**记一次对steam盗号病毒的发现到拿下病毒的服务**

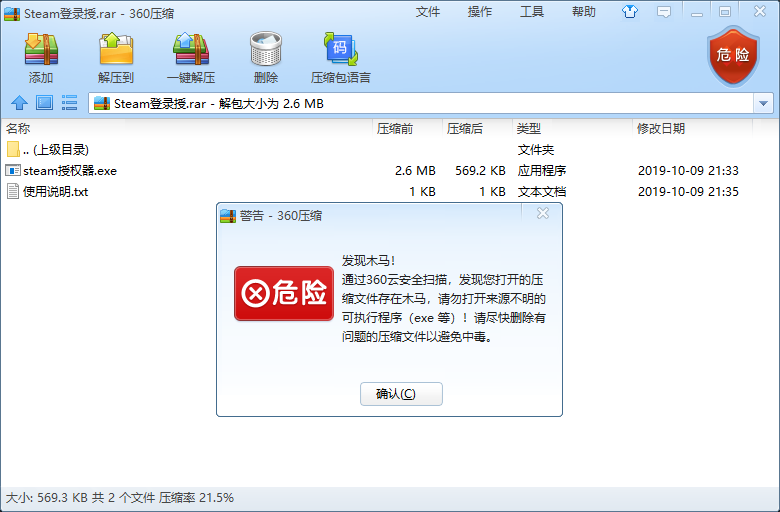
作者：夜无名

事情起缘

事情起缘与上周的一个无聊的周末，周末总是无聊而又乏味，好基友每周也准时约我吃鸡，可奈何我这个穷逼买！不！起！号！怎么办呢？对！卡盟，随便百度了一个某某卡盟，果然里面各种各样的游戏的账号都有，而且大部分的都是黑号，至于什么是黑号就不多说了（像这种倒卖黑号的也是一条产业链），就随便下了个订单。

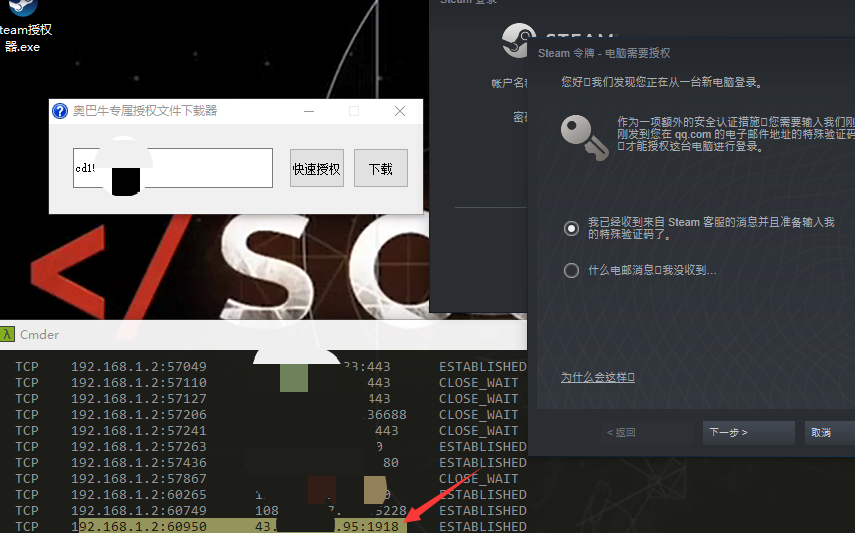
果不其然，并不是简简单单的直接发账号密码给你，而且还需要你下载所谓的登录器，有些安全意识的朋友都知道陌生可执行的文件千万不要随便去运行。

下载下来后，AV软件不出所料的报红了。



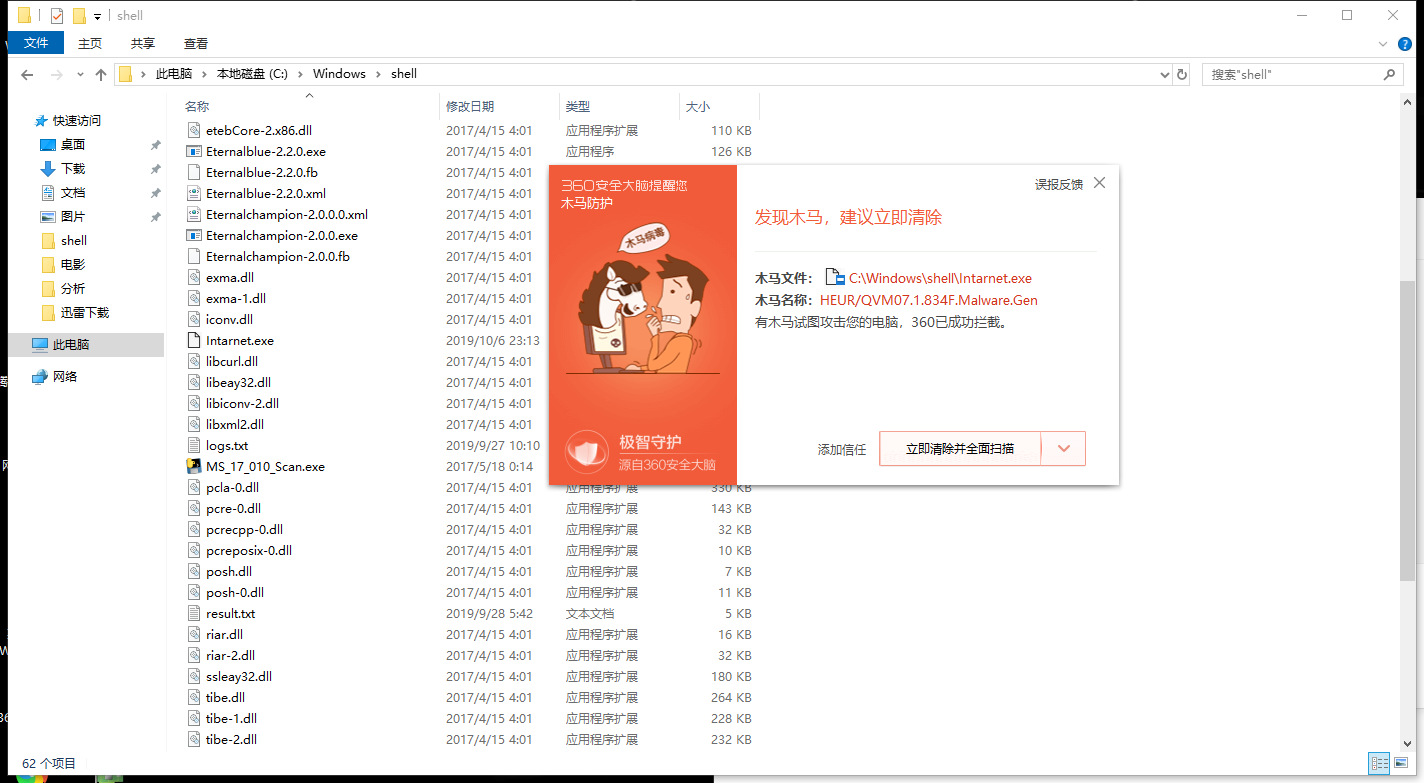
可是我头铁啊，不怂！直接解压添加白名单一气呵成运行软件，运行后我傻眼了，what？？软件居然在跟远程服务器做交互。





