

CMPUT 333

SECURITY IN A

NETWORKED WORLD

**LAB ASSIGNMENT 1 : INTRODUCTION TO
SCREEN**

PROBLEM

- **Cracking passwords/ciphers/etc. typically takes a lot of computation time**
- **Can you run a program (like Jack the Ripper — See Assignment 1 Part 4) for a long time without**
 - **tying up personal machine**
 - **having to maintain an ssh connection to an external machine**

We need a new tool!

NEW TOOL: SCREEN

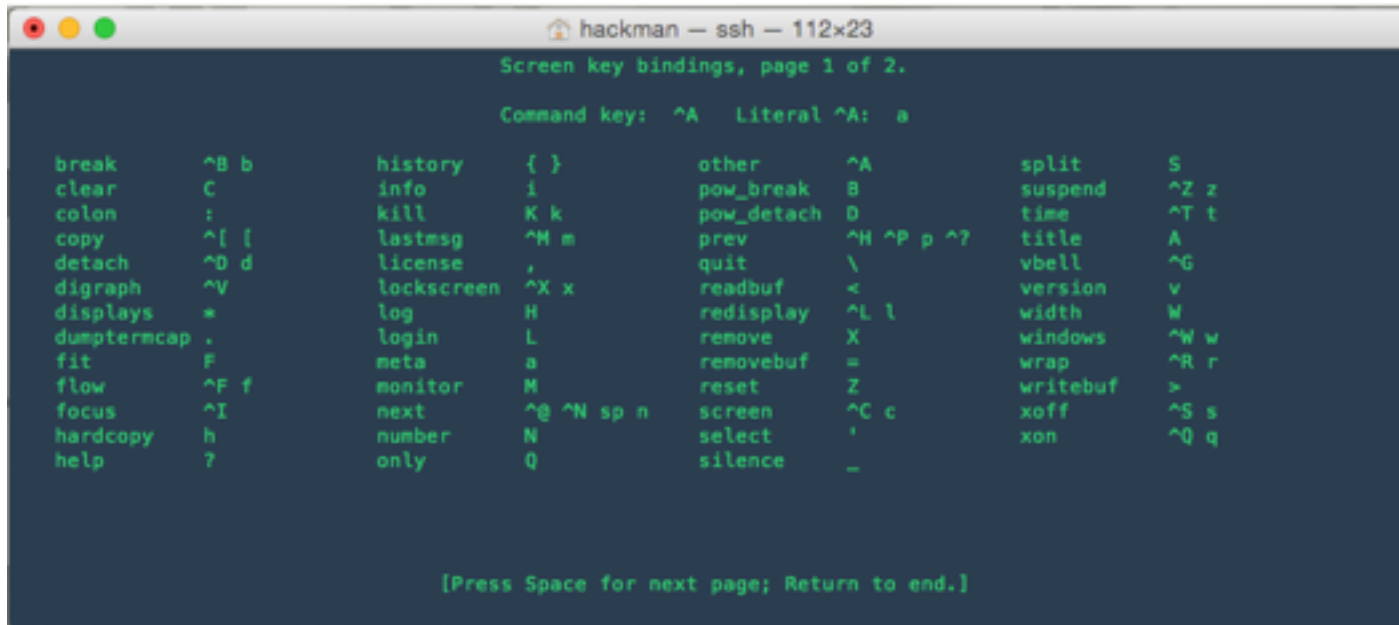
- **Screen** is a program for Unix-like systems (ie. the lab machines)
- Allows you to multiplex your terminal
 - from one terminal, you can create other terminal sessions
 - you can **detach and reattach** these terminal sessions at any time
 - detached terminal sessions **persist** even when the original terminal session ends
 - a new terminal session can **reattach** to detached terminals later

RUNNING SCREEN

- In a terminal, run the command **screen**
- You may get some splash text telling you that you are now running screen. Pushing any key will give you a command line
- it may look like you are where you started but you are now in a *new* screen terminal session
- From here, you can run commands like you normally would. But you also have the power of screen at your fingertips.

