Tshark + Opensearch/Dashboards

Tshark

Tshark és la versió en línia de comandes de Wireshark (a Ubuntu s'instal.la mitjançant el paquet homònim, a Fedora s'instal.la amb el paquet "wireshark-cli"). Per fer-lo servir, primer s´ha d´executar amb el paràmetre *-D* per llistar les interfícies disponibles, i seguidament, per començar a capturar, s´ha d´escriure la comanda amb el paràmetre *-i* seguit del número d´interfície observat abans amb *-D* (també es pot indicar com a valor de *-i* directament el nom de la tarja a fer servir per capturar)

*Per especificar filtres de captura cal fer-ho amb el paràmetre *-f "filtreCaptura"*.
*Per especificar filtres de pantalla cal fer-ho amb el paràmetre *-Y "filtrePantalla"*
*Per obrir un fitxer amb paquets capturats s'usa el paràmetre *-r nomFitxer*
*Per guardar els paquets capturats en un fitxer s'usa el paràmetre *-w nomFitxer* (llavors no es veurà res en pantalla, a no ser que s'indiqui també el paràmetre *-P*).

NOTA: Si no s'especifica el paràmetre *-w* anterior, Tshark per defecte guardarà tots els paquets capturats a "/tmp", carpeta que en els sistemes Linux moderns sol estar muntada directament sobre la RAM (concretament, en el sistema de fitxers "tmpfs"). Aquest comportament és idèntic a Wireshark fins que no se li indica si es vol guardar la captura en un fitxer "pcapng" (això es pot comprovar, al Wireshark, anant a la pestanya "Folders" del quadre que apareix en anar a l'opció del menú "Help->About Wireshark" i veient el valor de l'apartat "Temp"). En teoria, la ruta per defecte es pot canviar si prèviament a l'execució de Tshark/Wireshark s'estableix una nova ruta com valor de la variable TMPDIR (o TEMP o TMP), però quan s'ha provat aquest aspecte no ha funcionat

Per defecte Tshark mostra una línia per paquet detectat amb les mateixes columnes que les mostrades per defecte a la pantalla principal del Wireshark; és a dir:

nº | temps | IP origen | IP destí | protocol | longitud paquet | informació (com port origen i destí...)

Aquest format es pot personalitzar amb els següents paràmetres:

-T {fields|json|ps|psml|pdml|text|ek} : Indica el format de sortida de les dades: camps específics (cal fer servir a més el paràmetre *-e* per indicar-los, fent servir notació de filtres de pantalla), JSON, Postcript, XML ("psml" o "pdml"), text (per defecte) o preparat per enviar a un servidor OpenSearch, entre d'altres.

-e nom.camp1 -e nom.Camp2 : Si s'indica els valors "fields", "json" o "ek" al paràmetre *-T* anterior, aquest paràmetre cal escriure'l per especificar (fent servir notació de filtres de pantalla) cada camp que es vol veure. Per exemple, si es vol la IP d'origen d'un paquet s'escriurà el paràmetre *-e ip.src*

NOTA: Les columnes que proporciona el propi Wireshark com a comoditat de l'usuari però que no es corresponen a cap camp concret dels paquets es poden especificar també com a valors vàlids del paràmetre *-e* simplement indicant el seu nom precedit del prefix "**_ws.col.**" Per veure tots els filtres possibles que ofereix *tshark* incloent aquests "especials" (llistats a https://www.wireshark.org/docs/dfref/_/_ws.col.html) , es pot fer *tshark -G fields*

-E headers=y/n : Indica si s'imprimeix la capçalera dels camps indicats amb *-e*
-E separators={ /t | /s | caracter }: Indica si el separador dels camps indicats amb *-e* és el tabulador, l'espai en blanc o un altre caracter a indicar.

Per personalitzar la sortida de Tshark també es pot fer servir el paràmetre *-o* (el qual, en general serveix per establir una determinada preferència de configuració del programa, amb el format *-o nomPreferencia:valor*). Més en concret, així:

-o column.format:""nomColumna1","%format","nomColumna2","%format",..."

on "*%format*" pot ser algun d'aquests valors (entre altres): "*%m*" (Nº paquet), "*%t*" (Temps), "*%s*" (IP origen), "*%d*" (IP destí), "*%p*" (Protocol) o "*%i*" (Informació). D'altra banda, a més a més d'aquests formats (que ja es veu que són pocs) es pot utilitzar el valor *"%Cus:filtrePantalla"* per definir exactament el contingut que es vol mostrar a una columna. Per exemple, si volguéssim veure una columna amb el valor TTL dels paquets capturats (a més de la IP d'origen i de destí), podríem fer:

-o column.format:""IP Origen","%s","IP Destí","%d","TTL","%Cus:ip.ttl""

Altres paràmetres interessants de Tshark són:

-n : Deshabilita la resolució de noms (de hosts i de ports)
--color : Mostra els paquets en colors segons el seu tipus
-l : Mostra a pantalla el paquet rebut tot just ha sigut interpretat per *tshark* (és a dir, sense esperar a què s'ompli el seu "buffer" intern de sortida); aquesta opció és útil quan es vol encanonar la sortida de *tshark* amb una altra comanda posterior

-V : Mostra per cada paquet la seva dissecció, similar al panell intermig de Wireshark. Una altra opció similar és *-O protocol,unaltre,...*, com ara *-O icmp* (en aquest cas, però, només es mostra la dissecció dels paquets pertanyents al/s protocol/s indicat/s i no de tots com *-V*)

> NOTA: Per saber els protocols que es poden indicar amb *-O* es pot fer *tshark -G protocols*

-x : Mostra directament en hexadecimal el contingut de cada paquet (i la seva correspondència en ASCII), similar al panell inferior de Wireshark

-q : No mostra per pantalla els paquets capturats en temps real (útil per si només volem realitzar estadístiques amb les dades finals o les volem guardar en un fitxer sense "embrutar" la pantalla). Si no volem veure ni tan sols les estadístiques finals, usar *-Q*

-t {a|ad|u|ud|r|d|dd} : Especifica el format de la columna de temps del paquet capturat. "**a**" indica temps absolut local, "**ad**" indica el mateix però afegint la data, "**u**" indica temps absolut UTC, "**ud**" indica el mateix però afegint la data, "**r**" indica el temps relatiu (és a dir, la diferència) entre el temps del primer paquet capturat i l'actual, "**d**" indica el temps delta (és a dir, la diferència) entre el temps del paquet capturat anterior i l'actual, "**dd**" indica el temps delta entre el temps del paquet mostrat anterior i l'actual. El valor per defecte és "r".

-c nº : Para la captura quan s'hagin capturat nº paquets
-a duration:nº : Para la captura quan hagin passat nº segons
-a filesize:nº : Para la captura quan s'hagi arribat a nºKB de paquets capturats
-a packets:nº : Para la captura quan s'hagi arribat al número de paquets visualitzats indicat

-b files:nº : Grava la captura en com a màxim el nombre de fitxers indicats, de forma rotativa (a combinar amb el paràmetre *-w*, que servirà per indicar el nom-base de cadascun d'aquests fitxers, i amb alguna de les següents opcions, que serviran per indicar el criteri per passar la gravació d'un fitxer a un altre):

-b duration:nº : Passa a gravar la captura al següent fitxer quan hagin passat nº segons
-b filesize:nº : Passa a gravar la captura quan s'hagi arribat a nºKB de paquets capturats
-b packets:nº : Passa a gravar la captura quan s'hagi arribat al nº de paquets capturats indicat

> NOTA: Les opcions anteriors permeten mantenir de forma infinita una captura mitjançant Tshark sense col·lapsar l'espai de disc ni la memòria RAM

-s nº : Indica la quantitat de bytes a capturar de cada paquet (començant pel principi). Aquesta opció seria equivalent a la del menú "Capture->Options->Limit each packet n bytes" del Wireshark. Si nº fos **14** només veuríem la capçalera Ethernet de cada paquet; si fos **34**, fins la capçalera IP; si fos **42** fins la capçalera UDP i si fos **54** fins la TCP.