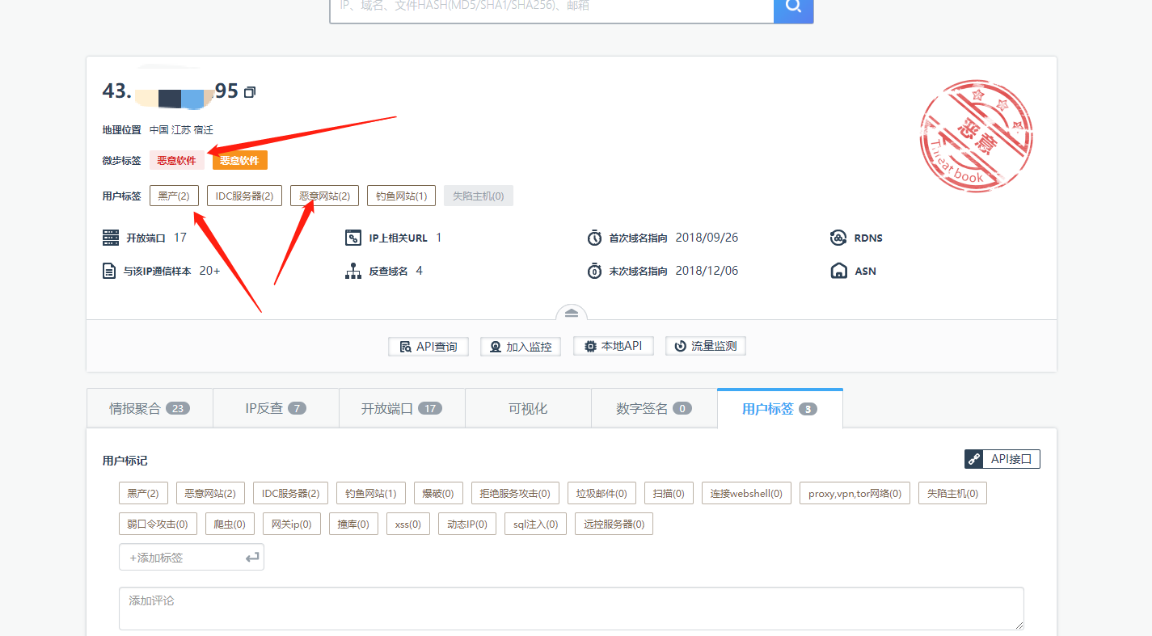
起初我以为这只是单纯的从远程服务器上下载账号所需的文件而已，大家都知道当steam在新的一台电脑上登陆的时候是要登陆验证的，其实这类的绕过验证的原理也很简单，steam登陆的时候在根目录产生了两个文件类似于cookie，估计是那些盗号者通过不为人知的手段盗取到号者的秘钥文件然后统一到远程服务器，在然后下载到本地替换文件从而到达了绕过验证的操作。

确实，这台服务器只是简单的储存账号的秘钥文件，可是！！可是！！登陆授权软件这个小碧池背着我又偷偷的在后台搞小动作，还好AV软件给我及时的拦了下来，省去了进一步对软件行为的分析。居然!在Windows的目录下生产个shell文件夹，里面还是有一套ms17-010工具，嗯？给你钱了还要搞我一波内网？这谁乐意啊。



信息探测

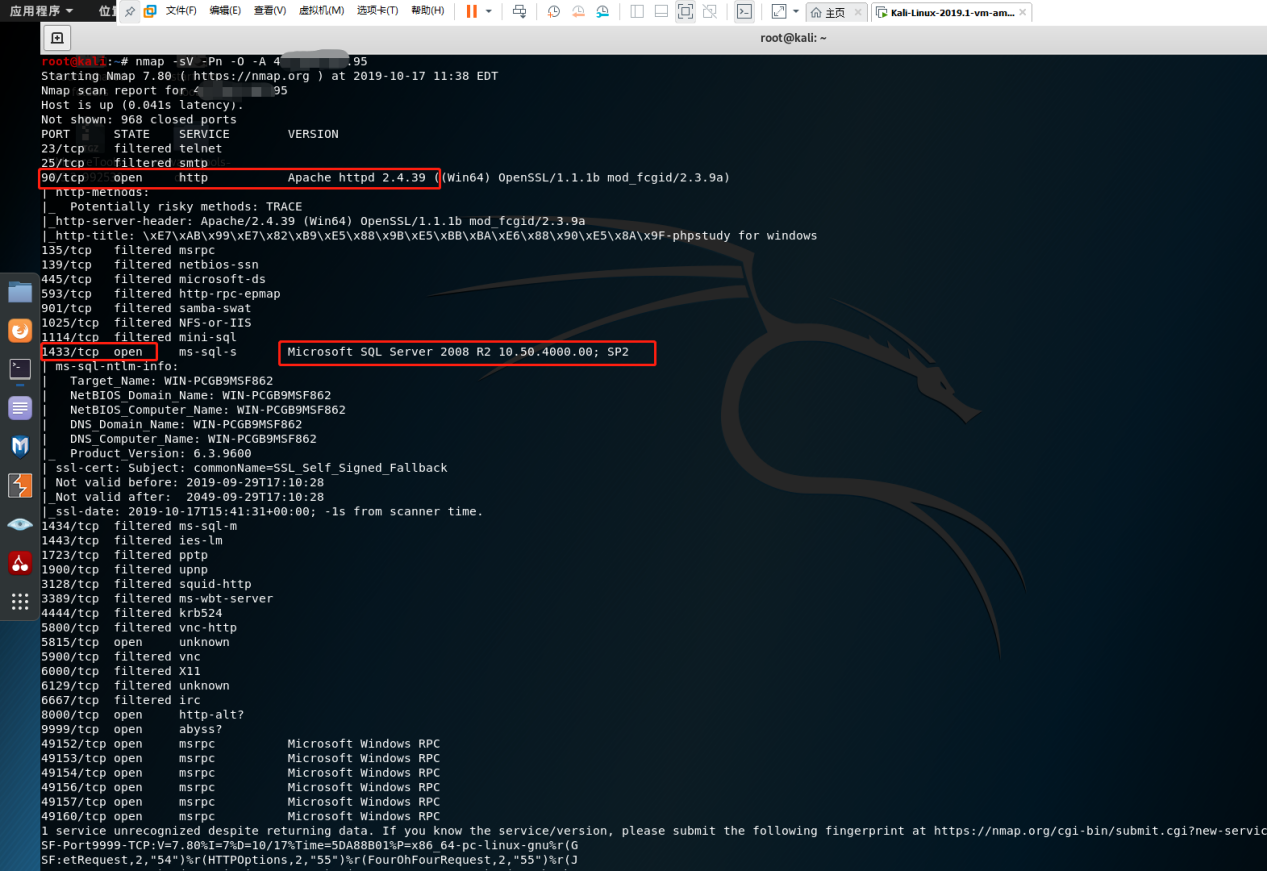
由于前面已经抓到了该软件跟服务器交互的ip，既然是跟盗号有关的我第一个念头想到的就是威胁情报，直接把该ip丢到了微步在线的威胁情报看看能不能找到什么线索，果不其然在五月份的时候就有用户标记出了此IDC服务器是用于steam盗号的，且爆出了疑似该服务器的所有者的一些联系方式。