GETTING HELP

- All screen commands start with **Control-a** (We will notate this as **^A** as is commonly done)
- To start, you can type **^A ?** for help. This will show a list of screen commands
- Do not keep holding Control after pushing a



The screenshot shows a terminal window titled "hackman — ssh — 112x23". The terminal displays the output of the command `^A ?`, which lists screen key bindings. The text is as follows:

```
Screen key bindings, page 1 of 2.

Command key: ^A  Literal ^A: a

break      ^B b      history   { }      other     ^A        split     S
clear      C        info      i         pow_break B        suspend  ^Z z
colon      :        kill      K k      pow_detach D      time      ^T t
copy       ^[ [      lastmsg   ^M m     prev      ^H ^P p ^?  title     A
detach     ^D d      license   ,        quit      \        vbell     ^G
digraph    ^V        lockscreen ^X x     readbuf    <        version   v
displays   *        log       H        redisplay  ^L l     width     W
dumftermcap .      login     L        remove     X        windows   ^W w
fit        F        meta      a        removebuf  =        wrap      ^R r
flow       ^F f      monitor   M        reset      Z        writebuf  >
focus     ^I        next      ^@ ^N sp n screen    ^C c     xoff      ^S s
hardcopy   h        number    N        select     '        xon       ^Q q
help       ?        only      Q        silence    _

[Press Space for next page; Return to end.]
```

CREATING WINDOWS

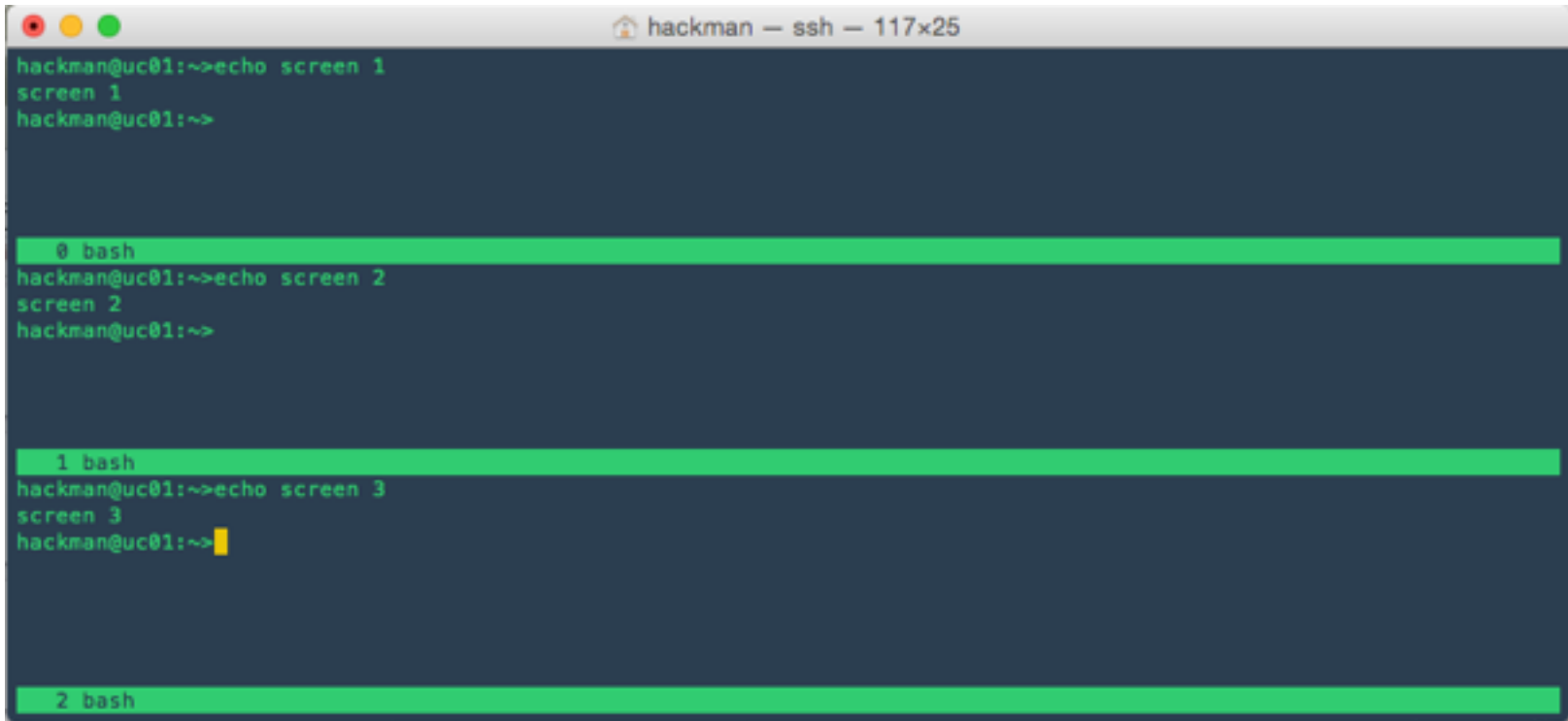
- Can create a new terminal window by typing **^A c** (c is for create)
- To switch between terminals type **^A n** (n is for next)
- Try echoing different text on the screen in each terminal and switching between them to see that you do in fact have multiple terminals.

