

A close-up photograph of a thicket of bare, snow-laden branches. The branches are dark and intricate, creating a complex web against a pale, overcast sky. Small, bright red berries are scattered throughout the branches, providing a sharp contrast to the white snow and dark wood. The overall mood is cold and wintry.

CSCI 3330 Cybersecurity

Word of the day

HACK MODE. n.

1. What one is in when hacking, of course.
2. More specifically, a Zen-like state of total focus on The Problem that may be achieved when one is hacking (this is why every good hacker is part mystic). Ability to enter such concentration at will correlates strongly with wizardliness; it is one of the most important skills learned during larval stage.

Agenda for today

- BCQs
- Chroot
- PIE (position independent executable)
- Final assignment/portfolio deliverable
- TOCTTOU

BCQs

- 15 minutes
- The College likes to see everyone submit them.
- If $(n-1)/n$ of you submit them, I will give everyone an extra two points on the final exam.

chroot()

- Recall: root directory is top level directory
- You can change your root directory!
- System call: `chroot(dir_path)`
- Requires root (as in superuser) privileges

chroot()

- “Jails” processes into a directory
- Useful for an HTTP or FTP server should only server files from files from a certain directory
- What attack that we studied earlier does this help prevent?

chroot()

- Recipe:

1) First, change root:

- `chroot("/var/www")`

2) Second, move into new chroot "jail":

- `chdir("/var/www")`

- Your new root is `/var/www`

Escaping jail

- Can escape a chroot “jail” if root:
 - mkdir(“foo”)
 - chroot(“foo”) // create jail inside your jail
 - chdir(“../../../../”) // allows you to “cd ..” out of jail
 - chroot(“.”) // now “/” points to real root
- Why is chroot(“.”) desirable?
- So does chroot() *really* prevent directory traversal attacks?