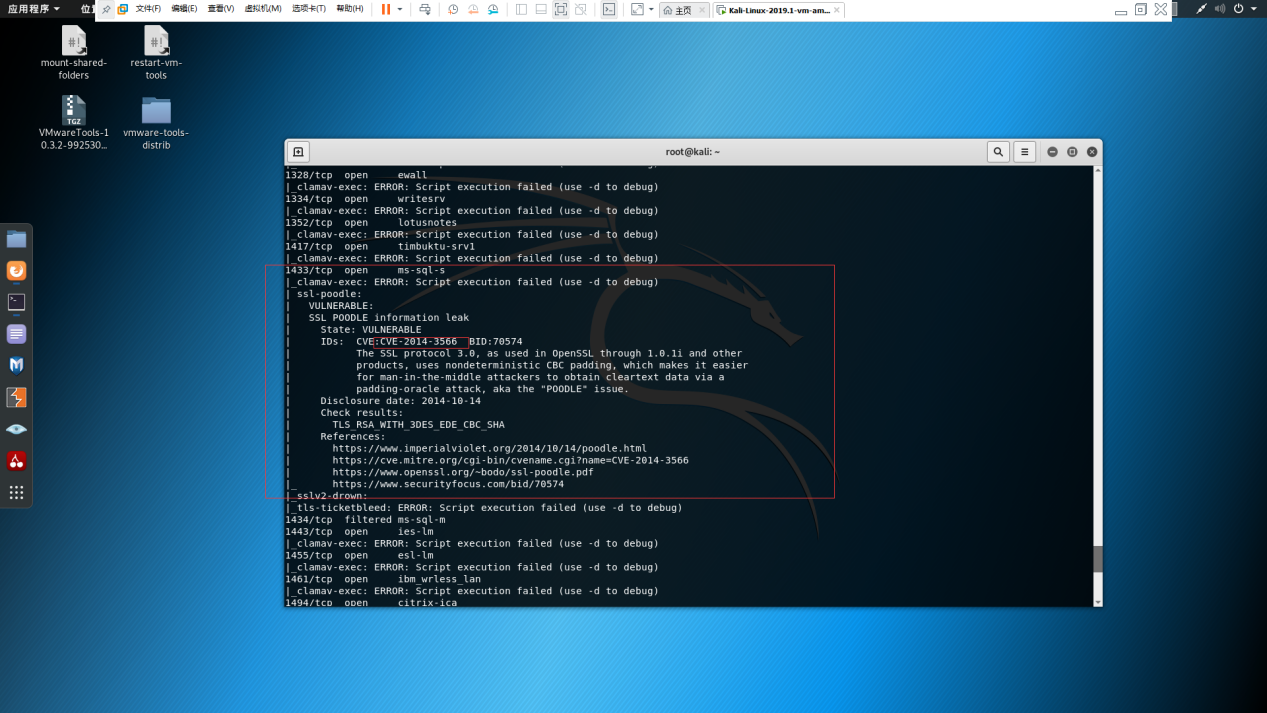
虽然得到了一些可能跟该服务器有关的一些信息，或许这些信息对后续需要爆破时制作密码表是有一定的帮助的，爆破属于后续且咱们现在还对该服务器一无所知，还不清楚开放了哪些端口提供哪些服务，废话不多说上nmap一顿扫描再说。

Nmap -sV -Pn -O -A 4x.xx.xx.95,得到以下信息，中间件使用的是Apache2.4.39版本，服务器是Windows server 2008 R2，且开放的web端口是90,1433的mssql数据库端口也是对外网开放的（这里我们可以结合上面得到的信息制作密码表对数据库进行爆破）。



上面只是对该服务器的一些端口信息初步的探测，接下来咱们在用nmap自带的漏洞脚本进一步对主机的漏洞进行探测，nmap --script=vuln 4x.xx.xx.95这里是利用nmap脚本对目标主机进行检查是否存在常见的漏洞，且如果存在漏洞nmap也会给出相应的cve编号，然后咱们在利用kali自带的metasploit渗透框架进行对应的cve编号搜索验证利用。

