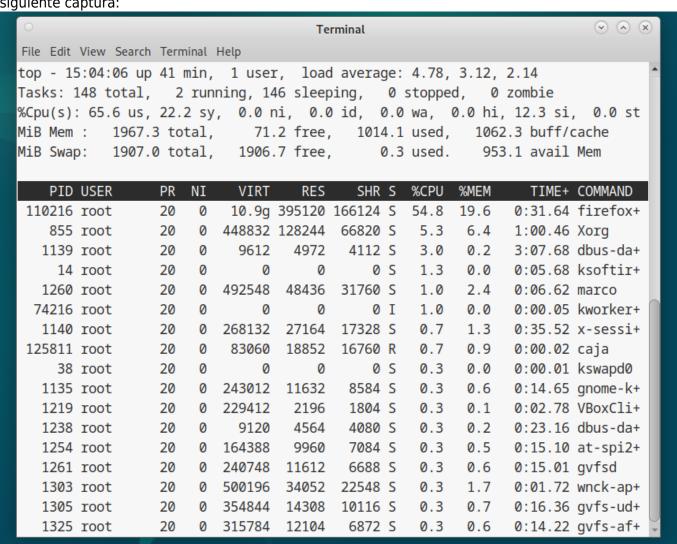
[Fort]Práctica 3: Securizando Aplicaciones

1. Abre un navegador y realiza una descarga

Revisa con el TOP el consumo de CPU

Firefox (Proceso 110216) está consumiendo en torno al 60% de la CPU como podemos ver en la siguiente captura:



Usa cpulimit para reducir el consumo de CPU a 1/5 del que usa

Para realizar esto ejecutaremos los siguientes comandos:

```
cd /sys/fs/cgroup/
mkdir firefox #Creamos el cgroup para firefox
cd firefox
```

```
root@fso2025:~/Downloads# cd /sys/fs/cgroup/
root@fso2025:/sys/fs/cgroup# mkdir firefox
root@fso2025:/sys/fs/cgroup# cd firefox/
root@fso2025:/sys/fs/cgroup/firefox# ls
cgroup.controllers
                        cpuset.cpus.effective
                                                memory.min
cgroup.events
                        cpuset.cpus.partition
                                                memory.numa_stat
cgroup.freeze
                        cpuset.mems
                                                memory.oom.group
cgroup.kill
                        cpuset.mems.effective
                                                memory.peak
cgroup.max.depth
                        cpu.stat
                                                memory.pressure
cgroup.max.descendants
                        cpu.weight
                                                memory.reclaim
                        cpu.weight.nice
cgroup.pressure
                                                memory.stat
cgroup.procs
                        io.max
                                                memory.swap.current
cgroup.stat
                        io.pressure
                                                memory.swap.events
                        io.stat
cgroup.subtree_control
                                                memory.swap.high
cgroup.threads
                        io.weight
                                                memory.swap.max
cgroup.type
                        memory.current
                                                memory.zswap.current
cpu.idle
                        memory.events
                                                memory.zswap.max
cpu.max
                        memory.events.local
                                                pids.current
cpu.max.burst
                        memory.high
                                                pids.events
cpu.pressure
                        memory.low
                                                pids.max
                                                pids.peak
cpuset.cpus
                        memory.max
root@fso2025:/sys/fs/cgroup/firefox#
```

Tras eso añadimos firefox al cgroup y procedemos a limitar su consumo de CPU

```
echo 110216 > cgroup.procs#Metemos el proceso de firefox en el cgroup echo 200000 1000000 > cpu.max #Limitamos el uso de CPU al 20%
```

```
root@fso2025:/sys/fs/cgroup/firefox# echo 110216 > cgroup.procs
root@fso2025:/sys/fs/cgroup/firefox# echo 200000 1000000 > cpu.max
root@fso2025:/sys/fs/cgroup/firefox#
```

Resultado

Como se puede observar, el consumo ha bajado y ahora consume un 20% máximo.

