|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名：崔文帅 | | 学号：2025060163 | 专业年级： 20网工 | 班级： 网三 |
| 分组： | | 实验室：402 | 指导教师：郭念 | 实验日期：2023-05-12 |
| **实验的准备阶段**  **(指导教师填写)** | **课程名称** | **网络协议分析与攻防技术** | | |
| **实验名称** | 实验十一 Web攻击技术 | | |
| **实验目的** | 1)     了解http头部信息  2)     了解二次注入机制 | | |
| **实验内容** | 1)     User agent   描述浏览器信息  2)  二次注入漏洞是一种在Web应用程序中广泛存在的安全漏洞形式。相对于一次注入漏洞而言，二次注入漏洞更难以被发现，但是它却具有与一次注入攻击漏洞相同的攻击威力。  1)     黑客通过构造数据的形式，在浏览器或者其他软件中提交HTTP数据报文请求到服务端进行处理，提交的数据报文请求中可能包含了黑客构造的SQL语句或者命令。  2)     服务端应用程序会将黑客提交的数据信息进行存储，通常是保存在数据库中，保存的数据信息的主要作用是为应用程序执行其他功能提供原始输入数据并对客户端请求做出响应。  3)     黑客向服务端发送第二个与第一次不相同的请求数据信息。  4)     服务端接收到黑客提交的第二个请求信息后，为了处理该请求，服务端会查询数据库中已经存储的数据信息并处理，从而导致黑客在第一次请求中构造的SQL语句或者命令在服务端环境中执行。  5)     服务端返回执行的处理结果数据信息，黑客可以通过返回的结果数据信息判断二次注入漏洞利用是否成功 | | |
| **实验类型**  （打☑） | ☑验证性 □演示性 □设计性 □综合性 | | |
| **实验的重点、难点** |  | | |
| **实验环境** | Windows 2008 + phpstudy(php5.3)  代码：C:\tools\http头部. txt | | |
| **实验的实施阶段** | **实验步骤及实验结果** | **一、         User agent**  1.1   打开浏览器 ，输入http://192.168.1.3/sqli-labs-master/Less-18/ 并回车，页面显示的表单结构，为post提交。**实验截图如下:**    1.2   输入admin1/admin1，点击提交。页面多出一行浏览器版本等信息。**实验截图如下:**    1.3   打开源码(C:\phpStudy\WWW\sqli-labs-master\Less-18\index.php)，可见其将http的user agent中的信息保存到数据库中并输出。**实验截图如下:**    1.4   单击浏览器右上角的 live http headers插件。**实验截图如下:**    1.5   在表单中再次输入admin1/admin1，单击提交，在live http headers插件中出现http包，选择最上面的一行，单击下方的 Replay… 按钮。**实验截图如下:**    1.6   在新弹出的窗口中选择user-agent 标签 ，将后面的浏览器信息全部删除，替换成构造好的语句（  'and extractvalue(1,concat(0x7e,(select @@version),0x7e)) and '1'='1），点击replay按钮 。**实验截图如下:**    1.7   点击replay按钮后，页面自动跳转，可见多了一行数据库的版本信息‘5.5.47’。**实验截图如下:**    **一、         二次注入**  1.1   打开浏览器输入http://192.168.1.3/phpMyAdmin 并回车，账号为root，密码root，登录phpmyadmin，选择security。**实验截图如下:**    1.2   选择users数据表，可见用户名与密码，本次实验选择更改admin用户的密码（此时密码为admin）。**实验截图如下:**    1.3   打开新标签页，输入http://192.168.1.3/sqli-labs-master/Less-24/，回车，在页面出现后点击”New User click here？“链接。**实验截图如下:**    1.4   注册一个新的账号“ admin’# “ 密码”123456”。**实验截图如下:**    1.5   刷新phpmyadmin页面，可见我们刚注册的账号。**实验截图如下:**    1.6   返回关卡网页，登录刚才注册的账号“admin’#”，密码”123456”，登录后自动跳转更改密码的页面 ,新密码为Simple，并单击reset按钮。**实验截图如下:**    1.7   此时回到phpmyadmin页面，刷新可见admin的密码已经更改为Simple。**实验截图如下:** | | |
| **实验结果的处理阶段** | **实验结果的分析与总结** | 1.本次实验修改密码所提交的sql语句是什么，执行时的语句又是什么？  Update password=’Simple’ where username=’admin’#’  实际执行的：update password=’Simple’ where username=’admin’ | | |