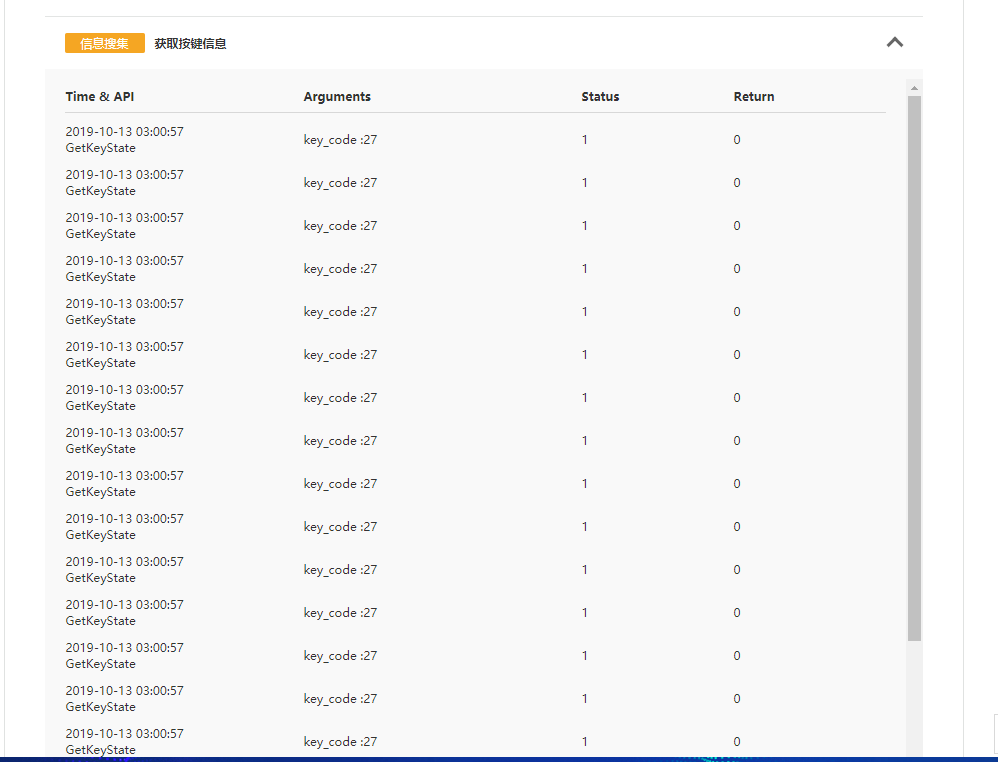
很遗憾，这里就探测出了一个有cve编号的漏洞cve-2014-3566,该漏洞是属于ssl3.0的信息泄露漏洞可进行中间人攻击，这里咱们就不深入研究了（实际利用的可能性不大）。



那么接下来就让我们看看该主机所开放的90端口都有什么web服务吧，如果是采用cms套件搭建的网站那对于我们后续进一步的渗透可能会有所帮助，可遗憾的是该主机所能访问的web端口并未建立起任何的服务。难道真的只能通过爆破1433数据库了？爆破成不成功先不说，爆破所花费的时间也是大量的，且直觉告诉我，这台服务器应该只是提供账号秘钥的存储与下载，就不花费太多的精力在上面了。

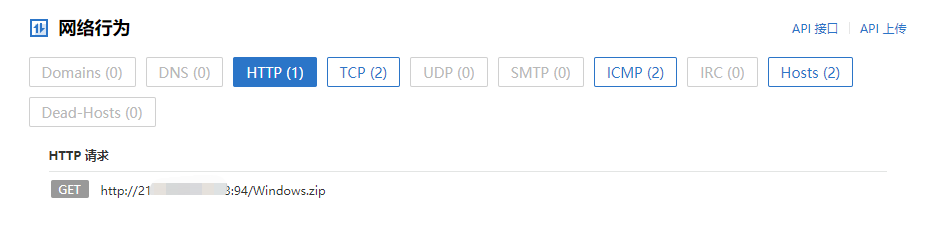
柳暗花明

正当我一筹莫展的时候，思路又回到了起始点，该授权登录器我还没好好的去分析呢，起初只是通过杀毒软件的拦截简单的进行判断而已，正好微步在线有个虚拟沙箱，扔进去跑一边看看。果然是盗号软件，赤裸裸的按键记录行为。



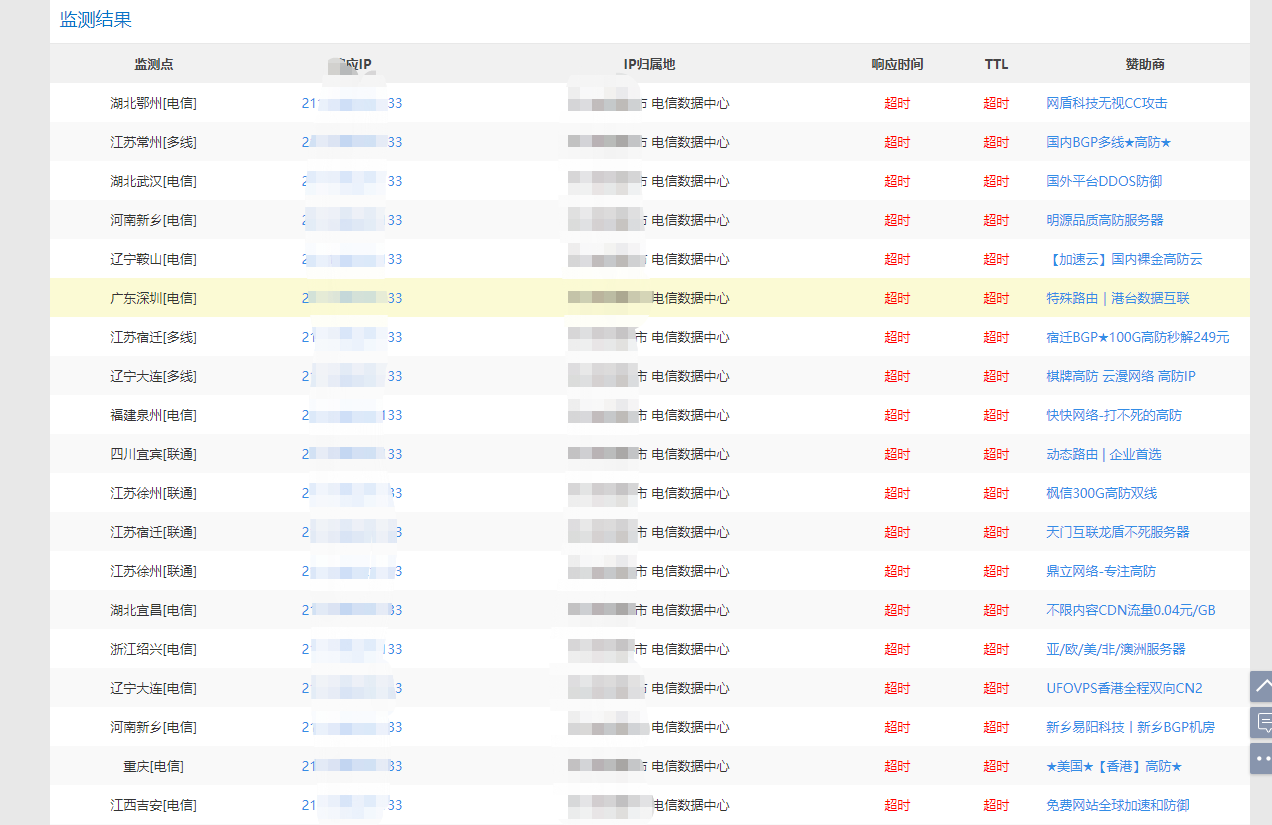
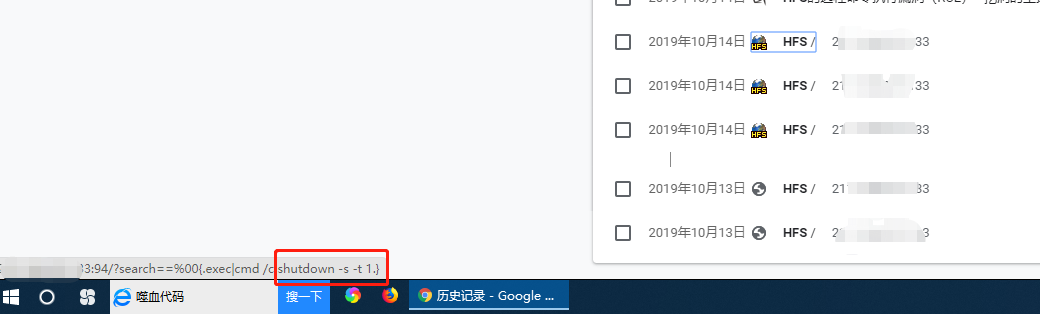
接下来在看看该盗号软件的执行流程，发现该软件释放出来的子程序还有跟另一台服务器有HTTP交互，且该服务器上的Windows.zip文件正是那套ms17-010的内网扫描工具。





