Aleix Roldán
Marc Salvador
Daniel Hernández

PRÀCTICA BASH

Sistema de monitorització per un entorn informàtic que permet controlar tots els serveis informàtics d'una empresa.

El sistema controla tant els serveis com els paràmetres interns dels diferents servidors.

El control dels serveis ho farem des del servidor de monitorització a través d'un escaneig de ports a cadascuna de les IP, mentre que el control dels paràmetres interns ho farem a través d'un agent instal·lat a cadascun dels servidors que vulguem controlar, de forma que el sistema de monitorització pugui anar interrogant als agents sobre els seus recursos.

La present solució aportada, consta d'un procés de seguiment mitjançant una sèrie d'arxius:

1. fotoini.sh

Obté informació inicial de la xarxa en la que les màquines estan connectades, obviant el Router i la IP de broadcast, així com els ports oberts per a cada servei, creant un arxiu (portsIP.txt) on es llistaran les diferents IP's, així com els ports que són oberts en el moment inicial de la monitorització per tal de poder veure canvis en un futur:

```
#!/bin/bash

# asignem una variable per a guardar la nostra IP
mip=$(ip a | grep "inet " | grep -v "127.0" | awk '{print $2}' | awk -F "/" '{print $1}')

# amb nmap escanejem totes les IP a les que te acces aquest servidor de monitoritzacio i les guardem
 en un txt
nmap -sn 10.0.2.0/24 -oG ips.txt >/dev/null 2>&1

# llegim el txt i filtre per certes paraules clau, obtenint les IP formatejades correctament
cat ips.txt | grep -v "Nmap" | grep -v "\.1 " | grep -v ".254 " | grep -v $mip | awk '{print $2}' |
sponge ips.txt

# creem un arxiu anomenat portsIP buit, en cas de existeixi que es borri
cat /dev/null > portsIp.txt

# fem un bucle que itera de ip en ip, fent un escanejat de ports per cadas cuna i guardant el result
at formatejat en l'arxiu anteriorment creat
for ip in $(cat ips.txt)
do
                nmap -p- -sS --min-rate 5000 -n -Pn $ip | grep open | awk -F "/" -v ip=$ip '{print i
p , $1 , "open"}' >> portsIp.txt
done
```

La creació de la foto inicial ens aportarà els dos següents arxius .txt:

ips.txt

```
root@monitor:/home/daniel/monitor# cat ips.txt
10.0.2.108
10.0.2.110
10.0.2.111
```

En aquest arxiu podem observar les IP les quals te contacte el servidor monitor i es farà servir per a iterar les IP per a poder escanejar els ports oberts

portsIp.txt

```
root@monitor:/home/daniel/monitor# cat portsIp.txt
10.0.2.108 22 open
10.0.2.108 1234 open
10.0.2.108 2222 open
10.0.2.108 4444 open
10.0.2.110 22 open
10.0.2.110 1234 open
10.0.2.110 4444 open
10.0.2.110 6969 open
10.0.2.111 22 open
10.0.2.111 420 open
10.0.2.111 1234 open
10.0.2.111 4444 open
```

En aquest arxiu podem observar els ports oberts per a cada IP, per a tindre una traçabilitat del estat d'aquests inicial.

2. evaluaPort.sh

Aquest codi, amb l'ajut del arxiu portsIP.txt (creat anteriorment per fotoini.sh) comprova si tot segueix en el mateix estat.

Per a això crearem un altre arxiu, nouPorts.txt amb la informació obtinguda actualment i la compararem mitjançant la comanda **diff** amb l'arxiu portsIP.txt.

