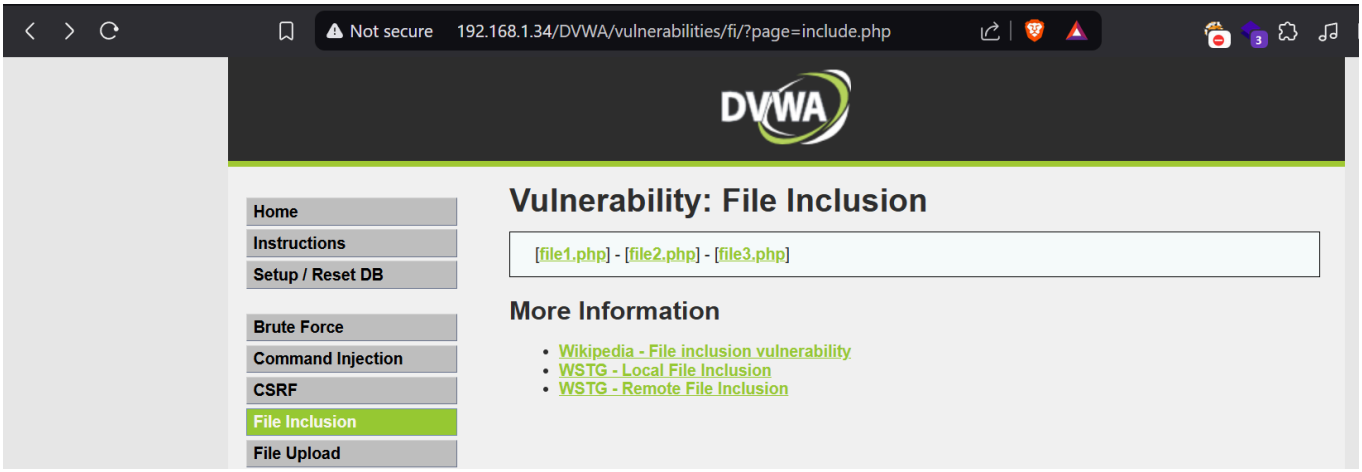
04. File Inclusion

LFI Low Level



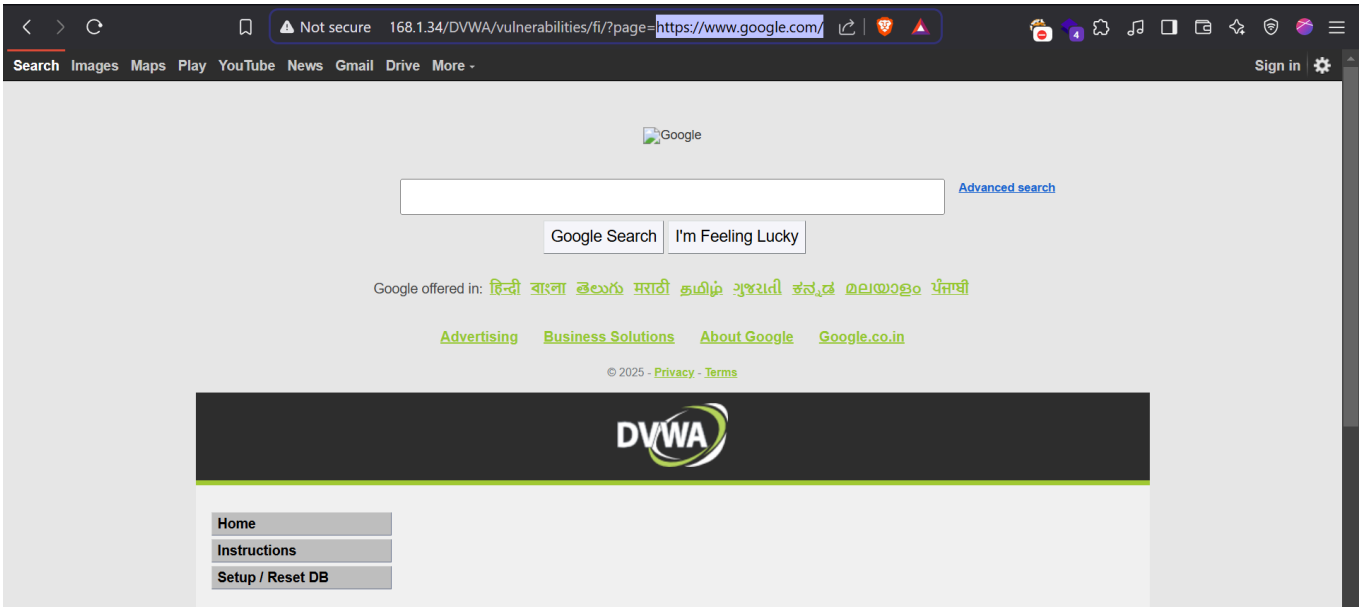

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

?>
```

there is no sanitization method in backend code

If we pass any file location path in the url as a parameter value it will be included in web page