top - 15:17:43 up 54 min, 1 user, load average: 0.08, 0.51, 1.27
Tasks: 145 total, 1 running, 144 sleeping, 0 stopped, 0 zombie
%Cpu(s): 12.8 us, 4.7 sy, 0.0 ni, 79.1 id, 0.0 wa, 0.0 hi, 3.4 si, 0.0 st
MiB Mem : 1967.3 total, 700.8 free, 1033.6 used, 439.1 buff/cache
MiB Swap: 1907.0 total, 1865.5 free, 41.5 used. 933.7 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
110216	root	20	0	3023872	337904	112236	S	20.3	16.8	3:16.39	firefox+
855	root	20	0	490648	109572	51580	S	0.7	5.4	1:15.46	Xorg
14	root	20	0	0	0	0	S	0.3	0.0	0:08.80	ksoftir+
15	root	20	0	0	0	0	Ι	0.3	0.0	0:05.32	rcu_pre+
1209	root	20	0	228896	1548	1276	S	0.3	0.1	0:01.28	VBoxCli+
1219	root	20	0	229412	1484	1212	S	0.3	0.1	0:04.24	VBoxCli+
1	root	20	0	103904	11088	6424	S	0.0	0.6	0:04.73	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	Ι	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	Ι	0.0	0.0	0:00.00	rcu_par+
5	root	0	-20	0	0	0	Ι	0.0	0.0	0:00.00	slub_fl+
6	root	0	-20	0	0	0	Ι	0.0	0.0	0:00.00	netns
7	root	20	0	0	0	0	Ι	0.0	0.0	0:01.66	kworker+
10	root	0	-20	0	0	0	Ι	0.0	0.0	0:00.00	mm_perc+
11	root	20	0	0	0	0	Ι	0.0	0.0	0:00.00	rcu_tas+
12	root	20	0	0	0	0	Ι	0.0	0.0	0:00.00	rcu_tas+
13	root	20	0	0	0	0	Ι	0.0	0.0	0:00.00	rcu_tas+

2. Crea un container Debian y arráncalo

Para arrancar un container debian usamos el siguiente comando:

lxc-create -t debian -n deb

Establece una contraseña para el usuario root del container

Para ello debemos entrar primero a la máquina con el siguiente comando para abrir la terminal del contenedor:

lxc-start #Arrancamos la máquina
lxc-attach -n deb /bin/sh #Llamamos a la terminal

```
root@fso2025:~# lxc-st
lxc-start lxc-stop
root@fso2025:~# lxc-start -n deb
root@fso2025:~# lxc-attach -n deb /bin/sh
#
```

Tras eso procedemos a establecer la contraseña del root con el siguiente comando:

```
root@fso2025:~# lxc-start -n deb
root@fso2025:~# lxc-attach -n deb /bin/sh
# passwd root
New password:
Retype new password:
passwd: password updated successfully
#
```

Añade 3 usuarios

Desde el mismo sitio que hemos establecido la contraseña para el root procedemos a crear los 3 usuarios:

```
su - root
useradd -m -s /bin/bash usuario1
useradd -m -s /bin/bash usuario2
useradd -m -s /bin/bash usuario3

# su - root
root@deb:~# sudo useradd -m -s /bin/bash usuario1
-bash: sudo: command not found
root@deb:~# useradd -m -s /bin/bash usuario1
root@deb:~# useradd -m -s /bin/bash usuario2
root@deb:~# useradd -m -s /bin/bash usuario3
root@deb:~#
```