-v : Mostra la versió del programa i les opcions de compilació

El Tshark també permet obtenir gràfiques i estadístiques tal com ja hem vist amb el seu "germà gran" gràfic, el Wireshark. Concretament, els següents paràmetres es poden fer servir:

-z io,stat,nºsec,*filtre,filtre,...* **:** Al final de la captura mostra la quantitat total de paquets i bits capturats per cada interval de temps (l'indicat, en segons), aplicant-hi opcionalment varis filtres de pantalla.

> NOTA: Com a filtre també es poden usar funcions "d'agregació" d'aquesta manera: *"AVG(capçalera.camp)capçalera.camp","MIN(capçalera.camp)capçalera.camp","MAX(capçalera.camp) capçalera.camp","COUNT(capçalera.camp)capçalera.camp"* o *"SUM(capçalera.camp)capçalera.camp"* sempre que *capçalera.camp* contingui un valor numèric. Per exemple: *-z io,stat,1,MIN(ip.ttl)ip.ttl*

-z io,phs,*filtre* : Al final de la captura mostra la jerarquia de protocols capturats, indicant per cadascun la quantitat total de paquets i bits capturats. Es pot indicar un filtre de pantalla.

-z conv,tipus,*filtre* : Al final de la captura mostra la llista de "conversacions" entre els hosts (IPs dels extrems, quantitat de paquets i bytes rebuts, transmesos i total, duració de la conversa i el seu ordre relatiu, etc). El "tipus" pot ser, entre altres: ***eth, ip, tcp*** o ***udp.***

-z endpoints,tipus,*filtre* : Al final de la captura mostra la llista d'"endpoints" entre els hosts (IPs dels extrems, quantitat de paquets i bytes rebuts, transmesos i total, duració de la conversa i el seu ordre relatiu, etc). El "tipus" pot ser, entre altres: ***eth, ip, tcp*** o ***udp***.

-z dests,tree,*filtre* : Mostra informació estadística (número de paquets, mitjana, etc) sobre les IPs, protocols de transport i ports de destí existents a la captura
-z ptype,tree : Mostra informació estadística sobre els protocols IPs existents a la captura
-z ip_hosts,tree : Mostra informació estadística sobre els hosts que intervenen a la captura

-z dhcp,stat*,filtre* : Ofereixen informes estadístics dels paquets DHCP capturats
-z dns,tree*,filtre* : Ofereixen informes estadístics dels paquets DNS capturats
-z http,tree,*filtre* : Ofereixen informes estadístics de transaccions HTTP, codis de resposta, enllaços trencats, redireccions, etc. També es poden veure només les estadístiques de peticions (***-z http_req,****filtre*) o de respostes (***-z http_srv,****filtre*).

-z follow,tcp,{hex|ascii|raw},ipOrigen:port,ipDesti:port : Al final de la captura mostra la conversa TCP entre els dos extrems indicats. També es pot mostrar una conversa indicant el seu número (nº) mitjançant ***-z follow,tcp,{hex|ascii|raw},nº***

EXERCICIS

1.-a) Utilitza (a la màquina real) els paràmetres adeqüats del Tshark per capturar durant 10 segons (*-a duration:10*) només els paquets UDP que surtin de la teva màquina amb destí al port 53 (*-f "src host la.te.va.IP and udp dst port 53"*) però mostrant només per pantalla les direccions IP de destí d'aquests paquets capturats (*-T fields -e ip.dst*). Executa en un altre terminal durant aquests deu segons la comand *dig* un parell de cops per tal de generar algun tràfic que coincideixi amb el filtre indicat. Un cop finalitzada la captura, obre aquest fitxer amb el Wireshark. ¿Què veus? ¿Quina relació té el que veus amb la sortida de la comanda *resolvectl dns*?

b) Obre un terminal i executa la comanda *ping 8.8.8.8* . Obre un altre terminal i executa la comanda *tshark -f "icmp" -T fields -e frame.number -e frame.time_relative -e ip.src -e ip.dst -e ip.ttl -e _ws.col.Protocol -e _ws.col.Info* ¿Què veus en cada columna de les files que van apareixent?

c) Sabent que les diferents "opcions" d'un missatge DHCP de resposta indiquen els diferents valors que un servidor DHCP pot assignar a un client, ¿què és el que veus si executes aquesta comanda: *tshark -n -Y "dhcp" -T fields -e dhcp.option.domain_name_server -e dhcp.option.router -e eth.src* ? (pots provar de provocar tràfic DHCP executant en un altre terminal les comandes *nmcli con down nomConnexioActual && nmcli con up nomConnexioActual*)

d) ¿Quin és el significat de la informació que es mostra a pantalla en executar aquesta comanda: *tshark -n -T fields -E separator=';' -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -Y "(tcp.flags.syn == 1 and tcp.flags.ack == 0)"* ? (pots provar de provocar tràfic simplement obrint el navegador, per exemple)

e) Digues què significa cadascuna de les columnes mostrades per la comanda següent i prova-la: *tshark -Y "ip.src==la.te.va.IP" -T fields -e tcp.stream -e tcp.seq -e tcp.ack -e tcp.flags.str -e tcp.flags* (pots provar de provocar tràfic simplement obrint el navegador, per exemple)

f) ¿Quin és el significat de la informació que es mostra a pantalla en executar cadascuna de les següents comandes (i navegar mentrestant per Internet)?

tshark -Y "http.request.method == GET" -T fields -e http.request.uri
tshark -Y "http.request.method == POST" -T fields -e http.file_data
tshark -Y "http.response.code == 200" -T fields -e http.file_data

> NOTA: Per saber més valors possibles a indicar amb el paràmetre *-e* es pot anar sel.leccionant, a la zona central del Wireshark, les parts del paquet (HTTP o de qualsevol altre tipus) que desitjem, i fixar-nos en el valor escrit entre parèntesis que apareixerà a l'esquerra de la seva barra d'estat

g) ¿Quin tipus d'informació observes si, mentre vas navegant per Internet, executes la comanda *tshark -O dns*? ¿O la comanda *tshark -O tls*? ¿O la comanda *tshark -O data*? ¿O la comanda *tshark -O tcp*? ¿O la comanda *tshark -O ip*? ¿O la comanda *tshark -O eth*? ¿O la comanda *tshark -O none*? ¿Quina diferència hi ha entre les comandes anteriors i la comanda *tshark -V*? ¿Què mostra, d'altra banda, la comanda *tshark -x*? ¿I la comanda *tshark -x -O none*?

2.-a) Digues quines dades apareixen per pantalla en finalitzar l'execució de les següents comandes (també es podrien usar sobre la informació obtinguda en un fitxer de captura prèviament generat):

tshark -q -z io,phs
tshark -q -z io,stat,20,ip.addr==la.te.va.IP
tshark -q -z io,stat,30,"COUNT(tcp.analysis.retransmission) tcp.analysis.retransmission"
tshark -q -z conv,tcp
tshark -q -z conv,tcp,http
tshark -q -z dns,tree -z http,tree

b) Executa la comanda *tshark -a packets:1000 -w captura.pcap* i tot seguit obre un navegador per generar tràfic. Un cop hagi finalitzat la comanda anterior, prova el següent script. ¿Què fa?

```
#!/bin/bash
for stream in $(tshark -r captura.pcap -T fields -e tcp.stream | sort -n | uniq) ; do
        echo $stream &&  tshark -r captura.pcap -w stream-$stream.pcap -Y "tcp.stream==$stream"
done
```

bII) De forma similar, digues per a què serviria aquest bucle:
for acapfile in (.pcap) do tshark -r $acapfile -Y "dns" -w $acapfile-dns.pcap*

bIII) ¿I aquesta comanda?:
tshark -r captura.pcap -Y "frame.time_relative >= 5 && frame.time_relative <= 7 " -w captura-cut.pcap

c) Executa la comanda *tshark -nr Conversa_HTTP_completa.pcap --export-objects http,/home/usuari* (l'arxiu "pcap" ha de ser el mateix que l'utilitzat a l'exercici 6 del PDF sobre Wireshark). ¿Què passa?

d) Després de llegir la teoria (o la pàgina de manual de *tshark*) per tal d'esbrinar per a què serveix el paràmetre *-b files:nº* usat conjuntament amb el paràmetre *-b filesize:nº* (i *-w nomfitxer.pcap*), prova'l.