```
../../../../../../../../etc/passwd
```

RFI (Remote File Inclusion) Low Level




```
We can give link to our web shells and try to gain reverse shell
```

LFI & RFI Medium Security

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\\" ), "", $file );

?>
```

```
here sanitization is applied we can see in backend source code
```

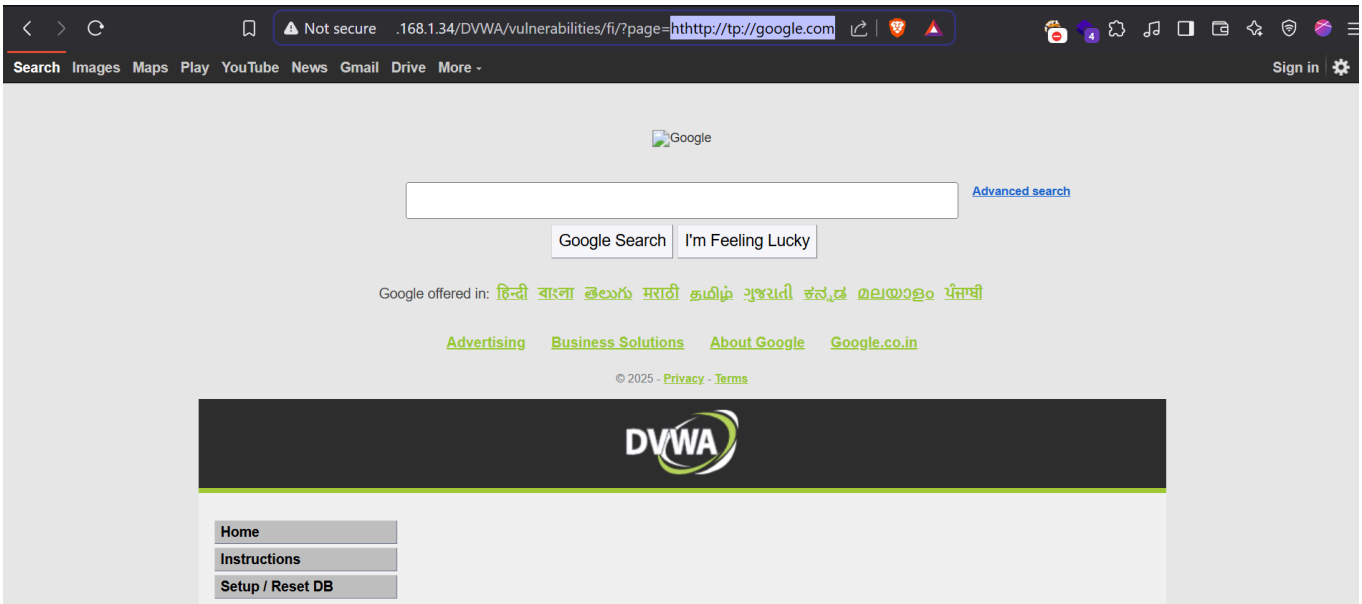
1. The code says it will not allow `"http://", "https://"`
2. As well these two characters are also banned `"../", "..\\"`

To bypass this blacklisting mechanism we can use this in value `.../.`

we can use this kind of payloads for bypassing these mechanism

```
..././..././..././..././..././..././etc/passwd
```

- For RFI blacklisting mechanism bypass we can use `hthttp://tp://`




```
hthttp://tp://google.com
```

LFI and RFI High Level Security

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
if( !fnmatch( "file*", $file ) && $file != "include.php" ) {
    // This isn't the page we want!
    echo "ERROR: File not found!";
    exit;
}

?>
```

This level of security have input validation and we can see that only file name is allowed here and so there is no chances of RFI vulnerability.

```
This is only including file name to bypass this we can include these 2
methods:
```

1. `file://`

2. `encoding mechanism`

```
file.php%0A../../../../../../../../etc/passwd
```

`%0A` which is the **URL-encoded version of the newline character (`\n` or `LF` in ASCII**)

Impossible LFI and RFI

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Only allow include.php or file{1..3}.php
$configFileNames = [
    'include.php',
    'file1.php',
    'file2.php',
    'file3.php',
];

if( !in_array($file, $configFileNames) ) {
    // This isn't the page we want!
    echo "ERROR: File not found!";
    exit;
}

?>
```

This code is uses whitelisting approach so it will include only particular type of input

---