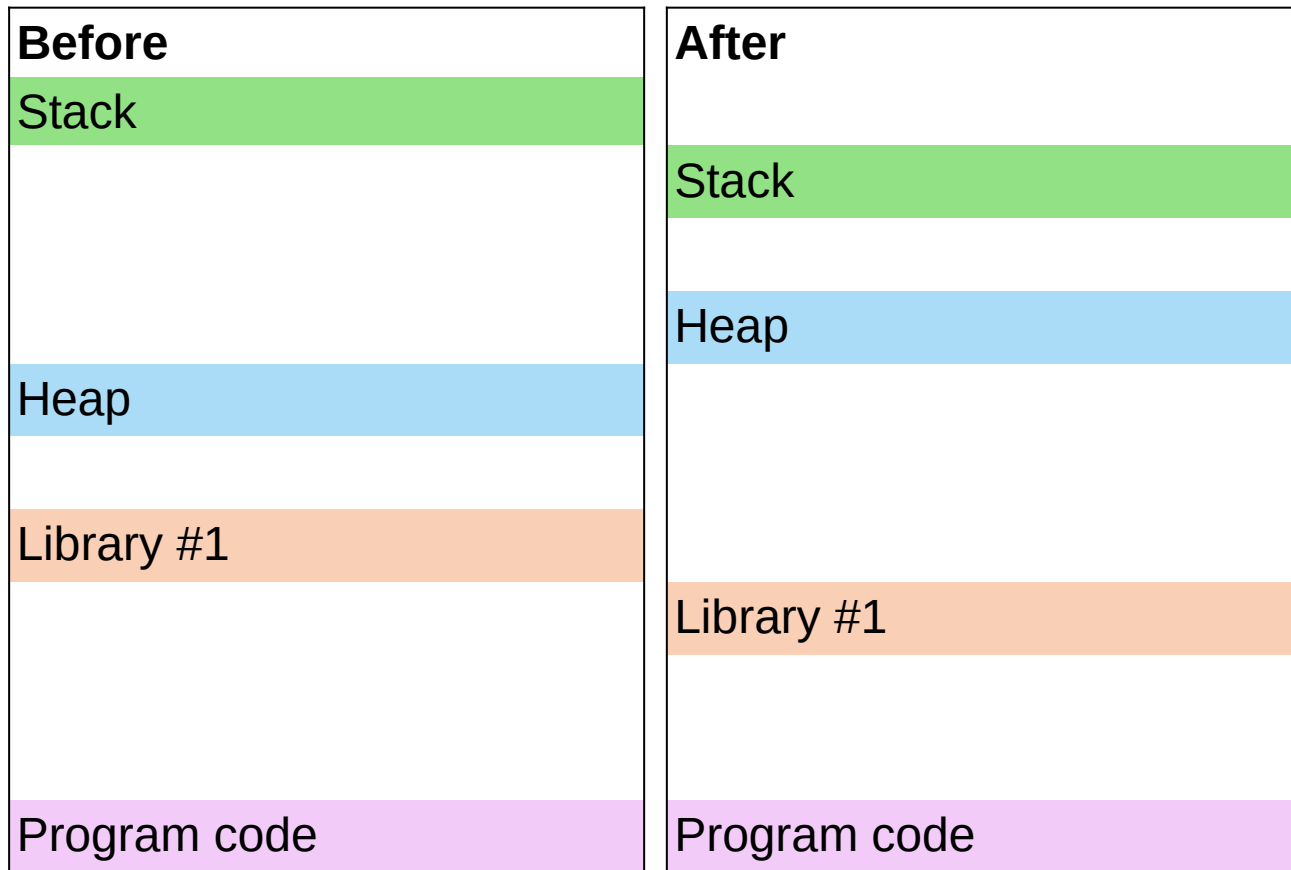
chroot() threat model

- Remember that calling chroot() requires
 - root permissions
 - the ability to execute arbitrary code
- chroot() is still effective when these steep requirements cannot met
- But! They *can* be met (such as in your format string vulnerability assignment)