dII) Esbrina també per a què serveix el paràmetre *-b files:nº* igual però ara usat conjuntament amb el paràmetre *-b duration:nº* (i *-w nomfitxer.pcap*) i prova'ls també.

e) Consulta el manual de *tshark* i digues per a què serveix el seu paràmetre *-d*

f) ¿Per a què serveix el paràmetre *-o* de tshark i quina funció concreta creus que tindria en aquesta comanda? (recorda la utilitat i el valor de la variable SSLKEYLOGFILE vist en el PDF sobre Wireshark):
tshark -o "tls.keylog_file:sslkeys.txt" -x -Y "http.content"

OpenSearch/Dashboards

Molt sovint necessitarem tractar visualment els paquets detectats per Tshark (o Wireshark) per poder tenir accés més fàcilment a diverses dades estadístiques i/o agregades relacionades amb el comportament general del trànsit detectat (com ara l'evolució de l'ample de banda utilitzat al llarg del temps, o el percentatge d'ús d'un determinat protocol respecte el total de trànsit, o la comparativa de nombre d'adreces IP d'origen o de destí respecte el total, etc, etc, etc); és a dir, per poder analitzar, ja sigui en temps real o bé de forma històrica, el contingut i les metadades dels diferents paquets detectats, dades que pels administradors de xarxa o els experts en ciberseguretat poden ser molt valuoses a l'hora de permetre monitoritzar eventuals anomalies de tràfic, problemes de connectivitat i/o atacs als sistemes.

Una opció per fer tot aquest anàlisi visual i estadístic seria reenviar el trànsit detectat per Tshark a un servidor OpenSearch per tal d'indexar-lo allà (és a dir, guardar-lo de forma permanent i optimitzar-lo per fer-hi recerques) i, a partir d'aquí, estudiar-lo gràficament amb els panells web de Dashboards. Expliquem què són i què fan aquests dos programes:

*"**OpenSearch**" (https://opensearch.org/docs/latest/about): Aquest programa serveix per emmagatzemar en disc dades en format JSON; cada JSON guardat s'anomena "document" i seria l'equivalent a una fila d'una taula en una base de dades (però sense que la seva estructura de camps sigui rígida ni estigui predefinida, o gairebé); els documents, per la seva banda, s'agrupen en conjunts de documents anomenats "índexs" (que serien similars a les "taules" d'una base de dades). Els documents se solen rebre provinents d'orígens remots (és a dir, el servidor OpenSearch escolta per la xarxa, concretament al port 9200, l'entrada de documents) fent servir una API REST. OpenSearch està optimitzat per gestionar cadenes (per tant, és una sol·lució ideal per gestionar logs) i la seva especialitat és la de realitzar recerques (d'aquí el seu nom) perquè indexa tot el contingut que guarda i, a partir d'aquí, també en pot fer agregacions (és a dir, càlculs estadístics agrupats, com ara mitjanes, màxims/mínims, comptadors, etc) i fins i tot alertes (quan es detecta comportament "estrany" en les dades indexades, com ara un valor massa freqüent, o massa alt o massa baix, etc).

*"**OpenSearch Dashboards**" (https://opensearch.org/docs/latest/dashboards/index) : Aquest programa és un client d'OpenSearch que permet fer-li consultes sobre les dades que tingui emmagatzemades i, a la vegada, un servidor web (escoltant al port 5601) que permet mostrar a l'usuari diferents perspectives d'aquestes mitjançant un panel de control molt visual, oferint-hi multitud de gràfiques a disposar (mostrant tant dades directes com agregades) ja sigui en temps real o en forma de sèrie històrica, la possibilitat d'establir alertes segons el comportament de les dades observades, etc, etc

La combinació de Tshark més OpenSearch/Dashboards permet implementar una sol.lució integral de *recollida, filtratge i enviament de dades* → *emmagatzematge* → *visualització* Cal tenir en compte, en aquest sentit, que la ubicació dels elements responsables de la part indicada en color verd (en el nostre cas, són les tasques a realitzar pel Tshark) es correspon amb la màquina (o màquines!) de la/les qual/s se'n volen extreure les dades d'interès (en aquest cas, les dades filtrades dels paquets de xarxa triats) mentre que la ubicació de l'element responsable de la part indicada en color lila (en el nostre cas, són les tasques a realitzar per OpenSearch) es correspon amb la d'una única màquina (o clúster) que rep les dades remotes del/s orígen/s verd/s i les emmagatzema de forma centralitzada, controlada i homogènia dins d'un (o més) índexs per així poder, mitjançant el servei Dashboards (representat en color negre), accedir a elles gràfica i còmodament des de qualsevol navegador remot.

NOTA: Tècnicament, no és necessari que el servidor OpenSearch estigui instal·lat a la mateixa màquina que el servidor Dashboards (ja que aquest últim pot funcionar com un client remot del primer) però a la pràctica, el més habitual és que ambdós programes estiguin instal·lats a la mateixa màquina, de forma que Dashboards pugui accedir a les dades emmagatzemades per OpenSearch de forma local. En ambdós casos, l'usuari simplement ha de connectar-se (via web) al servidor Dashboards i ja està, ja que la comunicació interna entre ambdós programes és transparent per ell.

Cal tenir en compte, en tot cas, que OpenSearch només pot rebre dades mitjançant la seva API particular i no entén dades d'entrada que provinguin amb un altre format. És per això que moltes vegades, si l'orígen de les dades no sap enviar les dades en el format que OpenSearch necessita (o simplement, les genera però no les envia remotament, com podria ser el cas del Journald o bé de qualsevol servidor que generi logs locals, com per exemple l'Apache2 entre molts d'altres), s'utilitzin programes intermediaris que recullen les dades generades en l'origen i les reenvien en el format adient al servidor OpenSearch remot. Exemples d'aquests tipus de programes són **FluentBit** (https://fluentbit.io), **Vector** (https://vector.dev) o **LogStash** (https://www.elastic.co/es/logstash), entre molts d'altres. Afortunadament, en el cas d'utililitzar Tshark, aquest tipus de "reenviadors" (que a més també poden fer la funció de filtres, agregadors i d'enriquidors de les dades que els travessen) no són necessaris perquè Tshark sí que és capaç de generar una sortida compatible amb l'API d'ingesta d'OpenSearch. Concretament, el paràmetre *-T ek* de *tshark* formateja la seva sortida per a què sigui compatible amb l'API "Bulk" d'OpenSearch. Per tant, mitjançant l'ús d'aquest paràmetre la sortida estàndard generada per Tshark podria ser ingerida directament per un servidor OpenSearch.

> NOTA: El nom per defecte de l'índex on es guardaran totes les dades JSON generades per la comanda *tshark* amb el seu paràmetre *-T ek* és, tal com es pot veure observant el valor del camp *"index"."_index"*, "packets-YYYY-MM-DD" (on "YYYY" representa l'any, "MM" el mes i "DD" el dia actual)

Falta un detall per establir, que és com enviar "a través del cable" la sortida generada per Tshark (que ja està en el format adient) al servidor OpenSearch remot. Desgraciadament, Tshark no ofereix cap opció per realitzar la transmissió per xarxa d'aquesta sortida, així que haurem de fer servir eines addicionals. Tenint en compte que la interacció amb un servidor OpenSearch es fa a través d'una API REST, el més fàcil per fer l'enviament serà fer servir la comanda *curl* Més en concret, si suposem que tenim un fitxer (que anomenarem "captura.json") contenint dades capturades de paquets de xarxa (gràcies a haver executat Tshark amb el seu paràmetre *-w)* i formatades adientment (gràcies a haver afegit també el paràmetre *-T ek*), per fer efectiu el seu enviament a OpenSearch haurem d'executar una comanda similar a la següent...:

curl -X POST -k -u "admin:Un4C0ntraSs3ny4C0mplicad4_" -H "Content-Type:application/json" --data-binary @captura.json https://ip.serv.Open.Search:9200/_bulk

...on es veu que estem realitzant una petició de tipus POST a la URL "_bulk" del servidor OpenSearch (que és l'entrada de la seva API dedicada a fer importacions massives de dades, com és el cas), autenticant-nos amb l'usuari "admin" i contrasenya "admin" (són les credencials que el servidor OpenSearch implementa per defecte), per tal d'enviar-hi el contingut del fitxer indicat, "captura.json".