Al fer servir aquesta comanda, si hi ha hagut canvis, els arxius seran diferents, i per tant segurament un o varis ports han caigut, fent que torni un **$? erroni**, substituint l'arxiu anterior amb les IP i ports oberts. Posteriorment faríem servir l'arxiu avisMail.sh per assabentar-nos, i també es faria una subrutina (interroga.sh) amb les diferents IP que connectarien amb els diferents agents.

```
#!/bin/bash

# creem un arxiu anomenat nouPorts i si existeix el buida
cat /dev/null > nouPorts.txt
# creem una variable de control booleana
canvi=false

# amb un bucle formatejem els resultats de la lectura de l'arxiu portsIP i obtenim les variables per
 separat per a poder treure-les per pantalla posteriorment, ja estigui tancat o obert tot comprovat
amb un echo >/dev/null/{IP}/{PORT}
while read line
do
        estat=$(echo $line | awk '{print $3}')
        ip=$(echo $line | awk '{print $1}')
        port=$(echo $line | awk '{print $2}')

        (echo >/dev/tcp/$ip/$port) >/dev/null 2>&1

        if [[ $? -eq 0 ]]
        then
                echo "$ip $port open" >> nouPorts.txt
        else
                echo "$ip $port close" >> nouPorts.txt
        fi

done < portsIp.txt
```

```
# comprovem que el resultat sigui diferent al que tenia manteriorment, es a dir que hi ha hagut un c
anvi
diff nouPorts.txt portsIp.txt >/dev/null 2>&1

# si el resultat no es 0 es a dir, hi ha hagut un canvi, canviem la variable control "canvi"
if [[ $? -ne 0 ]]
then
        canvi=true
fi

# com hi ha hagut un canvi substituim els ports per els nous ports canviats
cat nouPorts.txt > portsIp.txt

#Si hi ha hagut canvi ens avisa del canvi i ens permet conectar-nos al servidors per a executar comm
andes
if [[ $canvi == true ]]
then
        # aqui es on hi hauria d'haver el ssmtp avisant del canvi, no ho hem implementat per que no
podem grabar-ho com caldría, si veiem que s'executa "./interroga" vol dir que hi ha hagut canvi i re
briam el correu que haguessim prefet en un altre arxiu .txt

        echo "Hi ha hagut un canvi, enviant correu a l'administrador"
        /avisMail.sh
        sleep 1

        # per a cada ip en l'arxiu dels ports creat en el pas anterior ho passarem com a argument al
 script interroga
        for ip in $(cat portsIp.txt | awk '{print $1}' | sort -u)
        do
                ./interroga.sh $ip
        done
fi
```

Imatge de portsIp.txt <u>posterior</u> a evaluaPort.sh amb ports actualitzats

```
daniel@monitor:~/monitor$ cat portsIp.txt
10.0.2.108 22 close
10.0.2.108 1234 close
10.0.2.108 2222 close
10.0.2.108 4444 close
10.0.2.110 22 close
10.0.2.110 1234 close
10.0.2.110 4444 close
10.0.2.110 6969 close
10.0.2.111 22 open
10.0.2.111 420 close
10.0.2.111 1234 close
10.0.2.111 4444 open
```

<u>Nota:</u> Aquesta captura s'ha realitzat posterior al vídeo ja que vam oblidar de ensenyar-la, aquest cas es un evaluaPorts.sh de nomes <u>1 servidor agent obert</u>, on demostrem que tots els ports estan tancats per als servidors no disponibles i només vam obrir els ports 22 i 4444 (per rebre la connexió ncat) en el servidor 10.0.2.111.

Principalment demostra que el script evaluaPorts.txt actualitza el document portsIP.txt si hi ha canvis i reflecteix els canvis que hi ha hagut respecte a la foto inicial, permetent posteriorment a través del propi evaluaPorts.sh determinar si hi ha hagut canvis i necessitem establir una connexió amb els servidors agents.