Instala apache 2 y ssh en el container

Comenzamos instalando apache 2:

```
root@deb:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
The following additional packages will be installed:
 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libbrotli1
 libcurl4 libexpat1 libgdbm-compat4 libgdbm6 libicur2 libjansson4 libldap-2.5-0 libldap-common liblua5.3-0
 libnghttp2-14 libperl5.36 libpsl5 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libsqlite3-0
 libssh2-1 libxml2 media-types perl perl-modules-5.36 publicsuffix ssl-cert
Suggested packages:
 apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser gdbm-l10n libsasl2-modules-gssapi-mit
  | libsasl2-modules-gsapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql perl-doc
 libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
 apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
 libbrotli1 libcurl4 libexpat1 libgdbm-compat4 libgdbm6 libicur2 libjansson4 libldap-2.5-0 libldap-common
 liblua5.3-0 libnghttp2-14 libper15.36 libps15 librtmp1 libsas12-2 libsas12-modules libsas12-modules-db
 libsqlite3-0 libssh2-1 libxml2 media-types perl perl-modules-5.36 publicsuffix ssl-cert
ð upgraded, 33 newly installed, 0 to remove and 0 not upgraded.
Need to get 22.3 MB of archives.
```

Tras eso procedemos a instalar SSH:

```
root@deb:~# apt install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 174 kB of archives.
After this operation, 187 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stable/main amd64 ssh all 1:9.2p1-2+deb12u4 [174 kB]
Fetched 174 kB in 0s (946 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package ssh.
(Reading database ... 12070 files and directories currently installed.)
Preparing to unpack .../ssh_1%3a9.2p1-2+deb12u4_all.deb ...
Unpacking ssh (1:9.2p1-2+deb12u4) ...
Setting up ssh (1:9.2p1-2+deb12u4) ...
root@deb:~#
```

Cambia la dirección de red en el container para usar una dirección de red estática

Para establecer una IP estática usamos el siguiente comando:

```
sudo ifconfig eth0 10.0.3.68 netmask 255.255.25 up
```

```
root@deb:~# ifconfig eth0 10.0.3.68 netmask 255.255.255.0 up
root@deb:~# ping google.es
PING google.es (142.250.200.131) 56(84) bytes of data.
64 bytes from mad41s14-in-f3.1e100.net (142.250.200.131): icmp_seq=1 ttl=114 time=26.4 ms
64 bytes from mad41s14-in-f3.1e100.net (142.250.200.131): icmp_seq=2 ttl=114 time=26.8 ms
^C
--- google.es ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 26.444/26.641/26.839/0.197 ms
root@deb:~#
```

Cambial el puerto de SSH al 222

Vamos a la ruta /etc/ssh y modificamos el archivo sshd config:

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
Include /etc/ssh/sshd_config.d/*.conf
Port 222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
                                ^W Where Is
  Help
                ^O Write Out
                                                 ^K Cut
                                                                    Execute
                                                                                    Location
                                                                                                  M-U Undo
  Exit
                   Read File
                                   Replace
                                                 ^U Paste
                                                                    Justify
                                                                                    Go To Line
                                                                                                     Redo
```

Modifica el archivo de configuración del container

```
/var/lib/lxc/deb/config *
 GNU nano 7.2
# (Be aware this has security implications)
lxc.net.0.type = veth
lxc.net.0.hwaddr = 00:16:3e:3b:74:2a
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.rootfs.path = dir:/var/lib/lxc/deb/rootfs
# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf
# Container specific configuration
lxc.tty.max = 4
lxc.uts.name = deb
lxc.arch = amd64
lxc.pty.max = 1024
lxc.start.auto = 1
lxc.start.delay = 3
```

Crea un cgroup

Creamos un nuevo csgroup con el siguiente comando:

