CS 370, Assignment 3 Report

Cody Malick
malickc@oregonstate.edu

November 30, 2016

Question 1

Steps taken:
1. used chmod u-s /usr/bin/passwd 2. Tested passwd again to see what permissions were needed Error received: passwd: Authentication token manipulation error 3. Researched what capabilities passwd needs under the posix capabilities system 4. Capbilities needed: cap_chown, cap_dac_override, cap_fowner, all needed to be set to ep

From the man pages, the following capabilities grant the following permissions:

cap_chown: Make arbitrary changes to file UIDs and GIDs cap_dac_override: Bypass file read, write, and execute permission checks. cap_fowner: Bypass permission checks on operations that normally require the filesystem UID of the process to match the UID of the file. Set extended file attributes on arbitrary files. Set Access Control Lists on arbitrary files.

Question 2

cap_dac_read_search: Bypass file read permissions checks and directory read and execute permission cehcks.

We can demonstrate this capability by removing the 'ls' command's set-UID capability using:

Listing 1: Removing set-UID capability on ls

```
root@code-virtual-machine: chmod u-s /bin/ls
```

Then, we can test out the lack of set-UID capability by running ls on a directory that requires root access, such as /root:

```
code@code-virtual-machine:~/git/compsci/370/3\$ ls /root
ls: cannot open directory '/root': Permission denied
```

Now, we can add the cap_dac_read_search capability to ls:

```
code@code-virtual-machine:~/git/compsci/370/3\$ sudo setcap cap_dac_read_search=ep /bin/ls
code@code-virtual-machine:~/git/compsci/370/3\$ ls -a /root
.  ..  .bash_history  .bashrc  .bzr.log  .cache  .profile
```

We've demonstrated that, using the cap_dac_read_search capability, we can remove the need for ls to run at an elevated level.

Question 3

Question 4

Question 5

Question 6