3. interroga.sh

El funcionament es ben senzill en aquest cas. Donat un paràmetre IP, ens connectarem remotament al port 4444 del paràmetre, **on prèviament hem col·locat l'agent.sh**, que veurem en l'apartat següent.

```
#!/bin/bash

# Interficie "dinamica" per a conectarnos al servior / ip que passem com a argument i poder enviar c
omandes predefinides

clear
echo -e "Conectant a la IP $1"
sleep 1
clear
echo -e "Conectant a la IP $1."
sleep 1
clear
echo -e "Conectant a la IP $1.."
sleep 1
clear
echo -e "Conectant a la IP $1..."
sleep 1
clear
sleep 1

ncat -nv $1 4444 2>/dev/null
```

Servidor Agent interrogat:

Prèviament, als servidors interrogats (contenent agent.sh) els hi haurem d'obrir el port 4444 i afegir una comanda per **escoltar amb ncat** fent que **executi l'agent.sh** a l'hora de confirmar que ha rebut una connexió exitosa.

Afegim aquesta línia al final de l'arxiu ***.bashrc*** del usuari "**root**" per tal que el port només sigui obert pel propi administrador del sistema un cop ha fet login (la opció -e permet fer que l'interrogador executi l'agent, situat també en una carpeta de root.)

Comanda emprada per a la bind Shell a .bashrc:

```
ncat -knlvp 4444 -e /root/agent.sh
```

4. agent.sh

El propòsit d'aquest script es rebre la informació que entra pel port 4444 des del servidor de monitorització, proveint d'un menú comprensible i fàcil de fer servir amb instruccions, i aquest servidor agent ser capaç de fer sevir una de les instruccions per extreure la informació pertinent

```
#!/bin/bash
clear

# Menu interactiu per a poder enviar comandes al servidor corresponent
echo -e "\n[+] Benvingut al menu principal de $HOSTNAME, siusplau escull una opcio:"
echo -e "\n\t- mem - Per a observar la memoria dels 10 procesos principals."
echo -e "\t- file - Per a poder comprovar els filesystems."
echo -e "\t- cpu - Per a analitzar el consum de la CPU per als 10 primers processos."

while true
do
        #read -p $'\nSiusplau introdueix una comanda:\n'$HOSTNAME" >> " comanda
        echo -en "\nSiusplau introdueix una comanda:\n$HOSTNAME >> "
        read comanda


        # si la comanda es mem extraurem el % de consum de memoria dels 10 procediments mes als
        if [[ $comanda == "mem" ]]
        then
                clear
                echo ""
                ps aux | sort -rk 4 | awk '{print $1"\011\011"$2"\011\011"$4"\011\011"$11}' | head -
11
        # si la comanda es file obtindrem els filesystems del servidor
        elif [[ $comanda == "file" ]]
        then
                clear
                echo ""
                df -h
        # si la comanda es cpu obtimdren el % de consum de CPU dels 10 proceso mes alts
        elif [[ $comanda == "cpu" ]]
        then
                clear
                echo ""
                ps aux | sort -rk 3 | awk '{print $1"\011\011"$2"\011\011"$3"\011\011"$11}' | head -
11
```

```
        # amb exit sortirem del programa poden seguir al següent server
        elif [[ $comanda == "exit" ]]
        then
                clear
                echo -e "\nExiting"
                sleep 1
                clear
                echo -e "\nExiting."
                sleep 1
                clear
                echo -e "\nExiting.."
                sleep 1
                clear
                echo -e "\nExiting..."
                sleep 1
                clear
                exit 0

        else
                clear
                echo "Comanda incorrecta siusplau fes servir una de les següents comandes:"
                echo ""
                echo -e "\n\t- mem - Per a observar la memoria dels 10 procesos principals."
                echo -e "\t- file - Per a poder comprovar els filesystems."
                echo -e "\t- cpu - Per a analitzar el consum de la CPU per als 10 primers processos.
"
        fi
done
```

Segons s'inicií l'agent, aquest ens obrirà un menú per a cada servidor amb les següents opcions de visualització:

```
[+] Benvingut al menu principal de server1, siusplau escull una opcio:

        - mem - Per a observar la memoria dels 10 procesos principals.
        - file - Per a poder comprovar els filesystems.
        - cpu - Per a analitzar el consum de la CPU per als 10 primers processos.

Siusplau introdueix una comanda:
server1 >>
```

