

Mason CC Advanced Track

Python and Ruby in Security



Recent CTFs



- National Cyber League
 - Hoa 82nd overall (79th in Gold/top tier), Zaine 212st (187), Roberto 201st (178th), Nihaal 211st (186), out of 4452 pending audit.
 - Zaine got 53rd overall in Cryptography.
 - Everyone registered participated
 - This was not the case last year, sucks b/c \$15
 - Team game to come soon
 - We will have two teams, one likely inter-school (max of 7 per, 12 are registered)

Recent Exec Work and Stuff



- New Constitution Approved
- Bench Painting Submitted
- Executive Board Meeting

What are we doing?



- Python and Ruby CTF Scripting, Regex
- Regex, Python, and Ruby
- Plenty of real world applicability
 - Python, Ruby, Regular Expressions, etc
 - If you've never used libraries or handled objects, this is a good introduction

Similarities



- Both are commonly interpreted
 - .pyc is a common counterexample
- Both are easy to start - procedural is possible/common
 - Makes both slightly easier to write bad code
- Both generally expect an external application
 - Kinda goes with the interpreter bit

Python Differences



- Very tab/intent oriented/dependent
- 2.7 v 3 clash
 - Dependencies issues
 - Syntax (i.e. print, raw_input, etc)
- Better community
- Pretty fast

```
>>> f'Hello, {name}!'
'Hello, Bob!'
```

```
name = raw_input("Name please: ")
print "Your name is", name
```

```
import requests

print requests.get('https://competitivecyber.club').text
```

Ruby Differences



- Relatively unique syntax
- Uses terms like **if .. end**, not indentation dependent
- **Bad** memory collection
- Relatively clean for higher level tasks like requests
- Better string interpolation in my opinion
 - 3.6 got better
- Gem-crazy

```
puts "Please enter your name"
name = gets
puts "Your name is #{name}"
```

```
require 'httparty'

puts HTTParty.get('https://competitivecyber.club').parsed_response
```

CTF Context



- Automation, some CTFs will require it of you
- **pwntools in Python**
- Long list of gems (dependencies) in Ruby
 - This is standard in Ruby, people are freaks for gems
 - For web I'm a big fan of HTTParty and requests
 - pwntools in Ruby is forked but trash

Non-CTF Example

- Script to determine viable students
- From VSE Ice Cream Social last year
- Not the world's best, how could it be made better?

```
import requests

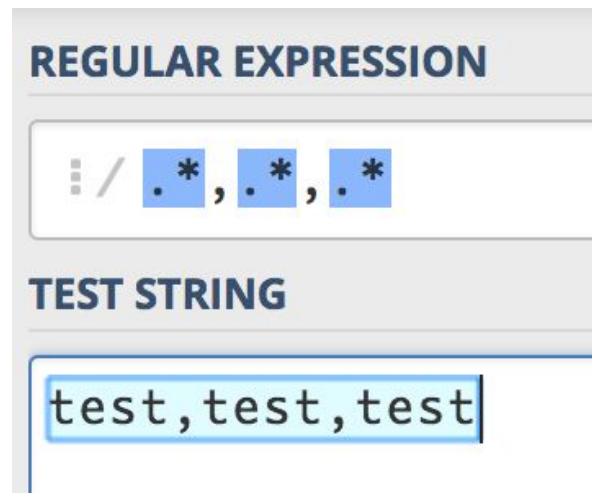
with open('list') as f:
    for user in f:
        r = requests.get('https://api.srct.gmu.edu/peoplefinder/v1/basic/all/'+user)
        res = r.json()
        if len(res["results"]) == 0:
            print user
```

Quick on Regular Expressions



- <https://go.gmu.edu/regexgolf>
- regex101.com
- Basically a set of rules for pattern matching strings
- Really useful in CTFs and in general
- People like rolling security rules in regex
- Example:
- Has named groups, partial matches, etc

`/*.*,*,*/g`



REGULAR EXPRESSION

:/.*,*,*/g

TEST STRING

test,test,test

Tips



- Lint/SASL your code
- `` in Ruby will run something, returns output
- os.system in Python will run something, returns exit code
 - os.popen to run with output

On Linting



- rubocop in Ruby, pylint in Python
 - Different tools exist for different languages
 - As well as standards, though you can override
 - Protip: Ruby can fix with **-a**

```
db/seeds.rb:9:76: C: Prefer single-quoted strings when you don't need string interpolation or special symbols.
user = User.find_or_create_by email: 'admin@bicdesignonfire.com.au', name: "Admin"
                        ^^^^^^^

db/seeds.rb:9:81: C: Line is too long. [82/80]
user = User.find_or_create_by email: 'admin@bicdesignonfire.com.au', name: "Admin"
                        ^^

test/test_helper.rb:5:7: C: Use nested module/class definitions instead of compact style.
class ActiveSupport::TestCase
  ^^^^^^^^^^^^^^^^^^^^^^^^^^

test/test_helper.rb:6:81: C: Line is too long. [82/80]
# Setup all fixtures in test/fixtures/*.yml for all tests in alphabetical order.
                        ^^

70 files inspected, 699 offenses detected
→ the-bic-lighters git:(develop) |
```

```
[root@9607fb561ba6:/usr/src/app# pylint test.py
No config file found, using default configuration
***** Module test
C:  4, 0: Unnecessary parens after 'print' keyword (superfluous-parens)

-----
Your code has been rated at 0.00/10 (previous run: -10.00/10, +10.00)
```

Challenge



TODO: PUT MY IP ADDRESS HERE

Cross-Organization Plug



- Mason SRCT writes loads of Python
 - go.gmu.edu is Django/Python
 - whatsopen.gmu.edu is Django/Python for API
 - schedules.gmu.edu is Rails/Ruby
 - You can contribute to/read all of this
- Join them at srct.slack.com / srct.gmu.edu
- Read their code at