Objectives

- Final Exam Questions?
- Security Considerations
- Course Evaluations

Security Considerations

- We've been trying to make our programs very flexible, general
  - E.g., User specifies input (file names)
- Must validate user input
  - We have done some of this
  - Need to improve, restrict user input
    - Microsoft security/reliability architect recommends that you restrict the sources of input (gets rid of trust issues)

Simple Example of Security Bug

- Example from a real web site
  - Simulated here
- More fundamental than a security bug
  - Hopefully, CS111 students would know better
    - May not always check user input but know paid programmers should do it!
- How to address this security issue?

Security Bug in Python

- Demonstrate with security_bug.py

Why the Security Bug?

- `input(prompt)` is actually `eval(raw_input(prompt))`
- Example:
  - `val = input("Enter a value: ")`
  - In terminal:
    - Enter a value: 1 + 5 + 6
  - `val` is assigned 12
- Example:
  - `totalGrades += input("Enter grade: ")`
  - Enter grade: totalGrades * 3
  - `totalGrades` is assigned totalGrades+totalGrades*3

Handling the Security Bug

- Cast `raw_input` to an integer

```
user_input = raw_input("Enter a grade")
totalGrades += int(user_input)
```

  - Casting to an int will throw an exception if we try to input totalGrades
  - Use try/except to handle error
- In the future, Python is replacing `input` function with `raw_input`
  - Forced to take this approach

More on `input` function

- Explains the error message we got if we tried to input B6
  - Looks for variable named B6

```
Enter your age: B6
Traceback (most recent call last):
 File "currentAge.py", line 22, in <module>
  main()
 File "currentAge.py", line 9, in main
  age=input("Enter your age: ")
 File "<string>", line 1, in <module>
NameError: name 'B6' is not defined
```

File Input from User

- Alternative 1:
  - Prompt user: What file do you want the program to process?
  - Prompt user: What file do you want the program to write/output?

File Input from User

- Alternative 1:
  - Prompt user: What file do you want the program to process?
  - Prompt user: What file do you want the program to write/output?
- **Issues**:
  - What if bad input file? (not just bad name)
  - What if output file writes over existing file?
    - May or may not be malicious; could just be dippy

File Input from User

- **Issues**:
  - What if bad input file? (not just bad name)
  - What if output file writes over existing file?
    - May or may not be malicious; could just be dippy
- How to address issues?
  - Check if (input/output) file exists first
    - Use the **os** module
  - Permission problems should be handled by OS

File Input from User

- Alternative 2:
  - Prompt user: Which of x files do you want the program to process?
    - The x files are known to be okay
  - Automatically create (valid) file name
    - Again, check that file does not already exist

Final Exam Take Home Questions

- 2 Essay questions about the Broader Issues
- Bring essay answers with you to exam
- Each essay should be about 1/2 a page
- Goal: answer the question clearly, precisely, and convincingly
  - Not too wordy
  - Evidence/examples to support your argument
  - Correct spelling, grammar, punctuation
- Second essay: can only answer partially until Friday's lecture

Reminders

- Broader Issues: Diversity in Computer Science
- Hand back BI grade sheets
- Final exam envelopes!
- Finish Lab 11
  - ➤ ConnectFour issue
- Study for Exam
- Do ConnectFour extra credit (additional 2D practice)
- Course Evaluations

Course Evaluations