Comanda "mem": Aquesta comanda ens proporciona informació referent als 10 primers processos que ocupen més memòria en el servidor mencionat.

```
USER            PID             %MEM            COMMAND
root            875             1.9             /usr/lib/snapd/snapd
root            554             1.3             /sbin/multipathd
root            1572            1.0             /usr/libexec/packagekitd
root            917             1.0             /usr/bin/python3
root            871             0.9             /usr/bin/python3
root            1790            0.8             python3
root            1775            0.8             python3
root            505             0.7             /lib/systemd/systemd-journald
root            879             0.6             /usr/libexec/udisks2/udisksd
root            1               0.6             /sbin/init
```

Comanda "file": Aquesta comanda ens dona els diferents "filesystems" que existeixen en el servidor connectat.

```
Filesystem                         Size  Used Avail Use% Mounted on
tmpfs                              195M  1,3M  193M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv  9,8G  3,7G  5,6G  40% /
tmpfs                              971M     0  971M   0% /dev/shm
tmpfs                              5,0M     0  5,0M   0% /run/lock
/dev/sda2                          1,7G  247M  1,4G  16% /boot
tmpfs                              195M  4,0K  195M   1% /run/user/1000
```

Comanda "cpu": Aquesta comanda ens mostrarà els 10 primers processos que fan més ús del processador del nostre servidor mencionat.

```
USER            PID             %CPU            COMMAND
root            1402            0.3             [kworker/1:1-pm]
root            1               0.1             /sbin/init
root            703             0.1             /usr/bin/vmtoolsd
root            28              0.1             [kworker/0:2-events]
root            875             0.0             /usr/lib/snapd/snapd
root            554             0.0             /sbin/multipathd
root            1572            0.0             /usr/libexec/packagekitd
root            917             0.0             /usr/bin/python3
root            871             0.0             /usr/bin/python3
root            1790            0.0             python3
```

En cas d'**error**:

```
Comanda incorrecta siusplau fes servir una de les següents comandes:


        - mem - Per a observar la memoria dels 10 procesos principals.
        - file - Per a poder comprovar els filesystems.
        - cpu - Per a analitzar el consum de la CPU per als 10 primers processos.

Siusplau introdueix una comanda:
server1 >>
```

Ens avisa de l'error i ens retorna la informació necessària per a fer servir el script.

5. avisMail.sh

En producció, s'envia un correu mitjançant un arxiu/mòdul extern als monitoritza-dors/agents que podrem canviar de forma independent sense perjudicar el codi font dels processos amb la següent rutina. Aquest avís ens informa d'una modificació en el ports respecte la "fotoinicial", juntament de la data on ha ocorregut aquest mateix:

```
#!/bin/bash

cat /dev/null > message.txt

echo -e "To:yautja@telefonica.net" > message.txt

echo -e "From:marc.salvador@gracia.lasalle.cat" >> message.txt

echo -e "Subject:Canvis en servidors |n" >> message.txt

echo -e "ALERTA - Canvis en els ports respecte a la situació inicial. Siusplau conectat als diferent
s servidorsa gent per a coneixer l'ocurregut e: `date+'%Y/%m/%d'`" >> message.txt

ssmtp yautja@telefonica.net < message.txt
```

Comentaris extra:

En el vídeo següent no demostrem que es pugui tornar a connectar un cop s'ha connectat però es veu clarament en les finestres de cada servidor agent que les connexions no es tanquen quan fem servir la comanda "exit" i això es gracies a la flag "-k" de ncat feta servir al .bashrc de root, lo qual ens permet tornar a executar el script evaluaPorts.sh i si hi ha hagut algun canvi (en els ports que hem fet la foto inicial) iniciar una connexió.

(L'error que es veu en el vídeo de que he de fer un enter de mes va ser literalment un read de més en el script del servidor 2 que vam solucionar borrant-lo)

Vídeo

En l'enllaç següent hem pujat un vídeo demostrant els passos a seguir i provant que funciona, explicant per sobre cada un d'ells:

Vídeo Monitorització de servidors amb Bash - Marc Aleix Daniel