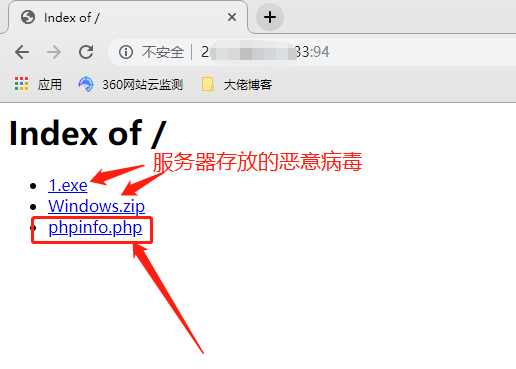
这里直接抓到了有交互的web端口，咱们访问之....看看病毒传播者是用什么容器来搭建的web服务，用的是HttpFileServer v2.3 b 271 随波汉化版，随而网上查阅了相关的资料，发现该版本存在高危远程代码执行漏洞随即进行了验证，由于验证时使用了shutdown -s -t 1导致了服务器关机，后面估计病毒传播者发现自己服务器被搞了就下线了该服务，更多详细的步骤也没能及时截图下来。（关于HFS的分析可参考安天实验室发布的文章https://www.antiy.com/response/hfs.html）。





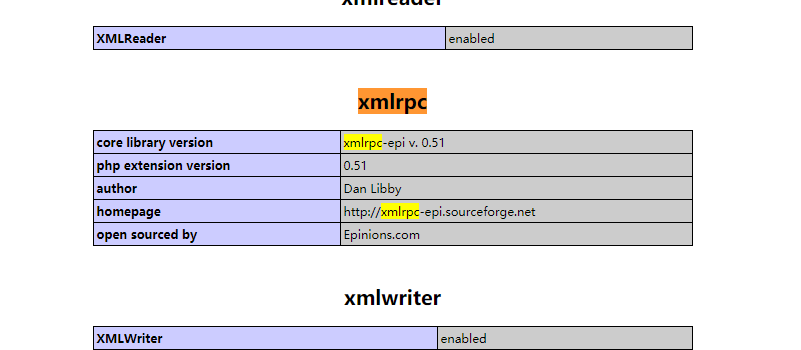
拿下权限

服务器关停了一天，不过这样也好至少市面上的软件无法从服务器上下载恶意软件进行传播了，我以为此事到这来就要结束了，可谁曾想网站又重新开起来了，只不过换了另一种搭建服务器。这刺眼的phpinfo.php想都不用想百分之百是用PHPstudy搭建的。



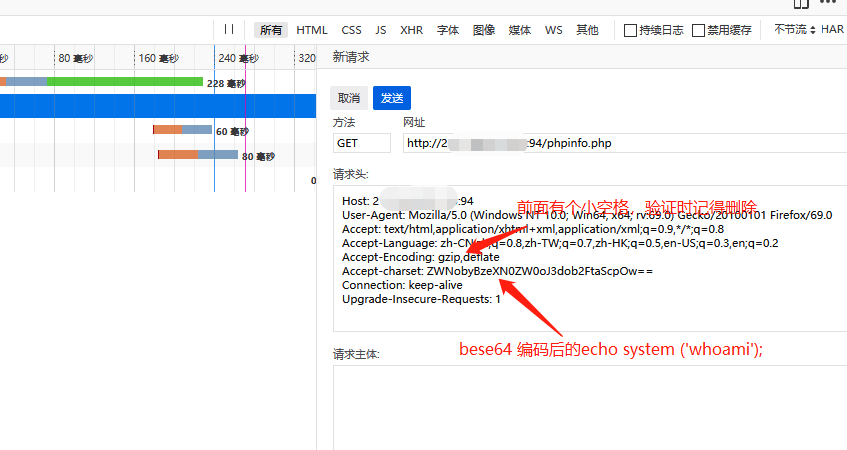
Phpstudy?这让我不禁想到了个把月前爆出的后门事件，也是妥妥的属于高危远程代码执行漏洞啊。至于phpstudy后门事件我这里也不多说了，就简单的说一说所可能存在后门的特性。

后门代码存在于\ext\php\_xmlrpc.dll模块中的，这里我们可以打开phpinfo.php文件通过浏览器Ctrl+f键快速的搜索查看是否它调用了xmlrpc这个模块。

