

Objectives

- Security Considerations
- Final Exam questions
- Course Evaluations - online

Apr 7, 2010

Sprenkle - CS111

1

Review

- How do you create a 2D list?
- How do you get the 2nd element in the 3rd "row" of a list?
- How do you find the number of lists in a 2D list?
- How do you find the number of elements in one of those lists?

Apr 7, 2010

Sprenkle - CS111

2

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)

Apr 7, 2010

Sprenkle - CS111

3

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?

Apr 7, 2010

Sprenkle - CS111

4

Another Example

- Modifying URL parameters

```
http://network.domain.com/login.jsp?room=4531&rate=9.99
```

```
http://network.domain.com/login.jsp?room=4531&rate=0.00
```

Learn more in CSCI335!

Apr 7, 2010

Sprenkle - CS111

5

Security Bug in Python

- Demonstrate with security_bug.py

Apr 7, 2010

Sprenkle - CS111

6

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:
 - `total += input("Enter deposit: ")`
 - Enter deposit: `total * 3`
 - `total` is assigned `total+total*3`

Apr 7, 2010

Sprenkle - CS111

7

Handling the Security Bug

- Cast `raw_input` to an integer
- ```
user_input = raw_input("Enter a deposit")
total += int(user_input)
```
- Casting to an `int` will throw an exception if we try to input `total`
  - Use `try/except` to handle error
  - In the future, Python is replacing `input` function with `raw_input`
    - Forced to take this approach

Apr 7, 2010

Sprenkle - CS111

`fixed_input.py`

8

## 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
```

Apr 7, 2010

Sprenkle - CS111

9

## 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?

Apr 7, 2010

Sprenkle - CS111

10

## 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

Apr 7, 2010

Sprenkle - CS111

11

## 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`

Apr 7, 2010

Sprenkle - CS111

12

## 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

Apr 7, 2010

Sprenkle - CS111

file\_input.py

13

## Handling Security

- Prevention
  - Don't allow bad stuff
  - Examples: validate input, restrict input files
- Detection
  - After see bad, stop it
  - Example: Microsoft's Halo

Apr 7, 2010

Sprenkle - CS111

14

## Final Exam Take Home Questions

- 2 essay questions about the Broader Issues
- Due before end of exam period
- 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

Apr 7, 2010

Sprenkle - CS111

15

## Final Exam Review

- What do you need to do to be able to use methods from a class?
- What are the different ways to iterate through a list?
- How can you iterate through a dictionary?

Apr 7, 2010

Sprenkle - CS111

16

## Reminders

- Broader Issue: One Laptop Per Child
- Final exam envelopes!
- Finish Lab 11
- Study for Exam
- Do ConnectFour extra credit (additional 2D practice)
- Course Evaluations
  - Sakai – tests & quizzes

Apr 7, 2010

Sprenkle - CS111

17