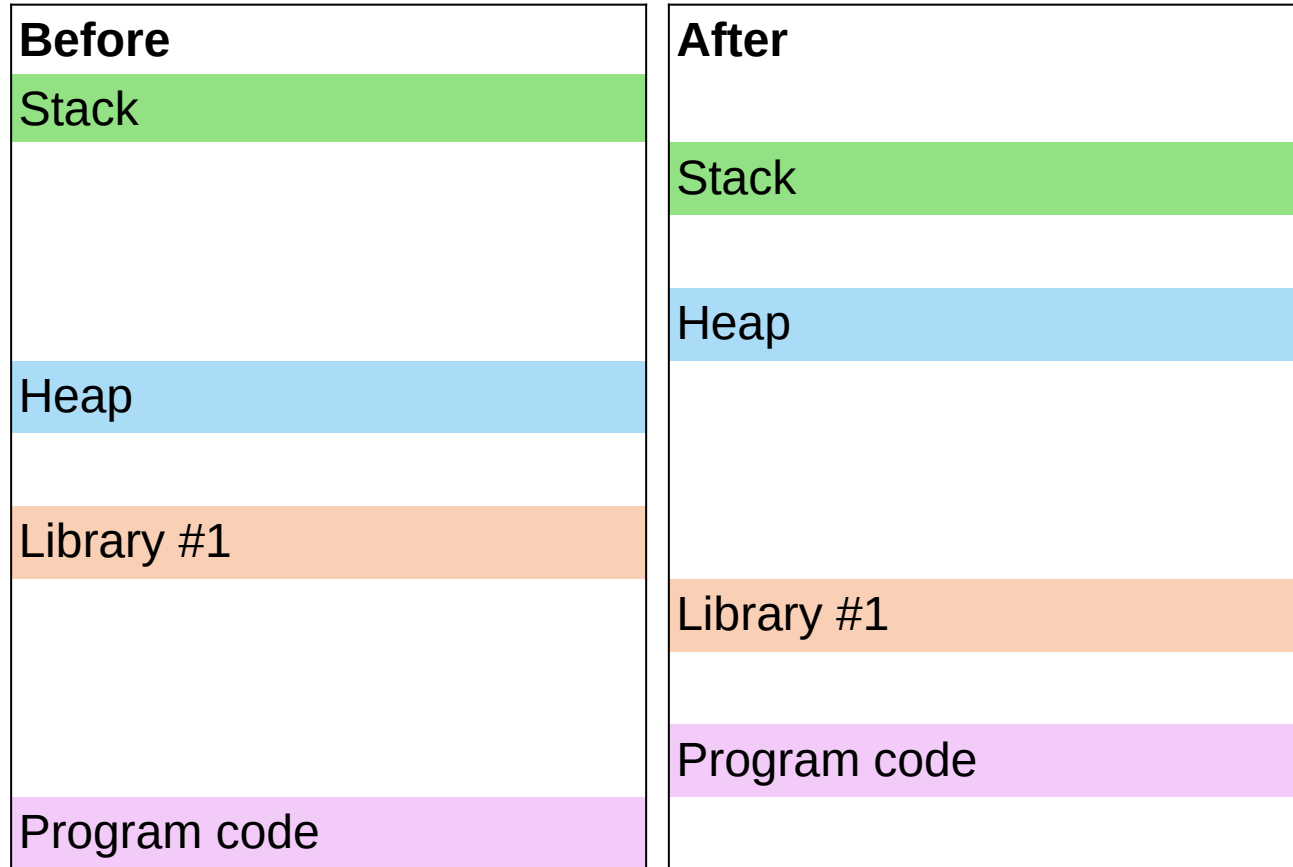
chroot() in python

- Relevant functions:
- `os.chroot()`
- `os.getcwd()`
- `os.mkdir()`
- `os.chdir()`
- `os.listdir()`

ASLR (without PIE)



ASLR (with PIE)



“Program code”?

- .text (machine code)
- .data/.bss (global variables)
- .rodata (constants such as string literals)
- GOT
- PLT

Finding a pointer to program code

- To global variable
- To string literal
- To function in program code
- Et cetera.

Portfolio Project

- Deadline: December 12, 2025, 10:00pm
- <https://tildesites.bowdoin.edu/~j.knockel/csci3330f25/portfolio.pdf>

TOCTTOU vulnerabilities

- Time-of-check to time-of-use
- Attacker exploits a *race condition*
- Changes something quickly after the check

TOCTTOU vulnerabilities

- Setuid root binaries: which UID gets root? (real vs. effective)
- access() system call: checks if *real* UID has access, **not** effective

Classic example

```
if (!access("/tmp/something", W_OK)) {  
  
    fd = open("/tmp/something", O_WRITE);  
    /* write to file */  
} else /* print error */
```

Classic example

```
if (!access("/tmp/something", W_OK)) {  
    // In another process:  
    // rm -f /tmp/something  
    // ln -s /etc/cron.d/xxx /tmp/something  
    fd = open("/tmp/something", O_WRITE);  
    /* write to file */;  
} else /* print error */
```

