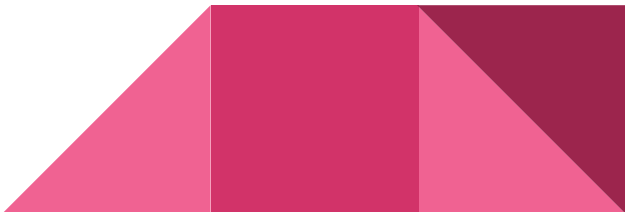


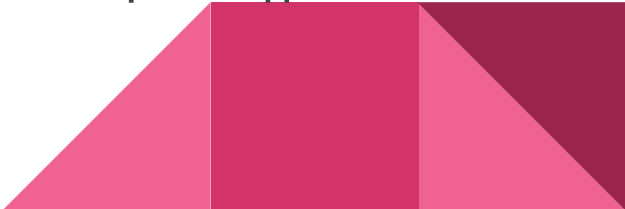
# Local File Inclusion (LFI)

# Τι είναι;

- Η ευπάθεια LFI δίνει τη δυνατότητα σε έναν επιτιθέμενο να συμπεριλάβει (include) ένα αρχείο σε μία σελίδα ενός server. Το αρχείο αυτό βρίσκεται εντός του server.
  - Αυτό συνήθως οφείλεται σε απουσία κατάλληλου ελέγχου της εισόδου του χρήστη από τη μεριά του εκάστοτε web application (unsanitized user input)
  - Με τον όρο include εννοούμε άλλοτε να γίνει απλώς render (προς ανάγνωση), άλλοτε πάλι να εκτελεστεί
- 

## Ε και;

Πρόκειται για σημαντική ευπάθεια καθώς μπορεί να οδηγήσει σε:

- Code execution στον web server
  - Sensitive Information Disclosure
  - Συνδυασμό με άλλες τεχνικές exploitation (για παράδειγμα XSS σε client side)
- 

# Typical PHP LFI case

Έστω ότι υπάρχει το παρακάτω php script (pages.php) εντός του server

```
<?php
    $filename = $_GET['page'];

    include('pages/' . $filename);
?>
```

Τι θα συμβεί εάν κάποιος ζητήσει τη σελίδα

<http://vuln.com/pages.php?page=../../etc/passwd>



# OOPS

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
alex:x:500:500:alex:/home/alex:/bin/bash
margo:x:501:501::/home/margo:/bin/bash
...
```

# Ακόμα χειρότερα...

Σε περίπτωση που έχει προηγηθεί το upload ενός malicious php script (**reverse shell reference!!**) από τον επιτιθέμενο στον server, τα πράγματα μπορεί να γίνουν πολύ χειρότερα

Τι θα συμβεί εάν κάποιος ζητήσει τη σελίδα

<http://vuln.com/page.php?page=../../uploads/malicious.php>



# RIPOLUS

```
(kali㉿kali)-[~/Downloads]  
$ nc -lnvp 1234  
listening on [any] 1234 ...  
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 35722  
whoami  
www-data  
█
```

## More LFI

LFI μπορεί να προκύψει όχι μόνο σε PHP αλλά και σε άλλες γλώσσες. Ακολουθεί παράδειγμα σε Python.

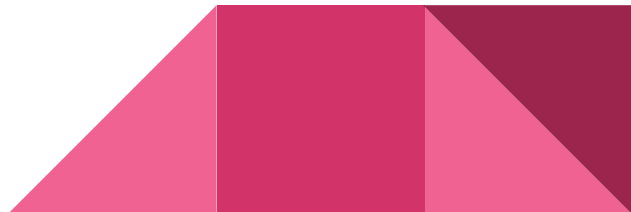
```
@app.route("/read_file")
def read_file():
    filename = request.args.get('filename')
    file = open(filename, "r")
    data = file.read()
    file.close()
    return jsonify(data=data),200

@rpc(String, _returns=String)
def read_file(ctx,file):
    file = open(file, "r")
    data = file.read()
    file.close()
    return(data)
```



# Παρατηρήσεις

- Προφανώς τα παραδείγματα που προηγήθηκαν είναι αρκετά απλά. Ωστόσο, οπουδήποτε υπάρχει input χρήστη με σκοπό την εμφάνιση (ή και επιλογή) αρχείου του server (`http://domain.com/<random>?file=...`), είναι πιθανό να υπάρχει ευπάθεια LFI.
- Τις περισσότερες φορές χρειάζεται `manually testing` για την εύρεση του σωστού path για το εκάστοτε αρχείο που θέλουμε (εφόσον υπάρχει LFI).
- Το Path Traversal μπορείτε να το σκέφτεστε ως υποκατηγορία του LFI



# Resources

- [https://owasp.org/www-project-web-security-testing-guide/v42/4-Web\\_Application\\_Security\\_Testing/07-Input\\_Validation\\_Testing/11.1-Testing\\_for\\_Local\\_File\\_Inclusion](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_Local_File_Inclusion)
  - <https://tryhackme.com/room/filepathtraversal>
  - <https://portswigger.net/web-security/all-labs#path-traversal>
- 