NAVIGATING

- View a list of all windows with command **^A w**
- Other useful navigation commands:
 - **^A “** gives you a navigable list of windows
 - **^A 0** to **^A 9** let you flip to a specific window

SPLITTING WINDOWS

- Can split a window to show several windows at the same time but this is beyond the scope of this little tutorial.



```
hackman@uc01:~>echo screen 1
screen 1
hackman@uc01:~>

0 bash
hackman@uc01:~>echo screen 2
screen 2
hackman@uc01:~>

1 bash
hackman@uc01:~>echo screen 3
screen 3
hackman@uc01:~>

2 bash
```

The image shows a terminal window titled 'hackman — ssh — 117x25'. It demonstrates the process of splitting a terminal window into three horizontal panes. The first pane shows the command 'echo screen 1' and its output 'screen 1'. The second pane, labeled '0 bash', shows 'echo screen 2' and 'screen 2'. The third pane, labeled '1 bash', shows 'echo screen 3' and 'screen 3'. A fourth pane at the bottom is labeled '2 bash' and is currently empty. Each pane has a green header bar with its label.

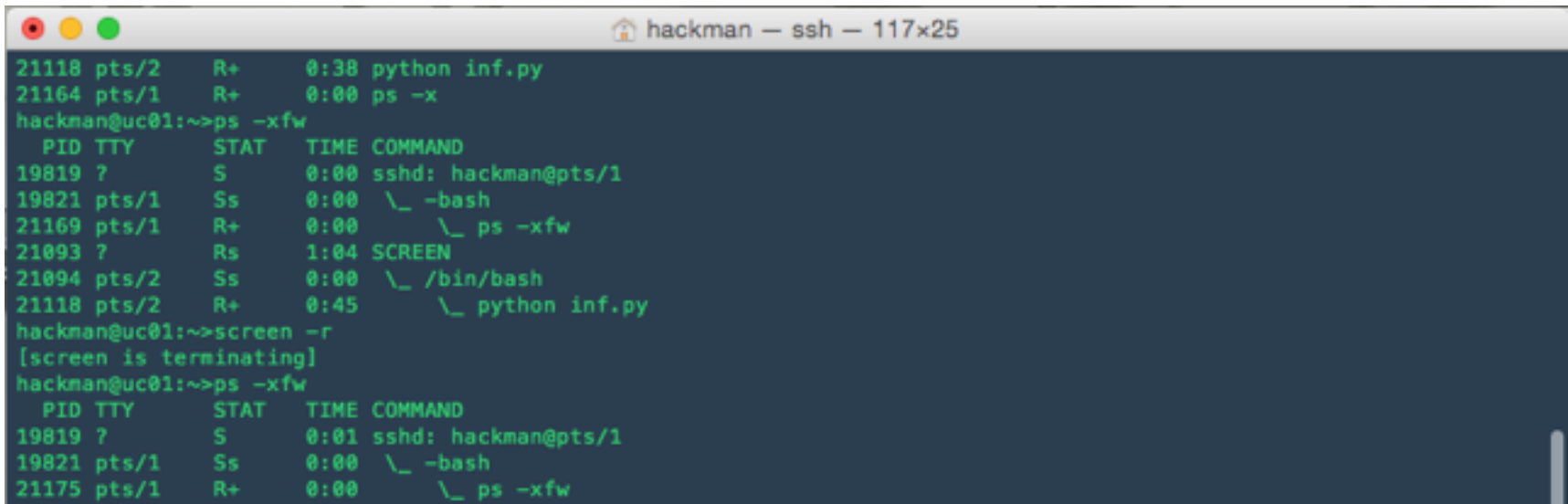
EXITING SCREEN

- to get rid of a screen window type **exit** or use the **^A k** command (this will ask you to confirm that you want to kill the window)
