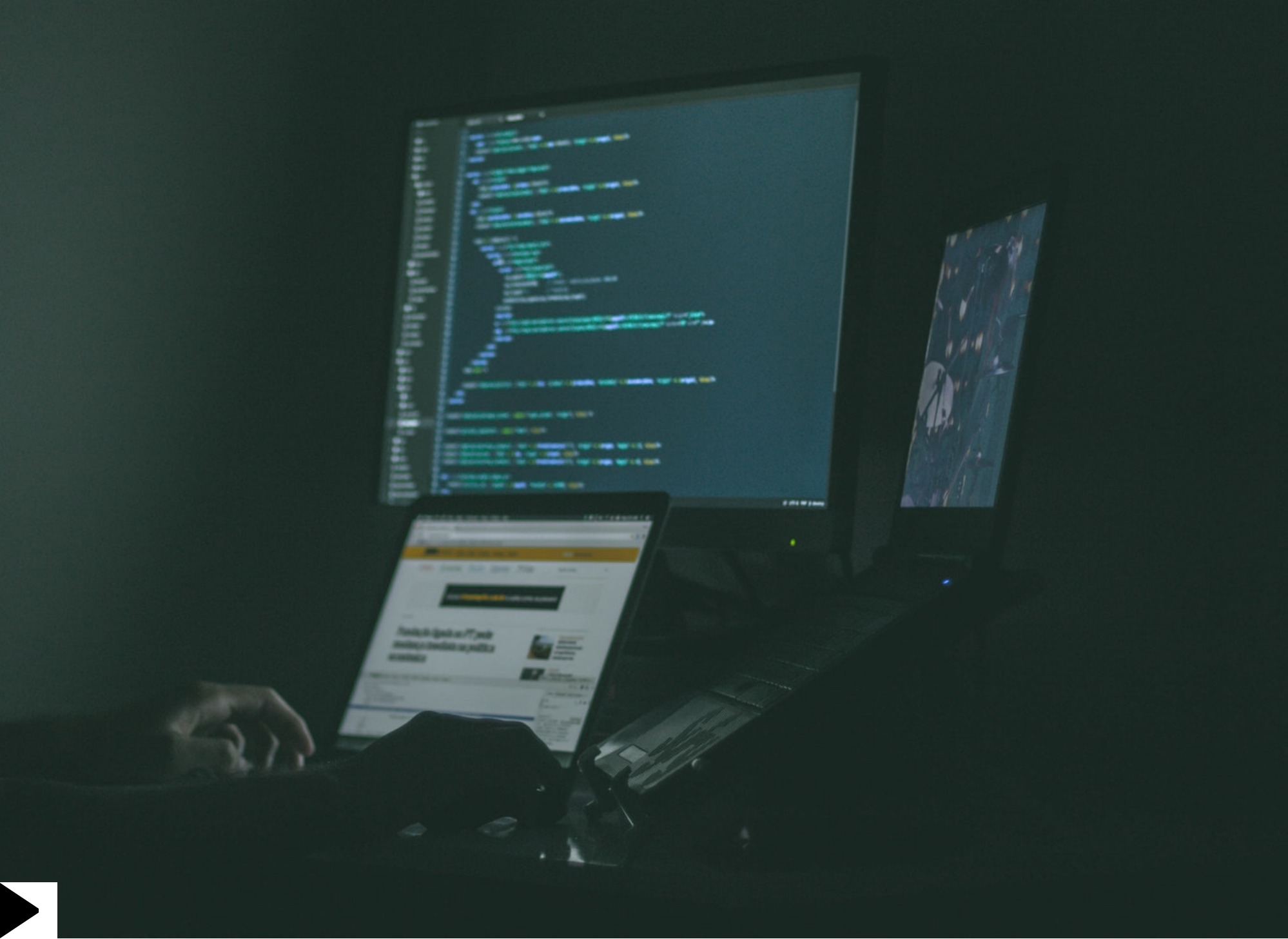
LFI / RFI

Easy access to your server



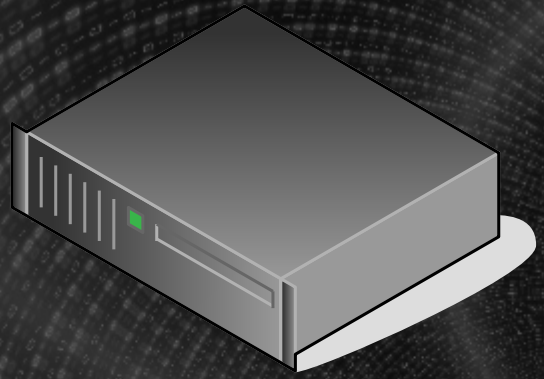

HOW TO GET IN?


HOW TO GET IN?
WHY NOT JUST USE THE URL BAR?