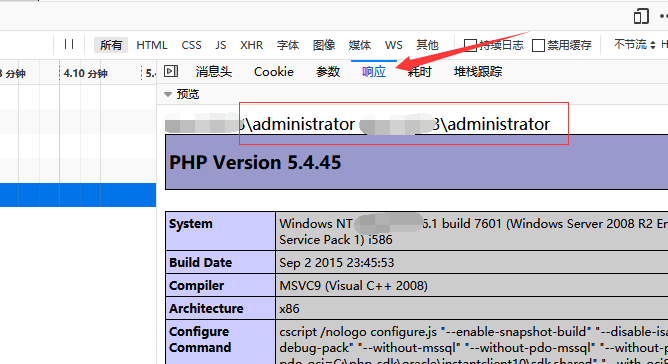


果然是有调用这个xmlrpc模块的，我估计这个病毒的传播者并不是很关注安全这一块的，前面的hfs早在15年就爆出有远程执行漏洞，现在重新搭建了服务还是有后门的(不禁笑出了声，可见关注安全情报的重要性)，接下来就是验证漏洞后门是否可以利用了，咱们先用火狐浏览器对他简单的抓包在重新构造数据包发送看看。

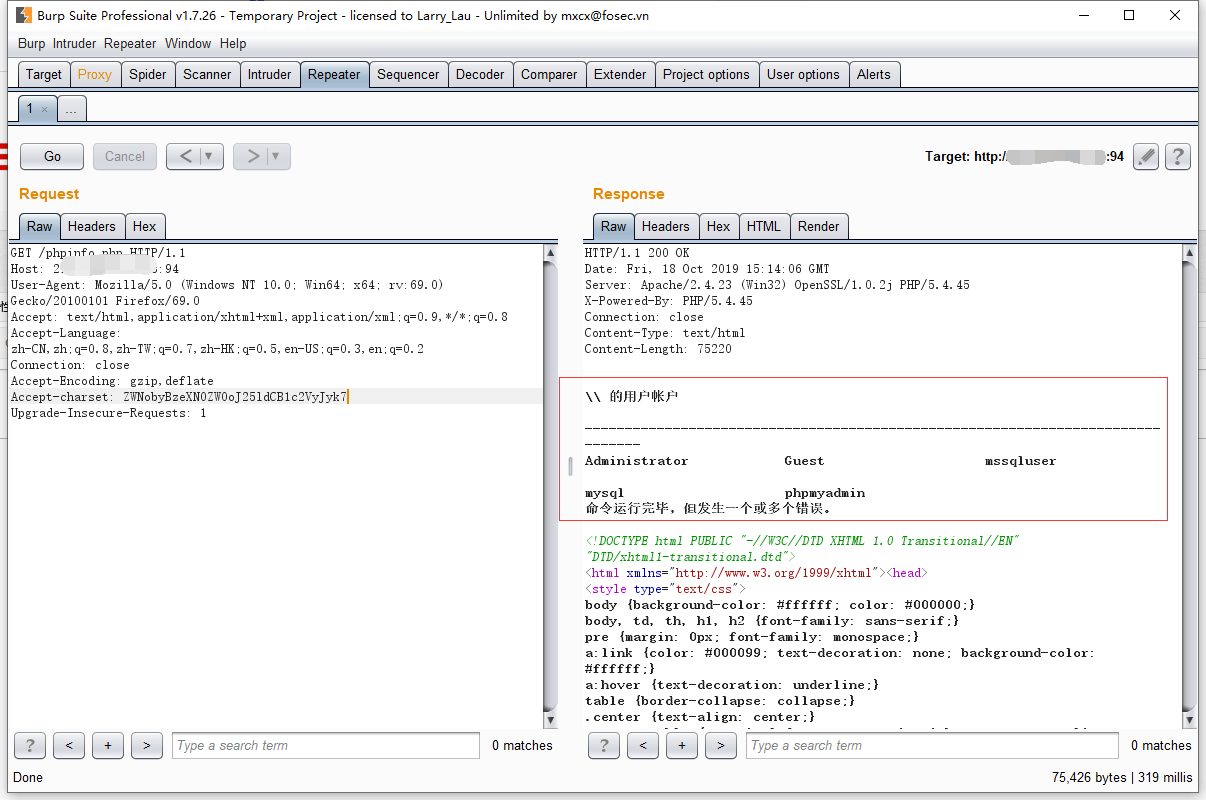
因为后门的特性所以需要对构造的恶意代码echo system(‘whoami’);进行besa64编码才可执行成功。



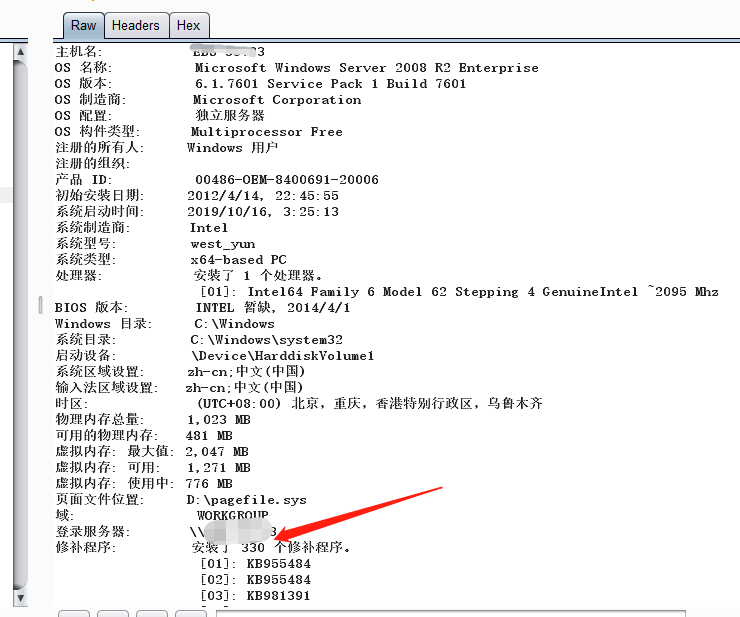
然后发完包后再点击重新发包后的连接查看所有返回的响应页面，果不其然的成功执行了whoami命令已可以确认该phpstudy是存在后门的。