- If you exit or kill all your open windows, your screen session will automatically exit, returning you to your original terminal session

DETACHING

- When in screen, you can detach your screen session (which contains all the windows you created/opened) using the **^A d** command
- This takes you back to your initial terminal session. **BUT your screen session persists and is still running on the machine**
- Use command **ps -xf** to see that your screenwindows are still there

VIEWING DETACHED SCREENS

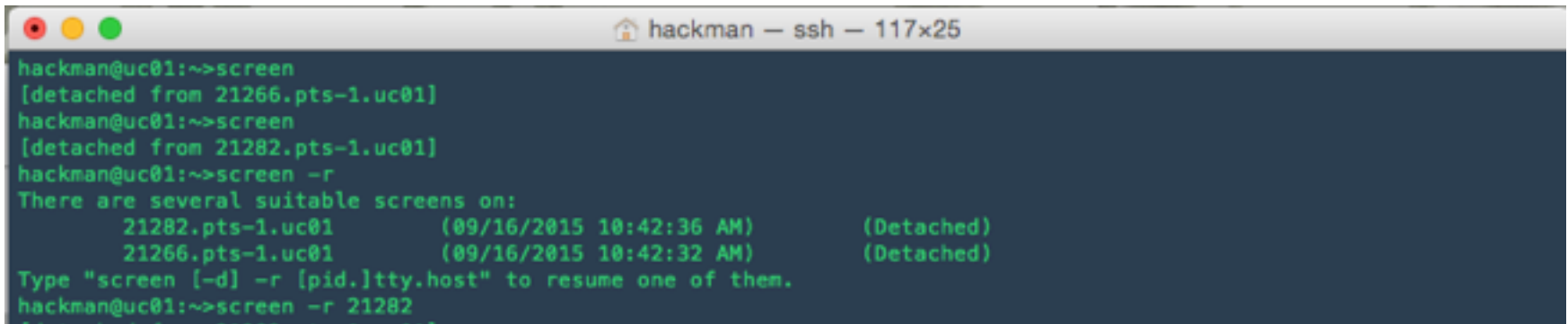


A terminal window titled "hackman — ssh — 117x25" displays the following commands and output:

```
21118 pts/2    R+      0:38 python inf.py
21164 pts/1    R+      0:00 ps -x
hackman@uc01:~>ps -xfw
  PID TTY          STAT       TIME COMMAND
19819 ?            S          0:00 sshd: hackman@pts/1
19821 pts/1        Ss         0:00  \_ -bash
21169 pts/1        R+         0:00      \_ ps -xfw
21093 ?            Rs         1:04 SCREEN
21094 pts/2        Ss         0:00  \_ /bin/bash
21118 pts/2        R+         0:45      \_ python inf.py
hackman@uc01:~>screen -r
[screen is terminating]
hackman@uc01:~>ps -xfw
  PID TTY          STAT       TIME COMMAND
19819 ?            S          0:01 sshd: hackman@pts/1
19821 pts/1        Ss         0:00  \_ -bash
21175 pts/1        R+         0:00      \_ ps -xfw
```