> NOTA: Una altra opció (potser més "professional") hauria pogut ser enviar el trànsit recollit igualment a un fitxer però llavors emprar algun servei recol·lector dels mencionats anteirorment per a què aquest llegís les dades presents en aquest fitxer i les processés/filtrés/transformés/enriquís per tal d'acabar enviant-les (mitjançant el seu "plugin" de sortida pertinent), al servidor OpenSearch.

Cal tenir en compte, d'altra banda, que la quantitat de camps JSON que per defecte es generen (recollits la majoria d'ells sota la secció "**layers**" del document) fent servir el paràmetre *-T ek* són molts (ja que es corresponen a tots els valors presents a tots els nivells -2,3,4,7- que formen part de cada paquet). És per això que a la comanda *tshark* normalment s'afegeixen alguns dels paràmetres addicionals següents:

> *-J "unnivell unaltre ..."* : a la secció "layers" del document JSON generat només s'inclouran els camps pertanyents al/s nivell/s indicat/s. Per nivell s'entén el valor "**frame**" (per incloure metadades dels paquets), el valor "**eth**" (dades de nivell 2), "**arp**" (dades del protocol ARP), "**ip**" (dades del protocol IP -nivell 3-), "**icmp**", "**tcp**" o "**udp**" (dades dels protocols corresponenents -nivell 4-) i "**http**", "**dns**" ... i, en general, qualsevol protocol de nivell 7 reconegut per Wireshark (dades dels protocols corresponents)
>
> *-P* : al document JSON generat només s'inclouran dades-resum (és a dir, les que són visibles habitualment a la zona superior de Wireshark, de les tres en les què normalment es divideix la seva pantalla), amb una secció "layers" buida.

NOTA: Si el que es vol és veure, a més de les dades-resum (gràcies al paràmetre *-P*), els camps dels eventuals nivells que s'hagin indicat amb el paràmetre *-J* corresponent, caldrà afegir llavors el paràmetre ***-V*** (si *-V* no s'indica, el paràmetre *-J* no es té en compte amb *-P*)
NOTA: En qualsevol cas, sempre es pot afegir el paràmetre ***-x***, que serveix per mostrar el payload dels paquets en format hexadecimal (és a dir, el contingut que és visible habitualment a la zona inferior del Wireshark de les tres en les què normalment es divideix la seva pantalla)

EXERCICIS

3.-Després de llegir els paràgrafs anteriors, prova cadascuna de les comandes següents mentre en un altre terminal executes, en cada cas, la mateixa comanda: *ping -c 1 1.1.1.1*

a) *tshark -f icmp -a packets:1 -T ek | jq "."* Observa l'estructura del document JSON generat i comprova que, a més d'incloure la secció "index" (on s'indica el nom de l'índex on s'hi insertaria aquest document, el qual, com ja hem dit a la teoria, és "*packets-YYYY-MM-DD*", on "YYYY" representa l'any, "MM" el mes i "DD" el dia actual) i el valor "timestamp" (on s'indica el temps en format UNIX quan s'ha detectat el paquet en qüestió), apareix la secció "layers" contenint les subseccions "frame" (incloent metadades del paquet capturat), "eth" (incloent dades de nivell 2), "ip" (incloent dades de nivell 3) i, en aquest cas, "icmp" (incloent dades de nivell 4) i prou

NOTA: La comanda *jq* serveix per mostrar el JSON generat per *tshark* d'una forma molt més llegible (amb colors, salts de línia, etc). Si no es posa, la sortida és la mateixa però tot escrit d'un sol cop i, per tant, molt més difícil de llegir
NOTA: La llista de tots els camps JSON que pot generar la comanda Tshark amb el paràmetre *-T ek* es pot consultar observant la sortida de la comanda ***tshark -G elastic-mapping***
NOTA: Si haguéssim afegit el paràmetre ***-x***, als valors mostrats anteriors s'hi hauria sumat els valors "raw" de cada nivell

b) *tshark -f icmp -a packets:1 -T ek* ***-J "icmp"*** *| jq "."* Observa l'estructura del document JSON generat i comprova que, a més d'incloure la secció "index" ja coneguda i el valor "timestamp", dins de la secció "layers" apareixen totes les seccions "filtered" excepte la indicada ("icmp", en aquest cas). Podem indicar si volem més seccions una rera l'altra dins del paràmetre *-J* (per exemple, escrivint *-J "eth icmp"* obtindríem només les dades d'aquestes seccions indicades i filtrar-ne tota la resta).

NOTA: Queda clar que, d'allò explicat al paràgraf anterior, la comanda *tshark -f icmp -T ek -J "frame eth ip icmp"* seria equivalent a la comanda *tshark -f icmp -T ek*
NOTA: En tot cas, sempre es poden utilitzar filtres de pantalla (amb el paràmetre *-Y* ja conegut) per restringir no les dades individuals visualitzades dels paquets sinó els paquets visualitzats en sí
NOTA: Existeix un paràmetre de Tshark similar a *-J* que és ***-j*** . Es pot consultar la diferència entre ells a *man thark* però, no obstant, a la pràctica no és tan útil i per això no el farem servir. D'altra banda, amb *-T ek* el paràmetre *-O* no té efecte

c) *tshark -f icmp -a packets:1 -T ek* ***-e "icmp.type"*** *| jq "."* Observa l'estructura del document JSON generat i comprova que, a més d'incloure la secció "index" ja coneguda i el valor "timestamp", dins de la secció "layers" només hi apareix la parella *nom:valor* corresponent al camp indicat amb el paràmetre *-e* (ja conegut quan vam veure l'opció *-T fields*), i res més

NOTA: Recordeu que el paràmetre *-e* es pot indicar tantes vegades com vulguem

d) *tshark -f icmp -a packets:1 -T ek* ***-P*** *| jq "."* Observa l'estructura del document JSON generat i comprova que, a més d'incloure la secció "index" ja coneguda i el valor "timestamp", la secció "layers" és buida de contingut i apareixen directament, i de forma independent, les dades que equivaldrien a les columnes mostrades al panell superior del Wireshark

e) *tshark -f icmp -a packets:1 -T ek -J "icmp" -P* ***-V*** *| jq "."* Observa l'estructura del document JSON generat i comprova que és una combinació de les dades obtingudes amb *-P* i les dades obtingudes amb *-J*

NOTA: Si no s'indica *-V*, només la combinació de *-P* amb *-J* no funciona com s'espera perquè en aquest cas *-J* és ignorat

4.-a) Instal·la, també a la màquina real, un servidor OpenSearch i Dashboards, així (si se't pregunta, tria el repositori d'imatges "docker.io")...:

podman pod create -n miOS -p 9200:9200 -p 5601:5601
podman container create --pod=miOS -e "discovery.type=single-node"
-e "OPENSEARCH_INITIAL_ADMIN_PASSWORD=Un4C0ntraSs3ny4C0mplicad4_"
docker.io/opensearchproject/opensearch:latest
podman container create --pod=miOS -e "opensearch.username=admin"
-e "opensearch.password=Un4C0ntraSs3ny4C0mplicad4_"
-e "opensearch.ssl.verificationMode=none"
docker.io/opensearchproject/opensearch-dashboards:latest

NOTA: Les comandes anteriors (disponibles només si està el paquet "podman" instal·lat, cosa que ja és així als ordinadors de l'aula) el que fan és posar en marxa un "pod" amb dos contenidors al seu interior, un pel servidor OpenSearch i un altre pel servidor Dashboards. Ho fem així per comoditat, per no haver d'utilitzar màquines virtuals on instal·lar via apt/rpm (per tant, com a "administrador") els paquets necessaris (però a l'hora d'interaccionar amb els servidors el seu comportament serà idèntic, tant se val com s'hagin instal·lat). Si no heu treballat amb cap contenidor i voleu saber què són i què ofereixen (i més en concret, els contenidors de tipus "podman"), podeu llegir aquest article introductori: https://elpuig.xeill.net/Members/vcarceler/articulos/introduccion-a-podman però d'entrada podem dir que un contenidor és una aplicació autocontinguda que té tot el necessari per funcionar independentment de l'estat del sistema operatiu subjacent