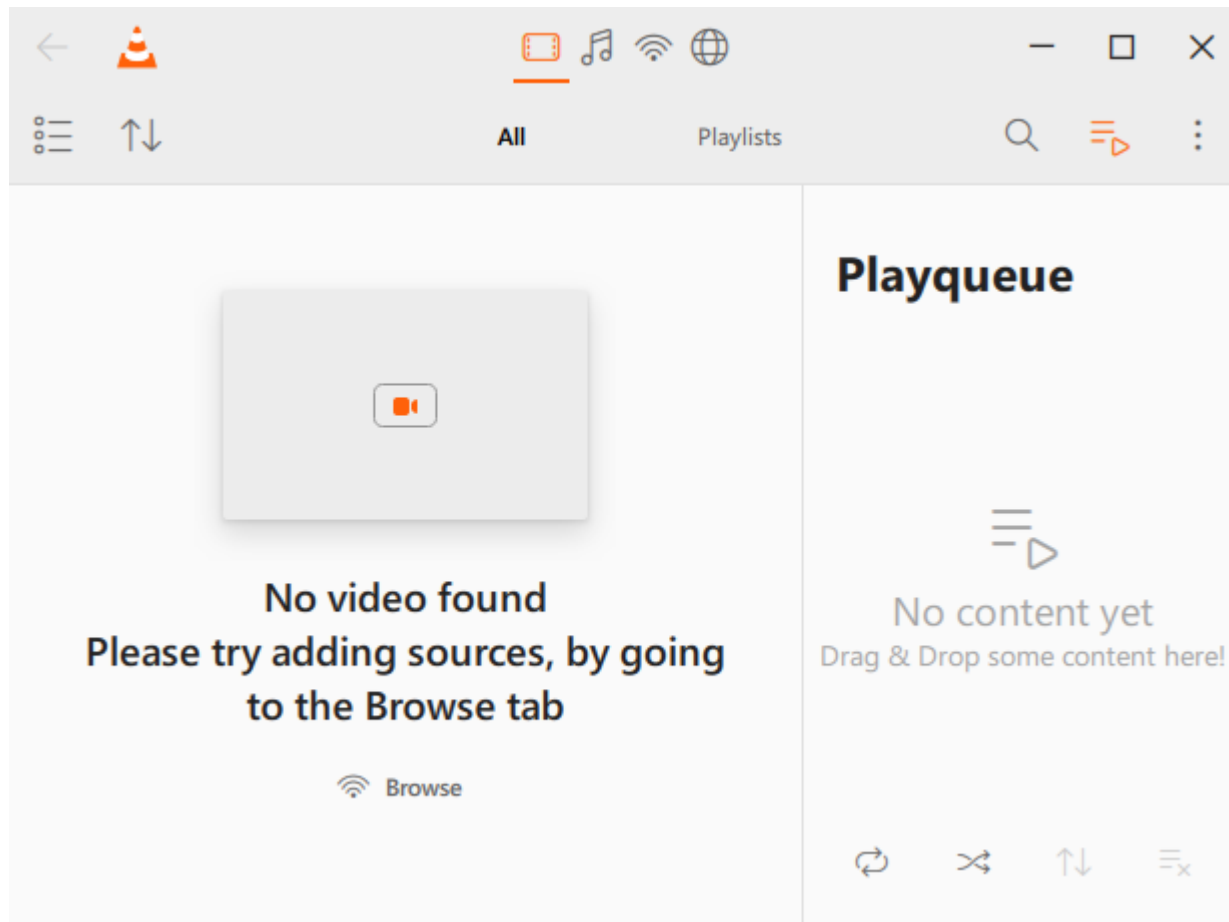
ln -s ?

- Shell command that creates a symbolic link
- Points to another file

You have to be very quick...

