**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机安全导论**

**实验项目名称­： 密码学实验**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 闫巧**

**报告人： 刘俊楠 学号： 2017303010 班级： 01**

**实验时间： 2021.11.25**

**实验报告提交时间： 2021.11.30**

**教务部制**

|  |
| --- |
| 实验目的与要求：  1. 掌握Web漏洞中的命令执行漏洞的形成原理，攻击方法及防御方法。  2. 掌握Web漏洞中的SQL注入漏洞的形成原理，攻击方法及防御方法。  3. 掌握Web漏洞中的XSS漏洞的形成原理，攻击方法及防御方法。 |
| 实验环境：  DWVA漏洞实验靶场  Chrome浏览器 |
| 实验原理： |
| 实验内容：   1. Command Injection，设定DVWA Security(Low and Medium)------------------2分 2. SQL Injection，设定DVWA Security(Low and Medium)---------------------------1.5分 3. XSS(Stored)，设定DVWA Security(Low and Medium)-----------------------------1.5分 |
| 实验步骤与结果：  DVWA靶场搭建  1、下载DVWA    2、打开phpstudy，配置mysql环境，启动apache mysql 以及ftp。    3、将下载好的压缩文件解压在phpstudy相关目录下。    4、修改 config.inc.php 里代码如下：    5、打开<http://127.0.0.1/DVWA-master/setup.php>，显示如下：    6、修改 config.inc.php 里秘钥如下：    7、创建成功后，输入用户名admin与密码password，进入界面    8、打开<http://127.0.0.1/DVWA-master/security.php>，并将其安全性设置成低，实验目的达到！。    **Command Injection**  **简单（low）级别**  1、点击Command Injection，输入127.0.0.1，我们可以看到ping得的ip地址返回了四次。    2、我们点击view source，查看源代码，我们可以看到，服务器通过判断操作系统执行不同ping命令，但是对ip参数并未做任何的过滤，导致了严重的命令注入漏洞。    3、因为其不会过滤命令，所以我们利用无论是linux系统还是windows系统，都可以用&&来执行多条命令语句。所以我们输入的ip地址可以是127.0.0.1&&whoami，可以发现返回了执行结果，其输出了我的用户，也就是本机。    中等（medium）级别  1、我们将等级调整为中等级别，查看command injection的源代码。我们可以看到其删除了&&与;的两个指令，只要有则将其转换为空指令，但是其并未删除&指令。    2、于是我们将ip地址改成127.0.0.1&whoami，可以看到还是输出了我们本机的用户，也就是我的用户。    SQL Injection  简单（low）等级   1. 我们修改等级为low，然后点击SQL injection，查看源代码，分析得到：  * 当其接收到submit指令后，不经过任何检查就将其拼接到查询语句并且返回。 * 只要有冒号结束，就可以返回查询语句结果。 * 故可能存在字符型sql注入      1. 接下来我们测试sql是否存在注入以及注入的类型，输入1' and '1'='1看到返回如下类型。     3、我们输入1' order by 2# 以及1' order by 3#以此来猜解SQL查询语句中的字段数，我们可以看到，SQL语句查询的表的字段数是2       1. 输入1' union select 1,2#，确定回显位置，我们可以看到回显位置为蓝框所示。     4、我们输入1' union select version(),database()#，以此来查询当前版本与数据库，可以看到当前版本为5.7.26，数据库为dvwa。    中等（medium）等级  1、我们将等级调整为medium，回到sql injection ，点击源码发现，其在源码中加入了一些防御，在网站中不需我们输入，只允许我们选择。      2、我们在源码中看到其使用了mysql\_real\_escape\_string函数转义sql语句中的一些特殊字符，说明其可能存在数字型sql注入。    3、打开Burp Suite，设置好代理，在火狐浏览器打开http://127.0.0.1/DVWA-master/vulnerabilities/sqli/并点击submit，可以看到抓到了包，我们进行更改。     1. 将抓到的包中的id=1改成id= 1' or 1 = 1#，点击forward，可以看到出错：   You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\' or 1 = 1#' at line 1      5、出错说明服务器端对特殊符号(')进行了转义。那我们就试试数字型注入。重新抓包构造payload：1 or 1 = 1，然后提交。    6、发现注入成功！可以看到查询成功且存在的sql注入类型是数字型。    7、再次点击submit，截得包并修改id为1 and 1=2 union select database(),2，在返回值中的first name中发现了该数据库的名称。      XSS(Stored)  简单（low）等级   1. 我们将安全等级调成low，点击XSS（Stored）查看源代码。      1. 可以看出，low级别的代码对我们输入的message和name并没有进行XSS过滤，而且数据存储在数据库中，存在比较明显的存储型XSS漏洞。我们输入 name为1 和 Message为<script>alert('hack')</script> ，可以看到，我们的js代码立即就执行了。      1. 此时我们发现了存储型XSS漏洞，此时我们可以修改攻击载荷<scirpt></script>，加入一些恶意代码执行。     中等（medium）等级  1、我们查看中等等级的源代码，发现与low等级并无变化，但是分析发现，由于对message参数使用了htmlspecialchars函数进行编码，因此无法再通过message参数注入XSS代码，但是对于name参数，只是简单过滤了<script>字符串，仍然存在存储型的XSS。    2、当我们重复上个实验low等级的操作时，发现其将我的代码转义掉了。    3、我们通过Burpsuite抓包改name参数为:<sc<script>ript>alert(/name/)</script>。    4、发现可以绕过。    5、name中的1消失了，说明命令执行成功！ |
| 实验结果分析：   1. 通过本次实验，我明白了web漏洞中的命令执行漏洞原理是指服务器没有对执行的命令进行过滤，用户可以随意执行系统命令，命令执行漏洞属于高危漏洞之一，也属于代码执行的范围内。  * 攻击者可以使用&、&&、|、||等命令拼接自己想要查看的信息的相关命令，这样，连同攻击者的命令就会一块执行，这样就造成了命令执行漏洞。 * 我们可以通过尽量不要执行一些外部命令；使用自定义函数或函数库来代替外部命令；使用一些过滤函数来对用户输入的命令参数进行过滤；等方法进行防御。  1. 我明白了web漏洞中的sql注入漏洞原理是指使用数据库的web程序没有对输入做严格的过滤，用户可以通过修改输入参数来构造并提交SQL，这使得攻击者有机会将程序命令当作用户输入的数据提交给web程序， 最终获取系统信息、甚至执行危险代码或系统命令。  * 攻击方式有数字型、字符型与搜索型注入方式。 * 可以使用安全的API(应用程序编程接口)；对输入的特殊字符进行Escape转义处理；以及使用白名单来规范用户的输入，对客户端进行控制，不允许输入SQL注入相关的特殊字符。  1. 我明白了web漏洞中的xss漏洞原理是指即跨站脚本攻击，某种意义上也是一种注入攻击，是指攻击者在页面中注入恶意的脚本代码，当受害者访问该页面时，恶意代码会在其浏览器上执行。  * 攻击方式有两种，一种通过外部输入然后直接在浏览器端触发，即反射型XSS；还有一种则是先把利用代码保存在数据库或文件中，当web程序读取利用代码并输出在页面上时触发漏洞，即存储型XSS。 * 我们可以通过特殊字符HTML实体转码来实现防御。最好的过滤方式是在输出和二次调用的时候进行加HTML实体一类的转码，防止脚本注入。也可以通过标签事件属性黑名单来实现防御。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。