...i posa'l en marxa:

podman pod start miOS

NOTA: En tot cas, un cop executades les comandes anteriors, el servidor OpenSearch estarà disponible al port 9200 de la màquina on s'hi estigui executant (en el nostre cas, serà "127.0.0.1"), tot i que normalment no consultarem/modificarem les dades existents en els índex emmagatzemats directament a través d'ell (que podríem, però caldria fer-ho des del terminal fent peticions REST a mà) sinó que per això utilitzarem el servidor Dashboards com a "intermediari" visual, connectant-nos des del navegador també a 127.0.0.1, però ara al port 5601 i així poder accedir llavors a aquestes dades de forma molt més còmoda via web.
NOTA: Si volguéssim en algun moment aturar el "conglomerat" OpenSearch+Dashboards cal executar ***podman stop miOS*** Per veure si estan encesos o apagats aquests contenidors, cal executar ***podman ps -a*** En tot cas, es poden veure els logs dels diferents contenidors del "pod" amb ***podman pod logs miOS***. Podeu consultar més opcions a *man podman*
NOTA: En el cas que no hi hagi prou espai de disc dur lliure, el servidor OpenSearch rebutjarà posar-se en marxa indicant un missatge d'error similar a "too_many_requests/12/disk usage exceeded flood-stage watermark". Per sol·lucionar aquest problema, es pot indicar que el servidor OpenSearch sigui més flexible en aquest aspecte i que accepti arrencar tot i que el disc dur estigui completament ple. Això es pot fer executant la següent comanda, la qual modifica la configuració pertinent:

```
curl -X PUT -k -u "admin:Un4C0ntraSs3ny4C0mplicad4_"
                                -H "Content-Type:application/json" "https://127.0.0.1:9200/_cluster/settings" -d '{
        "persistent": {
                "cluster.routing.allocation.disk.watermark.low": "98%",
                "cluster.routing.allocation.disk.watermark.high": "99%",
                "cluster.routing.allocation.disk.watermark.flood_stage": "100%"
        }
}'
```

Abans de fer cap ingesta d'informació per part del servidor OpenSearch, cal preparar aquest per a què quan rebi les dades (en aquest cas, provinents de Tshark), les pugui interpretar de la forma correcta (és a dir, que reconegui els tipus de cadascuna: de text, numèric, de data, IPs....ja que recordeu que totes les dades incloses dins d'un document JSON no tenen tipus). Això és important perquè si no s'identifiquen els tipus de dades correctes, moltes operacions específiques (com per exemple calcular la mitjana d'un valor numèric o obtenir el rang d'adreces entre dues IPs concretes, etc, etc) no es podran fer correctament (a més que, ja que només les dades no textuals no s'indexen, sense identificació de tipus totes aquestes dades s'indexaran de forma innecessària, amb el cost computacional que això suposa). Per fer-ho, cal crear primer de tot una "plantilla de mapeig", que no és res més que un esquema on s'indica el tipus de cada camp del document JSON que s'espera per emmagatzemar.

Per exemple, un arxiu (que anomenarem "plantilla.json") amb el següent contingut seria una plantilla de mapeig on, en aquest cas, s'estaria indicant que un hipotètic camp anomenat "timestamp" existent en qualsevol índex anomenat "packets-*qualsevolcosa*" és de tipus data (amb un format de data específic, a més):

```
{   "index_patterns": ["packets-*"],
    "mappings": {
          "properties": {  "timestamp": {"type":"date", "format":"epoch_millis" }  }
    }
}
```

NOTA: Justament el camp "timestamp" és un dels camps que sovint trobem en els "mappings" fets als arxius de plantilla perquè el que es vol no és que sigui de tipus cadena sinó de tipus data per poder llavors renderitzar després gràfiques en Dashboards on el valor d'aquest camp es tingui el valor temporal de referència.

Un cop tenim l'arxiu "plantilla.json" fet, per "introduir" la informació que allà hi ha al servidor OpenSearch només caldria executar una comanda similar a la següent:

curl -X PUT -k -u "admin:Un4C0ntraSs3ny4C0mplicad4_" -H "Content-Type:application/json" -d @plantilla.json https://127.0.0.1:9200/_template/lamevaplantilla

A l'hora, però, de decidir quins camps volem incloure a l'arxiu de plantilla en el cas concret que les dades provinguin de Tshark, primer hauríem de conèixer quins són aquests camps. Afortunadament, la comanda *tshark -G elastic-mapping* genera un llistat JSON amb tots ells, els qual es pot retallar per a què només contingui la definició dels camps que volem amb l'opció *--elastic-mapping-filter ip,udp,dns...*, (ja que afegir en una plantilla tots els eventuals camps JSON de tots els protocols possibles és molt!). No obstant, el llistat JSON generat per la comanda anterior no és directament compatible amb el format que OpenSearch esperaria en un arxiu de plantilla, així que cal "processar" aquest JSON per transformar-lo en una plantilla vàlida per poder-la introduir al servidor OpenSearch. Aquest processament (que consisteix bàsicament en eliminar camps repetits i salts de línia i afegir alguns camps faltants) es pot fer amb l'eina estàndard de manipulació de dades JSON anomenada *jq* (https://jqlang.github.io/jq), tal com de seguida veurem.

b) Després de llegir els paràgrafs blaus anteriors, executa la comanda següent per generar amb el format adient l'arxiu que representarà la plantilla de mapeig que farem servir als índex que anirem creant al llarg dels exercicis següents:

tshark -G elastic-mapping --elastic-mapping-filter frame,eth,arp,ip,icmp,udp,tcp,dhcp,dns,http,ssh,ntp | jq -c '{"index_patterns":["packets-"]} + .' > plantilla.json*

NOTA: Com es pot veure, de tota la llista de camps generats pel Tshark hem sel·leccionat només uns quants que hem considerat més rellevants: "frame" (per les metadades dels paquets), "eth", "arp", "ip", "icmp", "udp", "tcp", "dhcp", "dns", "http", "ssh" i "ntp". Òbviament, podrien haver sigut més o també menys.
NOTA: La comanda *jq* indicada a la cadena de comandes anteriors fa vàries coses: amb el paràmetre -c "compacta" la sortida de tshark per a què no hi hagi salts de línia (l'API "bulk" d'OpenSearch necessita que sigui així) i a més, "de sèrie", elimina tots els eventuals camps duplicats que hi apareguin al JSON (això és molt important perquè si no es fa la ingesta de dades per part d'OpenSearch dóna errors). D'altra banda, amb l'estructura *'{"nom":"valor" + . }'* el que fa és afegir la parella *"nom":"valor"* al document JSON; com es pot veure, el que s'hi està afegint és el nom (amb comodins) dels índex als quals se'ls aplicarà la plantilla, nom que per defecte al JSON generat per *tshark* no s'indica i, per tant, cal afegir-ho "a mà".
NOTA: Un arxiu "plantilla.json" que ha servit d'inspiració a aquest apartat és el que es troba disponible a https://github.com/H21lab/tsharkVM/blob/master/Kibana/template_tshark_mapping_deduplicated.json

bBIS) Malauradament, és necessari fer uns "retocs" a l'arxiu "plantilla.json" generat al pas anterior perquè alguns tipus de dades assignats a certs camps del JSON generen error quan la plantilla és introduïda a l'OpenSearch. El més fàcil és fer que aquests "retocs" consisteixin en eliminar els camps problemàtics de la plantilla (afortunadament, el seu tipus no és important i podran ser assimilats a cadenes sense problemes). Això ho farem executant la següent comanda *sed*, que modifica l'arxiu "plantilla.json" adientment:

```
sed -ie 's/"frame_frame_offset_shift":{"type":"date"},/"frame_frame_offset_shift":{"type":"float"},/;s/"frame_frame_time_epoch":
{"type":"date"},//;s/"frame_frame_time_relative":{"type":"date"},/"frame_frame_time_relative":
{"type":"float"},/;s/"frame_frame_time_delta":{"type":"date"},/"frame_frame_time_delta":
{"type":"float"},/;s/"frame_frame_time_delta_displayed":{"type":"date"},//;s/"ip_ip_dsfield":{"type":"short"},//;s/"ip_ip_id":
{"type":"integer"},//;s/"ip_ip_flags":{"type":"short"},//;s/"ip_ip_checksum":{"type":"integer"},//;s/"udp_udp_checksum":
{"type":"integer"},//;s/"udp_udp_time_relative":{"type":"date"},/"udp_udp_time_relative":
{"type":"float"},/;s/"udp_udp_time_delta":{"type":"date"},/"udp_udp_time_delta":{"type":"float"},/;s/"udp_udp_payload":
{"type":"byte"}/"udp_udp_payload":{"type":"text"}/;s/"tcp_tcp_checksum":{"type":"integer"},//;s/"tcp_tcp_time_relative":
{"type":"date"},/"tcp_tcp_time_relative":{"type":"float"},/;s/"tcp_tcp_time_delta":{"type":"date"},/"tcp_tcp_time_delta":
{"type":"float"},/;s/"tcp_tcp_payload":{"type":"byte"},//;s/"tcp_tcp_flags":{"type":"integer"},//;s/"tcp_tcp_options":
```

