

Terminal Challenge--The Name Game (part 1)



PowerShell Command Injection

The term “command injection” refers to a vulnerability where a program does not properly check user input. It allows attackers to execute commands in the program by entering commands into the form (web page, whatever.) It’s a vulnerability that has been around for a long time and appears again and again in code written for almost every language and operating system.

Step 1 Reconnaissance

First determine if the application is potentially vulnerable to command injection. If you haven’t read the articles mentioned in the hints, read the one in the PowerShell Command Injection now.

<https://ss64.com/ps/call.html>

A good first step is to enter special characters mixed in with regular alpha-numeric characters into all the fields. The article tells you which characters might work. If you can get the application to generate helpful error messages.

Step 2 Inject Commands

Once you have found a vulnerable field, try to inject commands. The semicolon is useful for this, since it is used to separate commands that are entered in one line. For example, `command 1; command 2; command 3`. This works in many languages and OSs and in the language this site appears to use, PowerShell. So, you can finish the command the application is running with a semicolon and then add your command. Simple commands to test with could be things like, `echo isthisworking`, `dir`, or `ls`.

Hand In

- 1) Which field is vulnerable to command injection?
- 2) Hand in a screenshot where you successfully inject a command.