Python for Web Pen-testers

(Snake bites)

whoami

Name : **Anant Shrivastava**

work at **7Safe** as a **Information Security Consultant**.

Primary focus area's include **Web** and **mobile**.

Certifications : **RHCE**, **CEH**, **SANS-GWAPT**

Speaker : **Nullcon**, **ClubHack**, **c0c0n**

Creator of Android Tamer (a Virtual Machine for Android Security)

Active member of **null** and **Garage4Hackers** open communities

Website : http://anantshri.info

Contact : anant@anantshri.info **Twitter** : @anantshri

Objectives

- Presentation is only focused on python. (No comparisons with other languages please)
- Idea is to introduce a regular web penetration tester with various functionalities available in Python and give them a head start
- There is no new tools release

Why Python

- Easy and Flexible Language
- Massive module base allowing lots of use case scenario's to be written in small timeframe and codebase efforts.
- Inbuilt Language support in number of tools
  - Burp
  - IRONWASP
- API support provided by OWASP-ZAP
- Custom application logics or requirements
- Or as simple as

JUST FOR THE FACT THAT YOU CAN DO IT

Example of Existing Python Tools

w3af

SpikeProxy

sqlmap

ProxyStrike

wapiti

sulley

Peach

Canvas

Pyscan

DeBlaze

Scapy

MonkeyFist

Pcapy

MyNav Idapython

Python Variations

- Cython
- Jython
- IronPython

Remember to consider these two as different

- Python 2.X
- Python 3.X

Noteworthy Python modules

Python Modules

- Python has a set of inbuild packages listed as

  http://docs.python.org/2/library/index.html (2.7.3)
- Besides these packages community has provided a large set of modules and this is what

Standard Modules

20. Internet Protocols and Support
    - 20.1. `webbrowser` — Convenient Web-browser controller
    - 20.2. `cgi` — Common Gateway Interface support
    - 20.3. `cgitb` — Traceback manager for CGI scripts
    - 20.4. `wsgiref` — WSGI Utilities and Reference Implementation
    - 20.5. `urllib` — Open arbitrary resources by URL
    - 20.6. `urllib2` — extensible library for opening URLs
    - 20.7. `httplib` — HTTP protocol client
    - 20.8. `ftplib` — FTP protocol client
    - 20.9. `poplib` — POP3 protocol client
    - 20.10. `imaplib` — IMAP4 protocol client
    - 20.11. `nntplib` — NNTP protocol client
    - 20.12. `smtplib` — SMTP protocol client
    - 20.13. `smtpd` — SMTP Server
    - 20.14. `telnetlib` — Telnet client
    - 20.15. `uuid` — UUID objects according to RFC 4122
    - 20.16. `urlparse` — Parse URLs into components
    - 20.17. `SocketServer` — A framework for network servers
    - 20.18. `BaseHTTPServer` — Basic HTTP server
    - 20.19. `SimpleHTTPServer` — Simple HTTP request handler
    - 20.20. `CGIHTTPServer` — CGI-capable HTTP request handler
    - 20.21. `cookielib` — Cookie handling for HTTP clients
    - 20.22. `Cookie` — HTTP state management
    - 20.23. `xmlrpclib` — XML-RPC client access
    - 20.24. `SimpleXMLRPCServer` — Basic XML-RPC server
    - 20.25. `DocXMLRPCServer` — Self-documenting XML-RPC server

External Modules

Requests

- Module alternative for urllib2 (HTTP handler)
- Internally handles the http or https transition.
- Simpler interface exposed
- Sample PoC

```
import requests
r = requests.get (get being one of the http method)
r.text/r.content = content
r.status_code
r.headers
```

Read more here :
http://docs.python-requests.org/en/latest/user/quickstart/

BeautifulSoup

- Text / HTML parser

  Sample (I am using standard parser works fine so far but people recommend using a different parser)

  - import request
  - from BeautifulSoup import BeautifulSoup
  - r=requests.get('http://www.anantshri.info')
  - soup=BeautifulSoup(r.text)
  - //searches for each div with class attachment
  - each_div=soup.find('div',{'class':'attachment'})

- Demo : Wordpress attachment enumeration

Argparse

Create a quick and simple program with various options and switches

Import argparse

desc="""Description goes here"""

parser = argparse.ArgumentParser(description=desc)

parser.add_argument("--url",help="URL",dest='target',required=True)

parser.add_argument("--debug",help="debug",action="store_true")

x=parser.parse_args()

x.target == url

x.debug == debug

Demo

- Scenario's
  - Regular expression pattern matching
  - XSS Fuzzing / parameter fuzzing
  - SVN files extraction (.svn publically exposed)

Regular Expression Pattern Match

```
#!/usr/bin/python
import requests
import re
regex_array = {
    'wordpress' : '<link rel=\"stylesheet\" [^>]+wp-content',
    'Drupal' : '<(?:link|style)[^>]+sites/(?:default|all)/(?:themes|modules)/'}
r=requests.get('http://www.anantshri.com')
for k, v in regex_array.iteritems():
    m=re.search(v,r.content)
if (m):
    print k, " : Detected "   , "  as the content has : ", m.group(0)
```

Simple XSS Fuzzer

```
#!/usr/bin/python
import requests
import sys
import argparse
import os
import urllib
url="http://fuzz/noxss.php?a="
f=open('xss.txt','r')
for row in f.readlines():
  r = requests.get(url + urllib.quote(row))
  if (row in r.content):
    print row
```

SVN extractor

https://github.com/anantshri/svn-extractor

simple script to extract all web resources by means of .SVN folder exposed over network.

http://blog.anantshri.info/svn-extractor-for-web-pentesters

More Stuff

- http://learnpythonthehardway.org/book/
- http://www.diveintopython.net/toc/index.html
- Go for SPSE or any on-line certification Course
- Read Violent Python
- Read Grey-hat Python
- Use Python on a daily basis for all operations.
- Use inbuilt support / features in
  - IRONWASP (In-line scripting)
  - BURP (Extension)

References for slides

- Large number of google searches and head scratching on various stackoverflow post's.
- Awesome documentations at docs.python.org
- Direct interaction with super helpful guys from null specially Akash Mahajan, Lava Kumar, Prasanna and Amol Naik for helping, criticizing forcing me to improve the skill set.

Thank You

If you wish to contact me after this presentation please use anant@anantshri.info For communication

PyInstaller

- Sometimes you want to shift your code as a one click executable without python dependency.
- PyInstaller packages every dependency into a single executable package

http://www.pyinstaller.org/

Python for Android

- Python 2.7 installable on android
- Can use android specific features with the help of SL4A (scripting language for android)

- Ssh or laptop are not the restrictions now.

- Sample scripts refer here : http://code.google.com/p/python-for-android/wiki/

Thank You