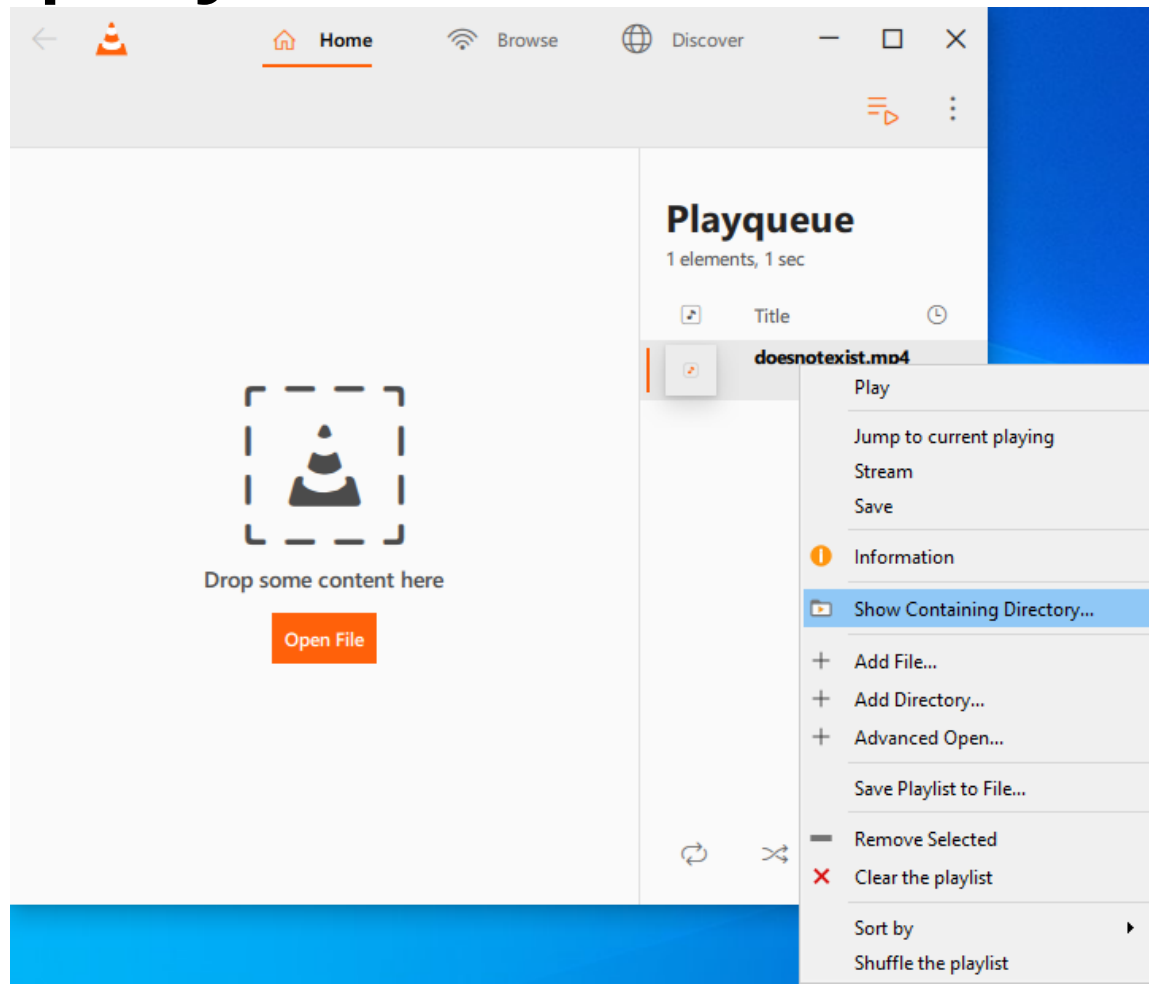
- You can just keep calling a setuid binary until you get lucky
- But sometimes you can make your own luck...

VLC



VLC has playlist files

- List of paths
- “Show containing directory”



```
@@ -322,8 +322,10 @@ void StandardPLPanel::popupAction( QAction *action )
    temp = model->getURI( index );
    if( ! temp.isEmpty() ) path = vlc_uri2path( temp.toLatin1().constData() );
    if( path == NULL ) return;

    QDesktopServices::openUrl(
        QUrl::fromLocalFile( QFileInfo( qfu( path ) ).absolutePath() ) );
    free( path );
    break;
```

- `QDesktopServices::openUrl()` API
- `c:\windows\song.mp3`
- `QDesktopServices::openUrl("c:\\windows\\")`
- If a directory, opens the directory in the file browser

