcp i udp-).

*Per a què faci de servidor DNS autoritatiu del contingut present a l'arxiu "/etc/hosts" local per tots els clients admesos (el local i/o els remots), cal indicar a l'arxiu "/etc/systemd/resolved.conf" la línia *ReadEtcHosts=yes* (per defecte ja és així)

En tot cas, un cop fetes les modificacions anteriors, caldrà reiniciar el dimoni executant la comanda *sudo systemctl restart systemd-resolved*

NOTA: Un altre programa que podria servir com a servidor DNS senzill, tant de tipus cau com autoritatiu a partir del contingut de l'arxiu "/etc/hosts" és Dnsmasq (http://www.thekelleys.org.uk/dnsmasq/doc.html)

A més a més, haurem de configurar el tallafocs de la màquina atacant per a què tot el tràfic detectat que vagi dirigit al port 53 sigui redirigit al servidor DNS local i no pas a Internet (degut a l'IP-Forwarding). És a dir, no modificarem pas la configuració de servidors DNS utilitzats per la víctima sinó que aprofitarem que tot el seu tràfic passa ara per nosaltres per simplement desviar les seves peticions DNS al nostre propi servidor DNS (independentment d'on anaven dirigides originalment).

3.-a) Configura el dimoni "Systemd-resolved" de la màquina atacant, tal com s'explica als paràgrafs blaus anteriors, de manera que funcioni com a servidor autoritatiu (de l'arxiu de zona "/etc/hosts") tant per peticions locals com remotes (compte perquè fent això també t'estaràs "autoatacant").

b) Edita l'arxiu "/etc/hosts" de la màquina atacant per tal d'indicar els noms DNS que vols "spofejar", indicant que es resoldran a la IP de la pròpia màquina atacant (podria ser també la IP d'un altre servidor sota control nostre). Per exemple:

192.168.12.111 www.google.com
192.168.12.111 www.hola.com

c) Crea un fitxer de text (que anomenarem "redir53.txt") que tingui les regles del tallafocs de Linux necessàries per tal de redirigir tot el tràfic destinat al port 53 UDP que provingui de qualsevol víctima externa al nostre propi servidor DNS (en comptes de reenviar-ho als servidors DNS d'Internet que tingui la víctima en qüestió, que és el que faria gràcies a l'"IP-Forwarding"). Concretament, el contingut d'aquest fitxer ha de ser el següent...

NOTA: Una altra opció hagués sigut interceptar les respostes dels servidors DNS legítims i modificar-les "al vol", però és una solució molt més complicada i no aporta cap avantatge.

```
#!/usr/sbin/nft -f
flush ruleset
table ip nat {
        chain prerouting {
                type nat hook prerouting priority 100 ;
                iifname enp0s3 udp dport 53 redirect to 53
        }
}
```

...i, un cop gravat, cal aplicar les regles allà escrites executant la següent comanda: *sudo nft -f redir53.txt*

d) Reinicia el servei *systemd-resolved* i, instal·la, a més, un servidor Apache2 (paquet "httpd" a Fedora, "apache2" a Ubuntu) i un servidor OpenSSH (paquet "openssh-server") en la màquina atacant. Confirma que els tres serveis estiguin encesos (fent *systemctl status systemd-resolved sshd httpd/apache2*)

e) Torna a implementar l'atac d'"ARP Spoofing" tal com es descriu a l'exercici 1 i 1BIS. Amb aquest atac en marxa i el servidor "Systemd-resolved" rebent les peticions DNS de les víctimes, la màquina atacant haurà de poder oferir les resolucions de noms indicades al fitxer "/etc/hosts". Això ho pots comprovar executant, a la màquina víctima, les següents comprovacions:

* La comanda *ping www.hola.com* ¿Qui et respon?
* La comanda *resolvectl query www.hola.com* (o també *dig www.hola.com*) ¿Què et mostra?
* Escrivint l'adreça http://www.hola.com al navegador de la màquina víctima. ¿Què veus i per què?
* La comanda *ssh usuari@www.hola.com* ¿Què passa? En aquest sentit, ¿quina importància té, en relació a adonar-se de què "passa alguna cosa", la pregunta inicial que mostra el "fingerprint" de la clau pública del servidor?