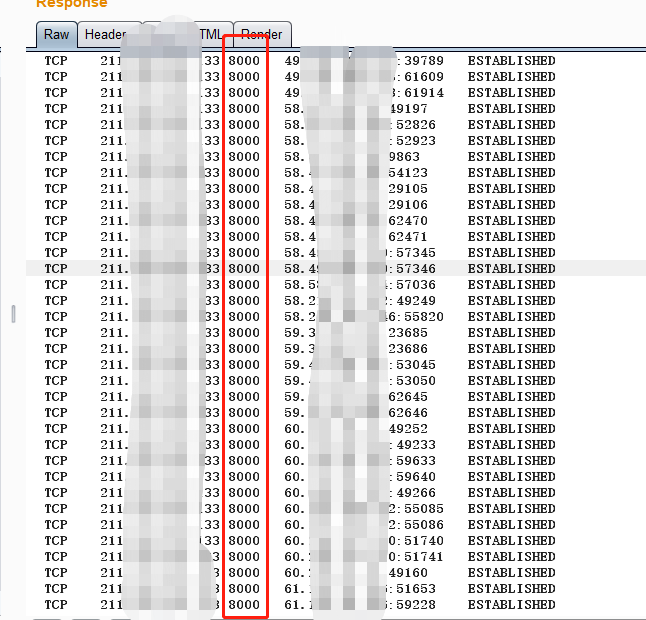
为了能更直观的展现出来也为了方便，咱们用burp来抓包发包，设置好到burp的代理后刷新一遍就可以抓到浏览器的发包了，在然后Ctrl+r发送到burp的Repeatr测试发包，这里试试net user的命令看看都有哪些用户。



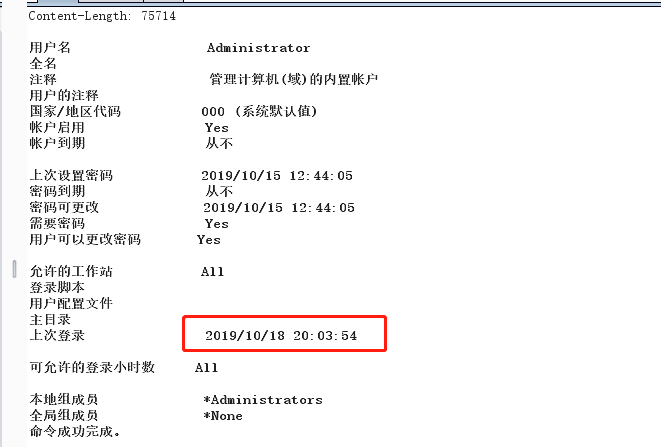
命令一样的执行成功了，在systeminfo看看系统的信息......过了许久服务器终于回包了，居然安装了330个补丁包！！！估计是关他服务器时感到了警觉把能打的补丁都打上了吧，可他万万没想到漏洞不是出在系统的。



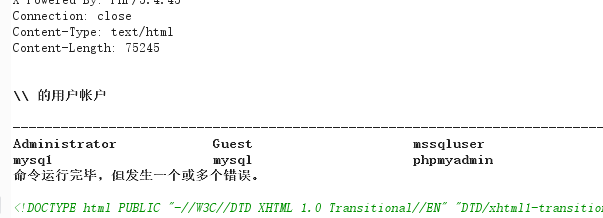
接下来在netstat -an看看都开了些什么端口，好对下一步进行工作有帮助，果然是控制肉鸡的服务器，大量的ip跟服务器的8000端口有tcp流量。



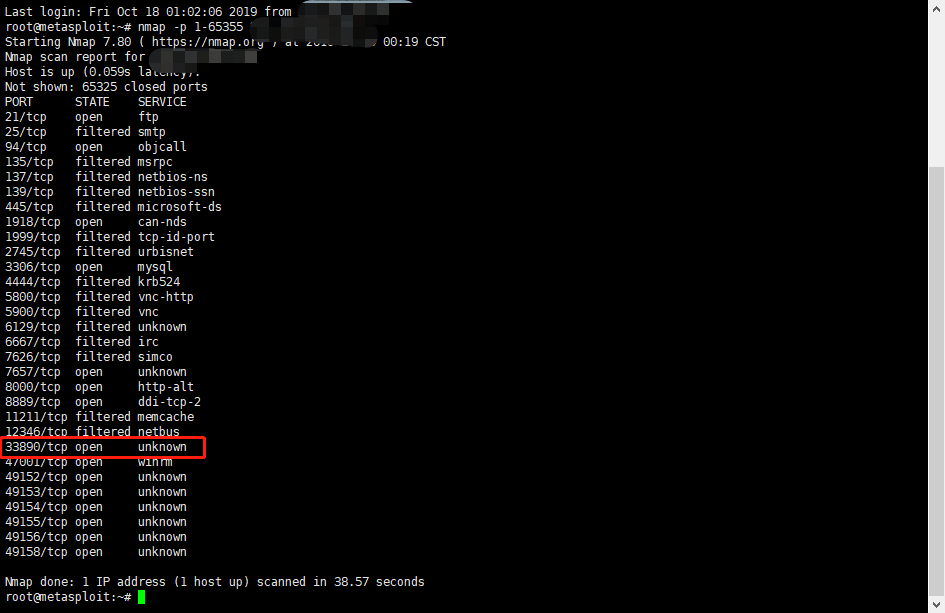
为了不引起管理员的警惕这里使用net user administrator看看管理员账号上次的登陆时间是多少然后我们在添加自己的账号进去，这里算是个小技巧吧不像第一次验证漏洞的可用性时直接把人家机子给关机了，八点的时候登陆的，现在已经凌晨了，估计管理员已经睡觉了可以放开手去干了。