Media

File Home Share View

← → ↕ ⬆ ⬇ This PC > Local Disk (C:) > Windows > Media > Search Media 🔍

en-US

> Fonts

GameBarPresenceWriter

> Globalization

> Help

> IdentityCRL

> IME

> ImmersiveControlPanel

InboxApps

> INF

> InputMethod

> Installer

L2Schemas

LanguageOverlayCache

LiveKernelReports

> Logs

> **Media**

> Microsoft.NET

> Migration

ModemLogs

Name	#	Title	Contributing artists	Album
windows Logon.wav				
Windows Menu Command.wav				
Windows Message Nudge.wav				
Windows Minimize.wav				
Windows Navigation Start.wav				
Windows Notify Calendar.wav				
Windows Notify Email.wav				
Windows Notify Messaging.wav				
Windows Notify System Generic.wav				
Windows Notify.wav				
Windows Pop-up Blocked.wav				
Windows Print complete.wav				
Windows Proximity Connection.wav				
Windows Proximity Notification.wav				
Windows Recycle.wav				
Windows Restore.wav				
Windows Ringin.wav				
Windows Ringout.wav				
Windows Shutdown.wav				
Windows Startup.wav				
Windows Unlock.wav				
Windows User Account Control.wav				

98 items

What if it isn't a directory?

- Such as a program?
- It will execute the program
- How can c:\windows\ be a program?
- \\1.2.3.4\webdav\directory\
- Server can respond with file for directory

```
... @@ -322,8 +322,10 @@ void StandardPLPanel::popupAction(  
    QAction *action )
```

```
322         temp = model->getURI( index );  
323         if( ! temp.isEmpty() ) path = vlc_uri2path(  
            temp.toLatin1().constData() );  
324         if( path == NULL ) return;
```

```
325         QDesktopServices::openUrl(  
326 -           QUrl::fromLocalFile( QFileInfo( qfu(  
            path ) ).absolutePath() ) );
```

```
327         free( path );
```

```
328         break;
```

```
329
```

```
...
```

```
...
```

```
... @@ -322,8 +322,10 @@ void StandardPLPanel::popupAction(  
    QAction *action )
```

```
322         temp = model->getURI( index );  
323         if( ! temp.isEmpty() ) path = vlc_uri2path(  
            temp.toLatin1().constData() );  
324         if( path == NULL ) return;
```

```
325 +         temp = QFileInfo( qfu( path ) ).absolutePath();
```

```
326 +         if( !QFileInfo( temp ).isDir() ) return;
```

```
327         QDesktopServices::openUrl(  
328 +           QUrl::fromLocalFile( temp ) );
```

```
329         free( path );
```

```
330         break;
```

```
331
```

```
...
```

```
...
```

```
@@ -322,8 +322,10 @@ void StandardPLPanel::popupAction( QAction *action )
    temp = model->getURI( index );
    if( ! temp.isEmpty() ) path = vlc_uri2path( temp.toLatin1().constData() );
    if( path == NULL ) return;
+   temp = QFileInfo( qfu( path ) ).absolutePath();
+   if( !QFileInfo( temp ).isDir() ) return;
    QDesktopServices::openUrl(
+       QUrl::fromLocalFile( temp ) );
    free( path );
    break;
```

TOCTTOU

- `isDir()` generates a request to the server
- Server can respond “true”
- Then `openUrl()` download the file
- The server can respond with a file

TOCTTOU

- If you ask someone to play a playlist with...
- `\\maliciouserver\directory\song.mp3`
- And they open the directory
- It will execute an arbitrary program

TOCTTOU

- Only one problem... Windows caches the server responses for 60 seconds
- Simple fix: add a 60 second delay in some intermediate network traffic so that the cache entry times out...
- Arbitrary code execution!

Portfolio Project

- Deadline: December 12, 2025, 10:00pm
- <https://tildesites.bowdoin.edu/~j.knockel/csci3330f25/portfolio.pdf>