LFI = Local File Inclusion


RFI = Remote File Inclusion

Client



Server

PHP

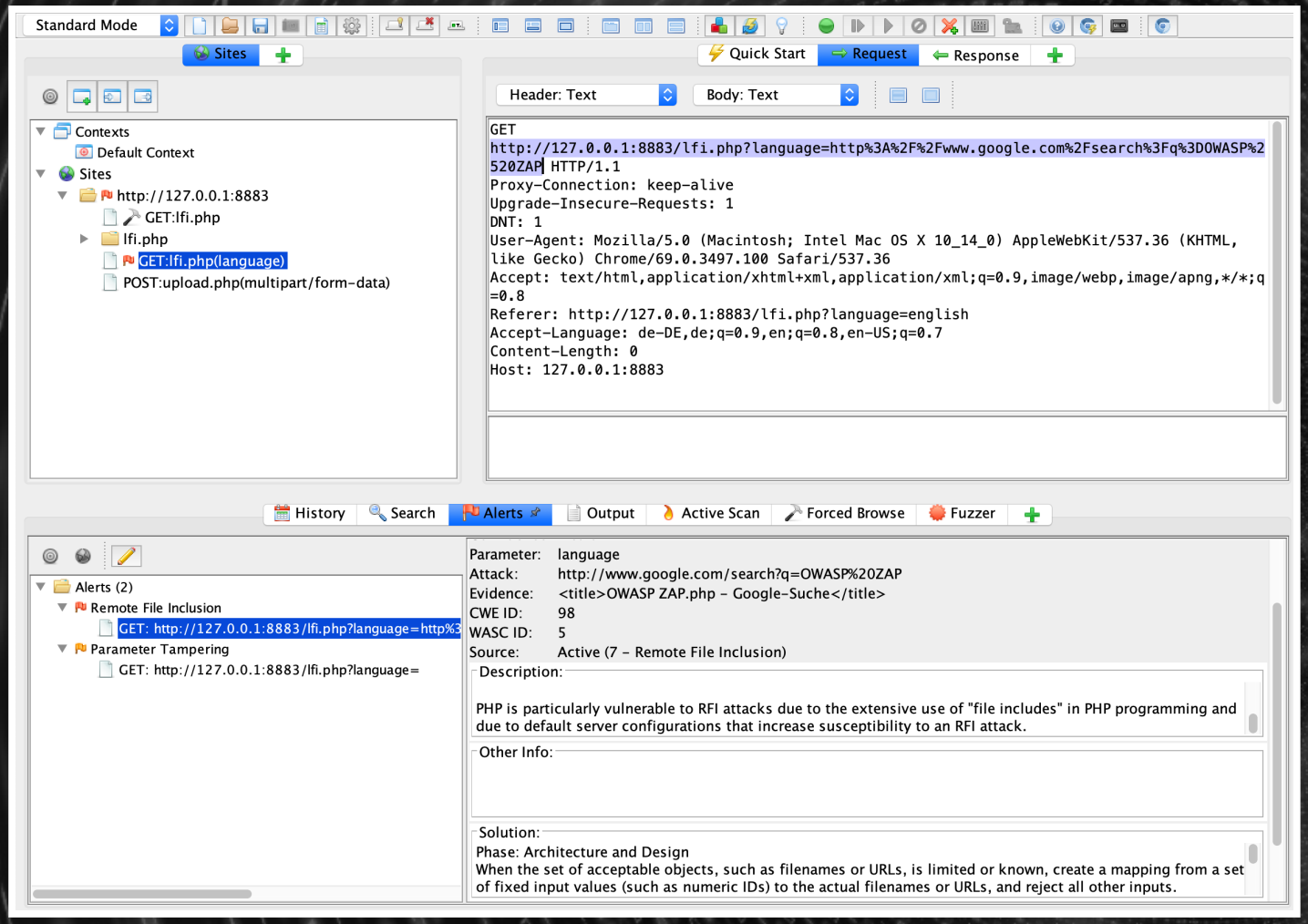
```php
<?php
        include( $_GET['pagetoload'] . '.php' );
?>
```

