En aquesta activitat veurem com funciona una alerta al Netdata i en crearàs una.

Les alarmes tenen tres estats: CLEAR, WARNING i CRITICAL. On CLEAR vol dir que tot està en ordre, WARNING que alguna cosa no va del tot bé, i CRITICAL que cal atenció ràpida. Quan configurem una alarma, nosaltres decidim quins intervals de valors corresponen a cada estat.
Netdata ve amb moltes alarmes preconfigurades que nosaltres podem modificar o completar amb altres de nostres. Anem a veure com fer-ho.
Prenem per exemple l'alarma d'utilització de la CPU:

Aquí podem veure:

- *warning when*: una expressió que decideix quan hi ha d'haver un avís del tipus *warning*.
- *critical when*: una expressió que decideix quan hi ha d'haver un avís del tipus *critical*.
- *db lookup*: una expressió que avalua paràmetres del sistema. El resultat s'emmagatzemarà a la variable **$this**.
- *check every*: cada quan avaluarà l'expressió definida a *db lookup*.
- *execute*: fitxer de notificacions.
- ***source*: fitxer de configuració d'aquesta alarma.**

Editem el fitxer de configuració d'aquesta alarma (el mateix Netdata ens proporciona l'editor *edit-config*):

cd /etc/netdata
sudo ./edit-config health.d/cpu.conf

```
 template: 10min_cpu_usage
       on: system.cpu
    class: Utilization
     type: System
component: CPU
       os: linux
    hosts: *
   lookup: average -10m unaligned of user,system,softirq,irq,guest
    units: %
    every: 1m
     warn: $this > (($status >= $WARNING)  ? (75) : (85))
     crit: $this > (($status == $CRITICAL) ? (85) : (95))
    delay: down 15m multiplier 1.5 max 1h
     info: average CPU utilization over the last 10 minutes (excluding iowait, nice and steal)
       to: sysadmin
```

template: Nom de l'alarma.
on: Nom del grafic que escolta.
lookup: Càlcul de l'expressió d'aquesta alarma, en aquest cas el promig dels valors *user, system, softirq, irq* i *guest* dels últims 10 minuts i s'avalua cada minut. El valor resultant és un percentatge i s'emmagatzema a la variable ***$this***.
every: cada quan fa el càlcul.
Avisos:

warn: $this > (($status >= $WARNING) ? (75) : (85))
crit: $this > (($status == $CRITICAL) ? (85) : (95))

Això són expressions *if-then-else* i les hem de llegir com:
if ($status>=$WARNING) then return 75 else return 85

Les variables $status, $WARNING i $CRITICAL emmagatzemen estats:

- $WARNING i $CRITICAL són constants.
- $status pot prendre els valors: CLEAR, WARNING o CRITICAL, on CLEAR<WARNING<CRITICAL.

$status conté l'estat de l'alarma just abans de l'últim càlcul de l'expressió *lookup*.

El control d'estats el realitza per evitar les *flapping alarms*. Això és l'enviament d'alarmes contínues pel mateix fet cada minut per exemple.
Per tant, l'avís *warn: $this > (($status >= $WARNING) ? (75) : (85))* el podem llegir com:

Si estem a WARNING O CRITICAL retorna 75 (tornem a l'estat CLEAR si baixem del 75%)
Si estem a CLEAR retorna 85 (s'activa *warning* quan superem el 85%)

info: missatge que volem rebre en la notificació de l'alerta.
to: a qui envia l'alerta.

Silenciar un alarma:

Canvia l'opció ***to:sysadmin*** per ***to:silent***

Activitat: Crea una alerta per detectar un escanig de més de 100 ports.

1. Primer investiga què és un TCP Reset. Fes-ne una petita explicació i digue's en quins casos s'utilitza.
2. El Netdata té alarmes creades per detectar TCP resets. Estan definides al fitxer *tcp_resets.conf*. Prova de fer nmaps a la teva màquina on hi tens el Netdata. Adjunta una captura de l'alerta que has rebut.
3. Després d'estudiar i entendre com funcionen les alarmes creades per Netdata sobre TCP resets, crea una alerta al mateix fitxer *tcp_resets.conf* que comenci amb el teu nom (per exemple elteunom_mes_de_100_TCP_Resets) i que t'avisi quan la teva màquina

enviï més de 100 tcp resets seguits. Adjunta una captura de l'alerta que acabes de fer. Com a mínim ha de contenir:

- *alarm:* el nom de la teva alarma
- *on: ipv4.tcphandshake*
- *lookup:* el teu càlcul
- *warn:* la teva alerta
- *info:* missatge d'alerta que vols rebre
- *to: sysadmin*

Quan crees una alarma, pots reiniciar el servei amb: *sudo netdatacli reload-health*

4. Adjunta una captura de la teva alerta que has rebut per correu electrònic.
5. Completa el teu exercici utilitzant el Wireshark amb el filtre adequat per obtenir la IP de la màquina que t'està fent l'*nmap*. Adjunta també una captura amb el filtre i la IP de l'atacant.

Expressions del Netdata:

- +, -, *, /, <, <=, <>, !=, >, >=, &&, ||, !, AND, OR, NOT.
- Als booleans l'1 és true, i el 0 és false.
- If-then-else: (condició) ? (expressió *true*) : (expressió *false*)
- *min2max:* retorna la diferència (*max-min)* d'un conjunt de valors
- *sum:* retorna la suma
- *incremental-sum*: suma les diferències entre cada valor i el següent
- *average*: calcula el promig
- *unaligned*: Netdata alinea les dades per minuts (del segon 0 al 59) o hores (del minut 0 al 59) seguint el rellotge per tal que els gràfics tinguin una forma constant. Si li diem unaligned agafarà els instants reals sense alinear la informació als intervals de rellotge.