RE-ATTACHING

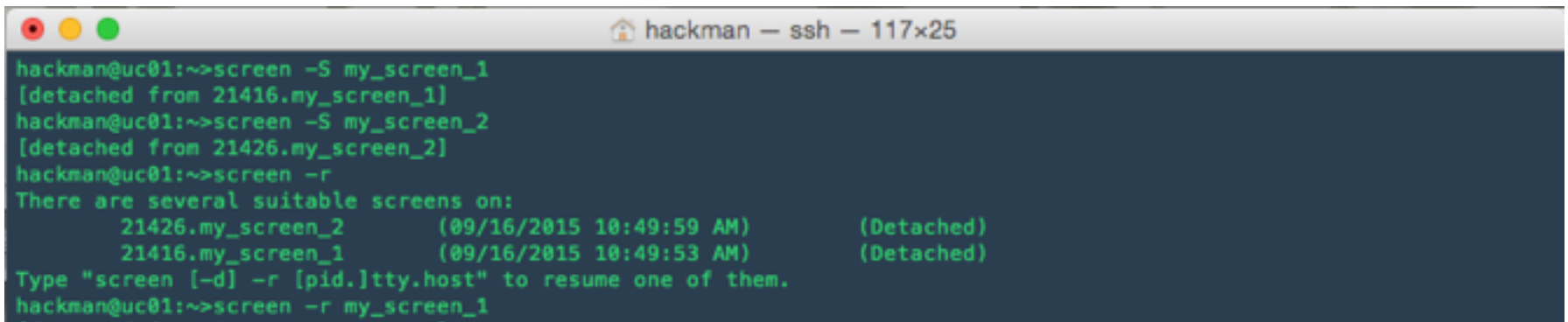
- You can re-attach to a detached screen session by running the command: **screen -r**
- If you have multiple screen sessions running, it will list all available screen sessions. Attach to a specific one by running **screen -r** and provide an **id** for the process

A terminal window titled 'hackman - ssh - 117x25' with a dark blue background and green text. The user 'hackman@uc01' runs 'screen' twice, each time getting '[detached from ...]'. Then they run 'screen -r', which lists two detached sessions: '21282.pts-1.uc01' and '21266.pts-1.uc01'. The user then runs 'screen -r 21282' to re-attach to the first session.

```
hackman@uc01:~>screen
[detached from 21266.pts-1.uc01]
hackman@uc01:~>screen
[detached from 21282.pts-1.uc01]
hackman@uc01:~>screen -r
There are several suitable screens on:
      21282.pts-1.uc01      (09/16/2015 10:42:36 AM)      (Detached)
      21266.pts-1.uc01      (09/16/2015 10:42:32 AM)      (Detached)
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
hackman@uc01:~>screen -r 21282
```

NAME YOUR SCREENS

- That if you frequently have multiple screens open you can name your screen sessions so it is easier to tell them apart
- Note that you won't need multiple screen session for this class but it can be useful. E.g.: a sys admin might have multiple screen sessions that correspond to different frequent monitoring tasks



```
hackman@uc01:~>screen -S my_screen_1
[detached from 21416.my_screen_1]
hackman@uc01:~>screen -S my_screen_2
[detached from 21426.my_screen_2]
hackman@uc01:~>screen -r
There are several suitable screens on:
  21426.my_screen_2      (09/16/2015 10:49:59 AM)      (Detached)
  21416.my_screen_1      (09/16/2015 10:49:53 AM)      (Detached)
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
hackman@uc01:~>screen -r my_screen_1
```

NAME YOUR WINDOWS

- You can also name your windows with the command **^A a**
- Again, this is likely more advanced than you need right now, but is useful to be aware of for when you do need it in the future.

PUT IT ALL TOGETHER

- **Now that you know about screen, you know that can:**
 - **log into your lab machine (in person, or remotely via ssh)**
 - **open a screen session**
 - **run your programs for breaking a password or cracking a cipher etc.**
 - **detach the screen session**
 - **log out (or terminate the ssh connection)**
 - **come back later and reattach**

And hopefully by the time you come back, all your work will be done!

REMOTE CONNECTIONS

- SSHing to CS in general: <https://uofa.ualberta.ca/computing-science/links-and-resources/technical-support/computing-resources>
- Want to SSH to your specific lab machine
- Lab machine IDs uc01...uc16: <https://uofa.ualberta.ca/computing-science/links-and-resources/labs-and-room-booking/csc-1-21>

USEFUL TUTORIALS

- <http://www.bangmoney.org/posts/2004-03-24-screen.html>
- <https://www.rackaid.com/blog/linux-screen-tutorial-and-how-to/>

QUESTIONS ?