VotD: OS Command Injection

Attack Pattern: See **CAPEC-88** (https://capec.mitre.org/data/definitions/88.html)

Weakness/Vulnerability: See **CWE-78** (http://cwe.mitre.org/data/definitions/78.html) .

Description: Your web application is written in some programming language (Python, JavaScript, ASP.NET, PHP) running on top of an Operating System (Windows, Linux). All programming languages have functions that allow them to execute operating system commands directly. For example, the following Python code will call the Windows command to list the contents of the current directory:

```
import os
os.system("dir")
```

OS Command Injection falls under the general category of "things that can be avoided if you perform proper input validation."

DVWA and Fuzzer:

- Log in to your DVWA installation and browse to the 'Command Injection' page.
- This page uses the operating systems 'ping' command to check the time it takes a DNS or IP address to receive a network packet. Try 127.0.0.1 (your machine) and uncw.edu in the form.
- Now try '127.0.0.1 && dir c:\' on Windows or '127.0.0.1 && ls /' on Linux/Mac
- You can, of course, run much more destructive commands, like those that delete files and directories.

Mitigations:

- Generally speaking, avoid entirely or be careful using these generic "run in the OS" calls. Usually, APIs exist for specific OS calls that can accomplish the same thing without the danger of injection. For example, in Python, **os.listdir()** (https://docs.python.org/3/library/os.html#os.listdir) can give you a list of all files and directories.
- Another good practice is to craft a "white list" that specifies what input is acceptable, rather than trying to sanitize the input string by removing dangerous characters.