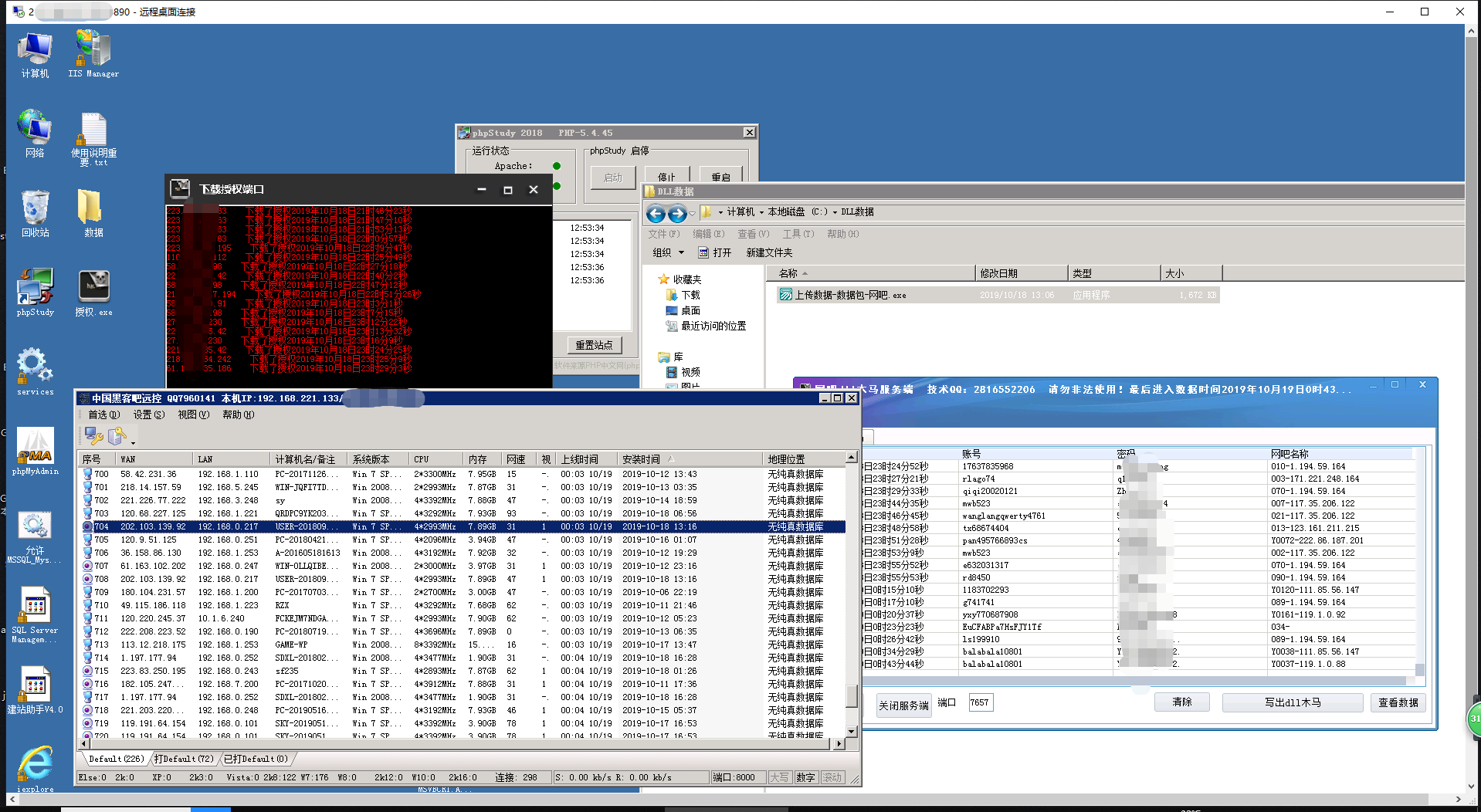
之前net user的时候有mysql这个账号那么我们就新建个mysq1的账号把L模仿成数字1，或者可以在账号后面加个$用于隐藏账号，这样至少在管理员粗心的情况下新建的账号能存活一段时间。



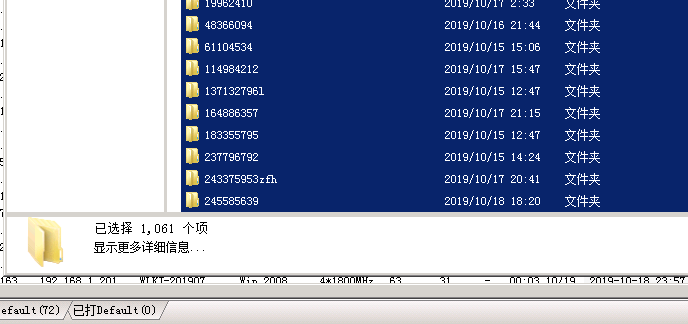
万事具备只欠连接端口了，尝试了一下默认端口3389发现连接失败，我估计可能是管理员修改了默认端口，没事，上nmap来一波-p 1-65355端口扫描，把所有开放的端口都给它扫出来，可以看到原有的3389被管理员修改成了33890。



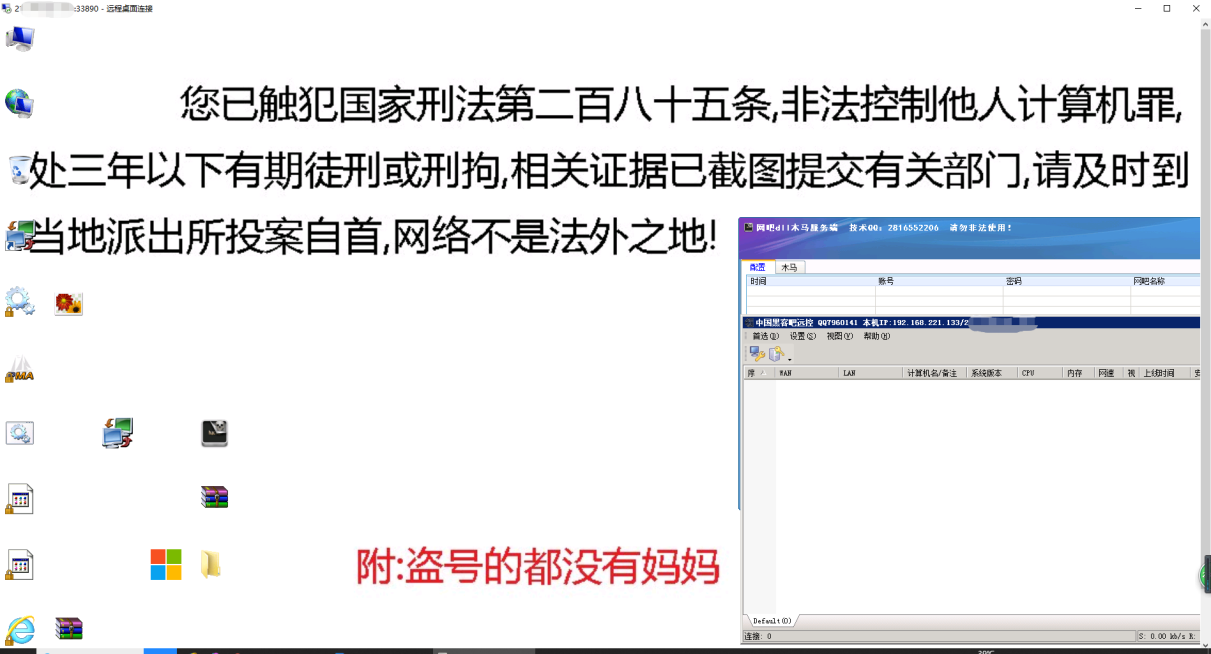
开启小飞机代理连接之.......成功拿下病毒代理服务器。



翻了翻资料，目前已经盗取到了1061个吃鸡账号。



盗号的太可恶了，把他的远控马全给下了也把盗到的账号数据也给清空了，最后留张图清日志走人。



**注：任何未经授权的入侵都是违法的，本文只供参考与学习！**