{"type":"byte"},//;s/"tcp_tcp_analysis_ack_rtt":{"type":"date"},//;s/"tcp_tcp_analysis_initial_rtt":
{"type":"date"},//;s/"tcp_tcp_analysis_rto":{"type":"date"},//;s/"tcp_tcp_segment_data":
{"type":"byte"},//;s/"icmp_icmp_checksum":{"type":"integer"},//;s/"icmp_icmp_time_relative":
{"type":"date"},/"icmp_icmp_time_relative":{"type":"float"},/;s/"icmp_icmp_time_delta":{"type":"date"},/"icmp_icmp_time_delta":
{"type":"float"},/;s/"icmp_icmp_data_time_relative":{"type":"date"},/"icmp_icmp_data_time_relative":
{"type":"float"},/;s/"icmp_icmp_payload":{"type":"byte"},//;s/"dns_dns_id":{"type":"integer"},//;s/"dns_dns_flags":
{"type":"integer"},//;s/"dns_dns_qry_class":{"type":"integer"},//;s/"dns_dns_resp_ext_rcode":
{"type":"short"},//;s/"dns_dns_resp_z":{"type":"integer"},//;s/"dns_dns_resp_z_reserved":
{"type":"integer"},//;s/"dns_dns_resp_opt_data":{"type":"byte"},//;s/"dns_dns_resp_opt_cookie_client":
{"type":"byte"},//;s/"dns_dns_resp_class":{"type":"integer"},//;s/"dns_dns_time":{"type":"date"},//;s/"dns_dns_opt_data":
{"type":"byte"},//;s/"dns_dns_opt_cookie_client":{"type":"byte"},//;s/"ssh_ssh_cookie":
{"type":"byte"},//;s/"ssh_ssh_packet_length_encrypted":{"type":"byte"},//;s/"ssh_ssh_kex_reserved":
{"type":"byte"},//;s/"ssh_ssh_encrypted_packet":{"type":"byte"},//;s/"ssh_ssh_padding_string":
{"type":"byte"},//;s/"ssh_ssh_ecdh_q_c":{"type":"byte"},//;s/"ssh_ssh_host_key_eddsa_key":
{"type":"byte"},//;s/"ssh_ssh_ecdh_q_s":{"type":"byte"},//;s/"ssh_ssh_payload":{"type":"byte"},//;s/"ssh_ssh_mac":
{"type":"byte"},//;s/"http_http_time":{"type":"date"},//;' plantilla.json

bTRIS) Afegeix el següent contingut just després de la clau que obre el primer camp "properties" existent a l'arxiu "plantilla.json" (és a dir, just abans del camp "timestamp":

"protocol":{"type":"keyword"},"info":{"type":"text"},

> NOTA: Aquest apartat és necessari fer-ho per a què els camps "protocol" i "info", generats pel paràmetre *-P* de Tshark es reconeguin com a d'un tipus vàlid per OpenSearch (concreta i respectivament, de tipus "keyword" per tal de poder utilitzar-lo per classificar els paquets segons els seus diferents valors en les gràfiques que farem, i de tipus "text" per poder-hi fer recerques a la pàgina "Discover" i similars).

bII) Un cop ja amb la plantilla ben formatada, executa finalment la comanda *curl* indicada en els paràgrafs blaus anteriors. ¿Per a què serveix?

> NOTA: Si volguéssim eliminar una plantilla ja introduïda (anomenada per exemple "lamevaplantilla"), caldria executar la comanda següent: ***curl -X DELETE -k -u "admin:Un4C0ntraSs3ny4C0mplicad4_" https://127.0.0.1:9200/_template/lamevaplantilla***
>
> NOTA: En lloc d'haver creat l'arxiu de mapeig de dades i d'introduir-ho al servidor OpenSearch des del terminal, una altra opció hagués sigut fer-ho tot des del propi Dashboards. En concret, anant al menú "**Management**"->"**Index management**"->"**Templates**" i allà pulsant el botó de "Create template". Al formulari web que hi apareixeria només caldria indicar tres ítems (la resta es poden deixar amb el seu valor per defecte): el nom de la plantilla (per exemple, "lamevaplantilla"), l'"índex pattern" (és a dir, el nom o noms -amb comodins, si es vol- dels índexs als qual afectarà aquesta plantilla, en aquest cas, "packets-*") i, a l'apartat "Index mapping" del final del formulari, es pot o bé indicar un a un el nom i tipus dels camps que volguem (amb la pestanya "Visual editor") o bé escriure'ls "a mà" (amb la pestanya "JSON editor").

c) Accedeix des del navegador de la teva màquina real al servidor Dashboards anant a la URL https://127.0.0.1:5601 i escrivint l'usuari "admin" amb contrasenya "admin" (aquestes són les credencials per defecte d'un servidor Dashboard recentment instal·lat). Hauràs de per poder accedir al panell web (després, això sí, d'haver clicat al botó "Explore on my own" per no treballar amb dades de mostra i havent indicat tot seguit l'opció "Global" per fer que totes les gràfiques generades puguin ser compartides per altres usuaris possibles del servidor OpenSearch -tot i que no en crearem cap a banda de l'usuari "admin" ja existent-). Un cop aquí, comprova, anant al menú **"Management"->"Index management"->"Templates"** que la plantilla introduïda a l'apartat anterior, efectivament, allà hi surt llistada.

d) Ara procedirem, per fi, a insertar dades provinents del Tshark al servidor OpenSearch. Per fer això, executa el següent script en un terminal d'aquesta manera (estem suposant que s'anomena "tshark2os.sh"): *./tshark2os.sh "udp port 53" "frame ip udp dns"*) i digues què fa el seu codi:

```
#!/bin/bash
while [[ true ]]; do
    tshark -f "$1" -a duration:10 -T ek -J "$2" -P -V  > captura.json
    jq -c "del(.index._type)" captura.json > capturabona.json
    curl -X POST -k -u "admin:Un4C0ntraSs3ny4C0mplicad4_"  https://127.0.0.1:9200/_bulk
                                -H "Content-Type:application/json" --data-binary @capturabona.json
done
```

NOTA: La comanda *jq* utilitzada en l'script anterior és necessària per "arreglar" el fitxer "captura.json" (concretament, per eliminar el subcamp "_type" dins del camp "index") ja que en realitat la sortida generada per *-T ek* està pensada per un servidor ElasticSearch i no OpenSearch (que és un "fork" del primer) i d'aquí aquestes divergències.
NOTA: La possibilitat que ens ofereix la comanda *jq* d'eliminar camps concrets d'un document JSON, com s'acaba d'explicar, podria ser molt interessant de fer-la servir (tot i que en aquest exercici no ho farem) per manipular altres elements dels documents JSON generats per Tshark. Per exemple, es podrien eliminar d'una hipotètica captura els camps "layers.tcp.tcp_tcp_payload" i "layers.http.http_http_file_data" (per exemple) amb una comanda com aquesta *jq -c "del(.layers.tcp.tcp_tcp_payload) | del(.layers.http.http_http_file_data)" captura.json* Es deixa com ampliació voluntària

dII) Mentre l'script anterior es manté en funcionament, en un altre terminal fes algunes consultes DNS (amb comandes com *resolvectl, dig* o altres) de noms FQDN qualssevol diferents d'Internet.