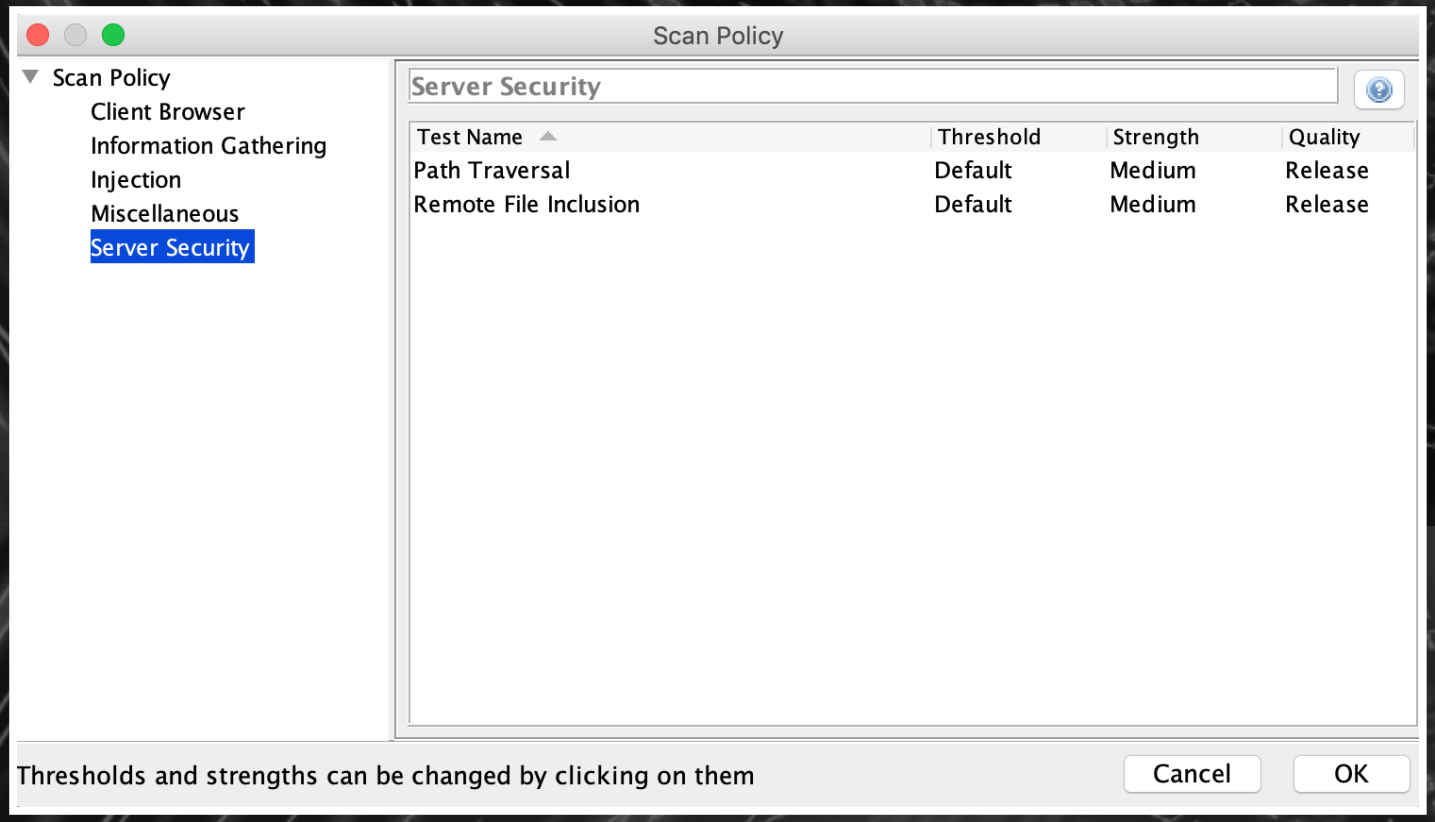
JSP

```
<head>
  <%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
</head>
...
<c:import url = "<%= request.getParameter(\"pagetoload\")%>"/>
```

ASP.NET RAZOR

```
@Html.Partial(HttpContext.Request.Query["pagetoload"])
```


Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
http://127.0.0.1:8883
GET:lfi.php
lfi.php
GET:lfi.php(language)
POST:upload.php(multipart/form-data)
GET
http://127.0.0.1:8883/lfi.php?language=http%3A%2F%2Fwww.google.com%2Fsearch%3Fq%3DOWASP%2520ZAP HTTP/1.1
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
DNT: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://127.0.0.1:8883/lfi.php?language=english
Accept-Language: de-DE,de;q=0.9,en;q=0.8,en-US;q=0.7
Content-Length: 0
Host: 127.0.0.1:8883
History
Search
Alerts
Output
Active Scan
Forced Browse
Fuzzer
Alerts (2)
Remote File Inclusion
GET: http://127.0.0.1:8883/lfi.php?language=http%3
Parameter Tampering
GET: http://127.0.0.1:8883/lfi.php?language=
Parameter: language
Attack: http://www.google.com/search?q=OWASP%20ZAP
Evidence: <title>OWASP ZAP.php - Google-Suche</title>
CWE ID: 98
WASC ID: 5
Source: Active (7 - Remote File Inclusion)
Description:
PHP is particularly vulnerable to RFI attacks due to the extensive use of "file includes" in PHP programming and due to default server configurations that increase susceptibility to an RFI attack.
Other Info:
Solution:
Phase: Architecture and Design
When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs.


Scan Policy
Scan Policy
Client Browser
Information Gathering
Injection
Miscellaneous
Server Security
Server Security
Test Name
Threshold
Strength
Quality
Path Traversal
Default
Medium
Release
Remote File Inclusion
Default
Medium
Release
Thresholds and strengths can be changed by clicking on them
Cancel
OK

MITIGATION

MANAGEMENT

Analyse your app - black and white box

MITIGATION

CODING

Do not allow client control what files to load

Use metrics in your framework and language choice

And after that keep it up-to-date

- File Inclusion Attacks (Ali, Hakin9, IT-Sec-Meetup Jan 2019)
- https://secf00tprint.github.io/blog/payload-tester/lfirfi/en


https://github.com/secf00tprint/payloadtester_lfi_rfi


THANK YOU!