```
cd /sys/fs/cgroup/
mkdir grupo #Creamos el nuevo cgroup
cd grupo
```

root@fso2025:/sys/fs# cd cgroup/ root@fso2025:/sys/fs/cgroup# mkdir grupo root@fso2025:/sys/fs/cgroup# cd grupo/ root@fso2025:/sys/fs/cgroup/grupo# ls cgroup.controllers cpu.stat memory.numa_stat cgroup.events cpu.weight memory.oom.group cgroup.freeze cpu.weight.nice memory.peak cgroup.kill hugetlb.2MB.current memory.pressure cgroup.max.depth hugetlb.2MB.events memory.reclaim cgroup.max.descendants hugetlb.2MB.events.local memory.stat hugetlb.2MB.max cgroup.pressure memory.swap.current hugetlb.2MB.numa_stat cgroup.procs memory.swap.events hugetlb.2MB.rsvd.current cgroup.stat memory.swap.high cgroup.subtree_control hugetlb.2MB.rsvd.max memory.swap.max cgroup.threads io.max memory.zswap.current cgroup.type io.pressure memory.zswap.max cpu.idle io.stat misc.current cpu.max io.weight misc.events cpu.max.burst memory.current misc.max cpu.pressure memory.events pids.current cpuset.cpus memory.events.local pids.events cpuset.cpus.effective memory.high pids.max cpuset.cpus.partition memory.low pids.peak rdma.current cpuset.mems memory.max cpuset.mems.effective rdma.max memory.min root@fso2025:/sys/fs/cgroup/grupo#

Abre un terminal y añade el shell de este al cgroup

echo \$\$ #0btenemos el ID del proceso del Shell (En este caso 1709) echo 1709 > cgroup.procs #Añadimos el proceso del shell al gproup

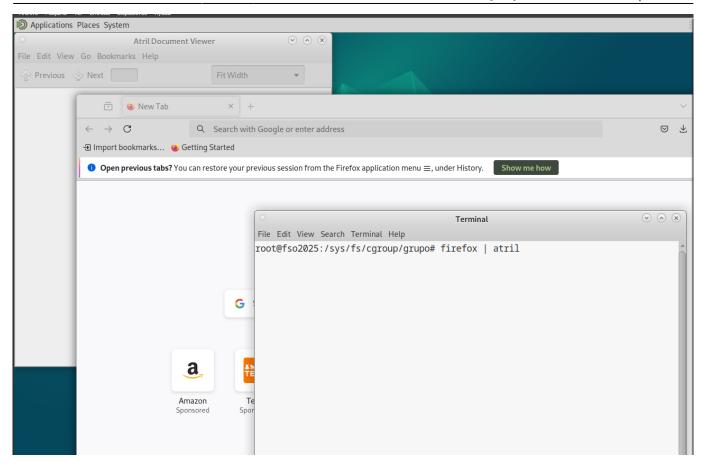
root@fso2025:/sys/fs/cgroup/grupo# echo \$\$

1709

root@fso2025:/sys/fs/cgroup/grupo# echo 1709 > cgroup.procs

root@fso2025:/sys/fs/cgroup/grupo#

Desde ese mismo shell abre firefox y atril



Revisa el contenido de cgroup.procs y memory.current

```
root@fso2025:/sys/fs/cgroup/grupo# cat cgroup.procs
1709
17940
17941
18036
18060
18140
18229
18238
18245
18252
root@fso2025:/sys/fs/cgroup/grupo# cat memory.current
556228608
root@fso2025:/sys/fs/cgroup/grupo#
```

Pon un 1 en cgroups.freeze ¿Que pasa? ¿Que hay ahora en memory.current?

Los programas que abrimos antes quedan congelados. El contenido de memory.current es ahora este:

root@fso2025:/sys/fs/cgroup/grupo# echo 1 > cgroup.freeze
root@fso2025:/sys/fs/cgroup/grupo# cat memory.current
555556864
root@fso2025:/sys/fs/cgroup/grupo# echo 1 > cgroup.freeze

Este valor que podemos observar es la memoria consumida por los procesos de cgroup en bytes. Ahora el valor parece ser superior al que vimos anteriormente. Tras esto descongelamos el proceso poniendo un 0 en groups.freeze

Pon los valores 30000 100000 en cpu.max ¿Que pasa?

root@fso2025:/sys/fs/cgroup/grupo# echo 30000 100000 > cpu.max
root@fso2025:/sys/fs/cgroup/grupo#

Estamos limitando cuanto pueden consumir los procesos del cgroup, en este caso estamos limitando los procesos a un 30%

From:

http://knoppia.net/ - Knoppia

Permanent link:

http://knoppia.net/doku.php?id=master cs:fortificacion:p3&rev=1739891727

Last update: 2025/02/18 15:15