e) Accedeix de nou des del navegador de la teva màquina real al servidor Dashboards i ara vés al menú "**Management**"->"**Index management**"->"**Indices**": allà hauries de veure com, entre altres índex interns, apareix un índex anomenat "packets-*yyyy-mm-dd*" (corresponent a les tuples rebudes). Clica sobre el seu nom per veure la pàgina de detalls de l'índex, on hauries d'observar, entre altres dades, la mida de l'index, el nombre de documents que conté, el "mapping" dels camps (és a dir, els seus tipus, que el servidor OpenSearch ha deduit, gràcies a la plantilla que hem establert a l'apartat bII)), etc. Aquí podràs confirmar que tot funciona correctament.

f) Abans de poder treballar en el Dashboards amb els documents JSON emmagatzemats dins d'índexs al servidor OpenSearch cal que indiquem al Dashboads, però, quin/s índex/s en concret volem que "tingui en compte", ja que al servidor OpenSearch hi poden haver molts índexs que no volguem visualitzar en el panell web. És per això que primer has d'anar al menú **"Management"->"Dashboard management"->"Index patterns**" i allà crear un "index pattern" anomenat "packets-*" (indicant, a la segona pàgina de l'assistent el camp "timestamp" com a camp que es prendrà com a referència temporal). D'aquesta manera, Dashboards entendrà que tots els índex que s'anomenin com l'"index pattern" que hem dit els haurà de tenir en compte a l'hora de visualitzar-los al panell web (i els que el seu nom no concordi amb aquest "index pattern" romandran "ocults").

g) Ara ja podràs observar les dades generades per Tshark directament des de la pestanya "**OpenSearch Dashboards"->"Discover**". Hauràs de veure com apareixen, distribuïts al llarg d'una línia temporal mostrada a la part superior de la pàgina, la quantitat de documents JSON rebuts de part de Tshark, el contingut de cadascun dels quals es pot visualitzar a la part central d'aquesta, mentre que a la zona esquerra de la pàgina es troben llistats de forma individual els noms dels diversos camps que poden estar presents en aquests documents. Si és així, tot és correcte i pots continuar.

5.-a) Atura l'execució de l'script "tshark2os.sh" arrencat a l'exercici anterior (pulsant CTRL+C vàries vegades) i ara torna'l a executa de nou així: *./tshark2os.sh "tcp port 80 or tcp port 443" "frame ip tcp tls http"* per tal de capturar i enviar a l'OpenSearch ara paquets de tipus HTTP/HTTPS.

aII) Mentre l'script anterior està en marxa, en un altre terminal executa la comanda *curl* vàries vegades contra diferents webs qualssevol d'Internet, ja siguin HTTP com HTTPS.

aIII) Actualitza la vista "Discover" del Dashboards per observar que has rebut nous documents, els quals ara tenen alguns camps diferents als rebuts a l'exercici anterior (concretament, els corresponents al protocol TCP -en lloc de UDP- i HTTP o TLS -en lloc de DNS-). Si és així, tot és correcte i pots continuar.

b) Atura l'execució de l'script "tshark2os.sh" i ara torna'l a executa de nou així: *./tshark2os.sh "icmp" "frame ip icmp"* per tal de capturar i enviar a l'OpenSearch ara paquets de tipus ICMP.

bII) Mentre l'script anterior està en marxa, en un altre terminal executa la comanda *ping* contra diferents destins qualssevol, ja siguin de la xarxa local o d'Internet

bIII) Actualitza de nou la vista "Discover" del Dashboards per observar que has rebut nous documents, els quals ara tenen alguns camps diferents als rebuts a l'exercici anterior (concretament, els corresponents al protocol ICMP -en lloc de UDP o TCP- i cap protocol del nivell 7). Si és així, tot és correcte i pots continuar.

NOTA: Podríem anar provant de captura altres tipus de tràfic, com DHCP, SSH, NTP, etc però ja es veu com seria

6.-a) Dins de la mateixa vista "Discover", clica sobre el botonet petitó "+" que apareix al costat dels noms dels camps que es veuen a la columna vertical a l'esquerra de la taula de files central per tal de personalitzar aquesta taula per a què mostri per exemple només els valors de tres camps (que pots cercar indicant part del seu nom a la caixa de text pertinent, si ho necessites): les corresponents a les dades dels camps "timestamp" (encara que per defecte aquesta columna sempre es mostra amb el títol de "Time"), "layers.ip.ip_**ip_src**" i "layers.frame.frame_**frame_len**".

NOTA: Els noms dels camps generats per *-T ek* són molt llargs però un truc per identificar el seu significat és fixar-se que el final del seu nom es correspon al filtre de pantalla corresponent a la dada que desitjem (substituïnt el "." per "_")
NOTA: Si has afegit varis "Index patterns" a la configuració de Dashboards, pots anar alternant les visualitzacions d'un o altre si cliques sobre el desplegable mostrat sobre la barra vertical que mostra els noms dels camps

aII) ¿Què es veu quan es clica sobre la icona de la "lupa sobre un full" mostrada a l'esquerra de cada registre de la taula central?

El quadre de recerca que apareix a dalt de la pàgina "Discover" es pot fer servir per buscar documents que continguin un valor exacte en algun camp concret, amb la sintaxi *nomCamp:valor* (o, si el valor és de tipus textual, en qualsevol camp amb la sintaxi *valor* -o entre cometes si el valor conté espais-). Més detalls:

*Les recerques dels noms de camps i els seus valors si són de tipus textual són case-insensitive
*Es pot fer servir el comodí * en els valors de tipus textual
*Es poden fer servir els operadors matemàtics >, >=, < o <= per fer comparacions entre valors numèrics, i també entre dates. Combinats, a més, amb els operadors booleans and, or o not es poden indicar també recerques per rangs (del tipus *nomCamp > valor and nomCamp < valor*)

NOTA: Una sintaxi alternativa per indicar rangs de valors per un camp és *nomCamp:[x TO y]* o *nomCamp:{x TO y}* (si s'escriu entre [] éls extrems del rang s'inclouen i si s'escriu entre {} els extrems no s'inclouen). Per més informació, en tot cas, sobre la sintaxi de les recerques, consulta https://opensearch.org/docs/2.11/dashboards/dql

NOTA:Alternativament, per tal d'escriure filtres de forma més fàcil, es pot utilitzar botó/assistent "Add filter" que apareix a la banda esquerra dalt de tot, sobre la llista de camps. D'altra banda, també es pot clicar sobre la icona que apareix al costat de cada nom de camp (en el llistat de camps de l'esquerra de la pàgina) amb el dibuix d'un lupa sobre un full per tot seguit clicar sobre el símbol "+" associat a algun dels valors mostrats, el qual volem fer servir per filtrar només els documents que tinguin en el camp seleccionat exactament aquest mateix valor

aIII) ¿Què veus si escrius *layers.tcp.tcp_**tcp_dstport** > 80* al quadre de recerca de la pàgina "Discover"?

b) Vés a la pestanya **"OpenSearch Dashboards"->"Visualize**" per generar els diferents gràfics de tipus "Pie". Un cop sel·leccionat aquest tipus de la pantalla on es poden triar els tipus de gràfics, tots els de tipus "Pie" es fan igual: l'apartat "Metrics" cal deixar-ho com està ("*Slice size->Count*") i a l'apartat "Bucket" cal sel.leccionar "*Split slices->Aggregation->Terms*" i llavors sel.leccionar el camp pel qual es dividirà el pastís. Quan hagis finalitzat cadascun d'aquests gràfics i comprovis que es visualitzen correctament, no t'oblidis de guardar-los amb el botó "Save" que apareix al menú horitzontal de links de la zona superior dreta del panel web (cada gràfic amb el nom que vulguis). Concretament, dibuixa el següent:

NOTA: Estem suposant que ja has generat prou tràfic ICMP, DNS i HTTP/S dels exercicis anteriors, però si noés així, l'hauràs de generar prèviament per tenir certa quantitat de tràfic i així poder veure gràfics amb un contingut interessant...

