PicoCTF: caas challenge

Sam Newman

April 2025

1 Website in Question

This website is one whose only functionality is to return the user's url message in a cool cow speech bubble from ASCII characters. You input the speech to be bubbled in the https://caas.mars.picoctf.net/cowsay/message part of the url.

2 Business Impact

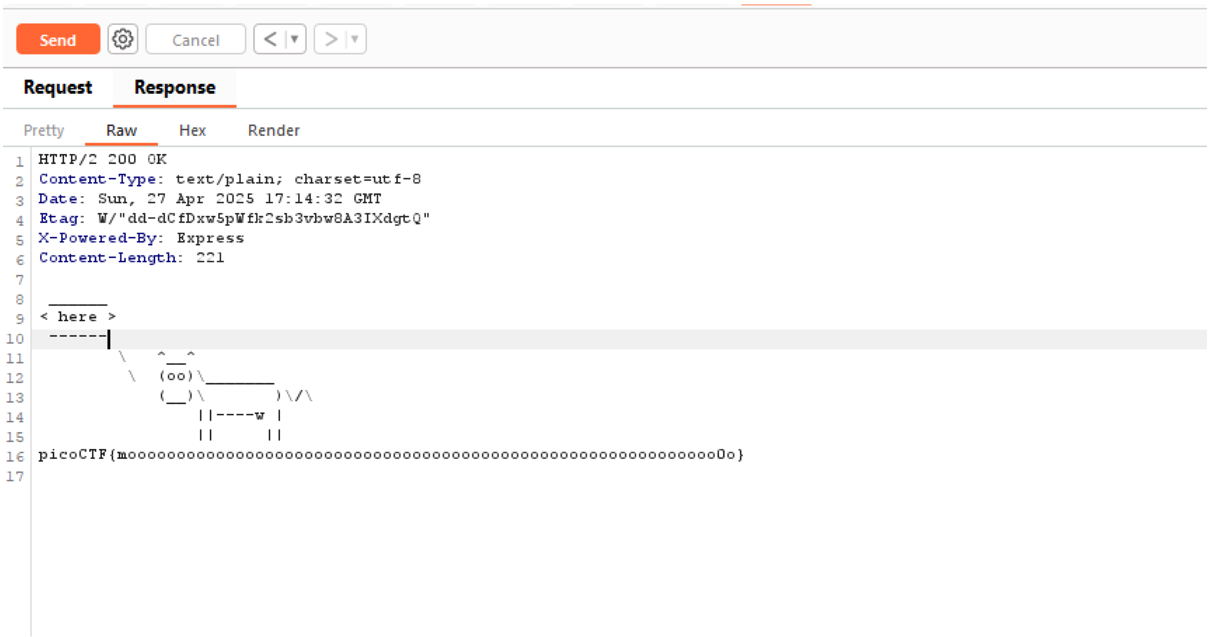
This vulnerability could allow an attacker to easily access any and all parts of the host machine system. In this case, we could view the flag and all files in the system directory which could include passwords, bank accounts, all user settings etc. We could essentially also execute any shell commands on the host machine system. It appears that we cannot gain sudo access as there is a 500 internal server error, but we can execute any user mode commands.

3 How It Works

The vulnerability is an extremely obvious one. The host system uses exec() to execute the command, which executes shell commands. This allows us to put any valid shell command into the URL and it will be executed by the host system as a shell command. All we have to do is input any message, like moo, and attach shell command to execute our code after the message runs. I began by simply doing moo ls to view the files in the folder, and then saw falg.txt which I then used moo cat falg.txt to open and view on the webpage.

4 Why It Works + Mitigation

The vulnerability works by exploiting the developer's use of the exec() command, and the fact that they trusted user input into it. This should never be done, and exec() should never be used in conjunction with any unverified user input. Simple checking of user input or not using the unsafe exec() function would prevent this attack.



```
HTTP/2 200 OK
Content-Type: text/plain; charset=utf-8
Date: Sun, 27 Apr 2025 17:14:32 GMT
Etag: W/"dd-dCfDxw5pWfk2sb3vbw8A3IXdgtQ"
X-Powered-By: Express
Content-Length: 221

 ______
< here >
 ------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
picoCTF{mooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo0o}

```

Figure 1: Solved Lab