*"Pie" comptant la quantitat de paquets detectats segons la seva mida
Pista: el camp a indicar en el valor agregat "Terms" és *layers.frame.frame_frame_len*

*"Pie" comptant la quantitat de paquets detectats segons la seva IP d'origen
Pista: el camp a indicar en el valor agregat "Terms" és *layers.ip.ip_ip_src*

*"Pie" comptant la quantitat de paquets detectats segons el seu protocol de nivell 4 (UDP, TCP, etc)
Pista: el camp a indicar en el valor agregat "Terms" és *layers.ip.ip_ip_proto*

NOTA: El valor 1 representa ICMP, el valor 6 representa TCP i el valor 17 representa UDP (més aquí: https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers)

*"Pie" comptant la quantitat de paquets detectats segons el seu port TCP de destí.
Pista: el camp a indicar en el valor agregat "Terms" és *layers.tcp.tcp_tcp_dstport*

NOTA: Més opcions interessants mostrades en aquest panell d'edició de gràfics són, a banda de les purament estètiques (les quals dependran del tipus de gràfica en qüestió), la possibilitat d'actualitzar de forma automàtica els gràfics (sense haver de clicar manualment, doncs, al botó "Refresh") gràcies a la manipulació de la icona de calendari (i concretament, el seu apartat "Refresh every"), la qual es convertirà llavors en la icona d'un rellotge, així com també la possibilitat d'afegir filtres (mitjançant el desplegable "Add filter" mostrat a la part superior esquerra de la finestra) per tal de només visualitzar al gràfic les dades que concordin amb la condició indicada, que generalment serà del tipus *"nomcamp" operador "valor"*

bII) ¿Quina informació obtindràs, d'altra banda, si fas un gràfic "Pie" on a l'apartat "Metrics" sel.lecciones "*Slice size->Sum->Field=layers.frame.frame_frame_len*" i a l'apartat "Bucket" sel.lecciones la IP d'origen com a terme divisor?

c) Vés al link "**OpenSearch Dashboards**" -> "**Dashboard**" del menú principal per tal de crear un nou "dashboard" que inclogui quatre panels, cadascun dels quals haurà de mostrart cadascuna de les visualitzacions creades (i guardades) a l'apartat b) d'aquest mateix exercici. ¿Quines opcions tens disponibles al botó de la icona de la roda dentada que apareix a la cantonada superior dreta de cada visualització present al dashboard? Guarda finalment aquest dashboard amb el botó "Save" que apareix al menú horitzontal de links de la zona superior dreta del panel web (amb el nom que vulguis).

7.-Un cop ja sabem com generar gràfiques bàsiques a partir de les dades obtingudes, a continuació aprofundirem en aquest aspecte. Concretament, dibuixa ara el següent:

a) Un gràfic "Pie" comptant els paquets segons el seu protocol de nivell 7 reconegut (HTTP,DNS,NTP, etc)
Pista: el paràmetre "*Metrics->"Slice size*" ha de valer "*Count*" i el camp a indicar en el valor agregat "Terms" sota el paràmetre *"Buckets"->"Split slices"* ha de ser *protocol*

b) Un gràfic "Pie" sumant el nombre de bytes per cada protocol de nivell 7 reconegut (HTTP,DNS,NTP, etc)
Pista: el paràmetre "*Metrics->"Slice size*" ha de valer "*Sum*" sobre el camp *layers.frame.frame_frame_len* , i el camp a indicar en el valor agregat "Terms" sota el paràmetre *"Buckets"->"Split slices"* ha de ser *protocol*

c) Un gràfic "Pie" sumant el nombre de bytes per cada IP d'origen detectada
Pista: el paràmetre "*Metrics->"Slice size*" ha de valer "*Sum*" sobre el camp *layers.frame.frame_frame_len* , i el camp a indicar en el valor agregat "Terms" sota *"Buckets"->"Split slices"* ha de ser *layers.ip.ip_ip_src*

d) Transforma qualsevol dels gràfics creats en els apartats anteriors per a què en lloc de ser de tipus "Pie" siguin de tipus "Data Table"

e) Un gràfic "Pie" sumant el nombre de bytes per cada protocol de nivell 7 reconegut segons cada IP d'origen
Pista: el paràmetre "*Metrics->"Slice size*" ha de valer "*Sum*" sobre el camp *layers.frame.frame_frame_len* , i el camp a indicar en el valor agregat "Terms" sota el paràmetre *"Buckets"->"Split slices"* ha de ser *protocol*
Un cop fet això, cal afegir de nou un altre paràmetre *"Buckets"->"Split slices"* i allà indicar com a valor agregat de tipus "Terms" ara el camp *layers.ip.ip_ip_src*

f) Dibuixa un gràfic de tipus "Line" amb les següents característiques. ¿Què se't mostra?

· Metrics -> *Y-Axis = Count*
· Bucket -> *X-Axis -> Aggregation = Date Histogram* i *Field= "timestamp"*

NOTA: "Date histogram" és un tipus de gràfica on s'agrupen, en blocs d'una mida determinada, els valors a mesurar (en aquest cas, el nombre de paquets detectats) a través de la línia temporal. És el tipus de gràfica, de fet, que es veu a la part superior de la pàgina "Discover" (en aquest cas, en format de barres en lloc de línies, però això tan sols és un detall estètic)

fII) ¿I si afegeixes, sobre el "Bucket" ja creat del gràfic anterior, un altre "Bucket" d'aquest tipus?:

· *Split series -> Aggregation = Terms i Field= "layers.tcp.tcp_tcp_dstport"*

fIII) ¿I si afegeixes, al gràfic anterior, un filtre que mostri només els valors del camp indicat al "Bucket" anterior que estiguin entre 1 i 1024?

fIV) ¿I si canvies el "field" *layers.tcp.tcp_tcp_dstport* del "Bucket" anterior pel "field" *layers.ip.ip_ip_src*?

fV) ¿I si canvies el "field" *layers.ip.ip_ip_src* del "Bucket" anterior pel "field" *protocol*?

g) Dibuixa ara un gràfic de tipus "Pie" amb les següents característiques. ¿Què se't mostra?

· Metrics -> *Slice size = Count*
· Bucket -> *Split slices -> Aggregation = Terms* i *Field= "layers.dns.dns_dns_a"*

gII) ¿I si canvies el "field" *layers.dns.dns_dns_a* del "Bucket" anterior per *layers.ip.ip_ip_src* i, a més, afegeixes un filtre que indiqui que només es volen tenir en compte en la gràfica els paquets amb el valor *true* en el camp *layers.dns.dns_dns_flags_response* ?

h) Dibuixa ara un gràfic de tipus "Line" amb les següents característiques. ¿Què se't mostra?

· Metrics -> *Y-Axis = Count*
· Bucket -> *X-Axis -> Aggregation = Date Histogram i Field= "timestamp"*
  + *Split series -> Aggregation = Terms* i *Field= "layers.ip.ip_ip_dst"*

hII) ¿I si canvies el "field" *layers.ip.ip_ip_dst* del "Bucket" anterior pel "field" *layers.udp.udp_udp_dstport* i, a més, afegeixes un filtre que indiqui que només es volen tenir en compte en la gràfica els paquets amb un valor entre 1 i 1024 d'aquest mateix "field", el *layers.udp.udp_udp_dstport*?

i) Dibuixa un altre gràfic de tipus "Line" amb les següents característiques. ¿Què se't mostra?

· Metrics -> *Y-Axis = Count*
· Bucket -> *X-Axis -> Aggregation = Date Histogram i Field= "timestamp"*
  + *Split series -> Aggregation = Terms* i *Field= "layers.http.http_http_response_code"*
· Amb el següent filtre activat: *layers.http.http_http_response is true*

iII) ¿I si canvies el "field" *layers.http.http_http_response_code* del "Bucket" anterior pel "field" *layers.http.http_http_content_length* i, a més, afegeixes un filtre que indiqui que només es volen tenir en compte en la gràfica els paquets amb el valor *true* en el camp *layers.http.http_http_request* ?