Local File Inclusion (LFI)

Risk: Critical	**Impact**: High	**Likelihood**: Critical

CVSS SCORE: 7.5 **CWE:** 53.0 - Base Finding: 78 - Attack Surface: 1 - Environmental: 0.697 CWSS Calculator	**CVSS_String:** **CVSS-3.1:AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N** **CWE_String:** TI:H,0.9/AP:N,0.1/AL:NA,1/IC:N,1/FC:T,1/RP:N,1/RL:NA,1/AV:I,1/AS:NA,1/IN:NA,1/SC:A,1/BI:C,1/DI:H,1/EX:H,1/EC:M,0.7/P:C,0.8/

OWASP Top 10: Security Misconfiguration

Component(s): /getfile.php?file=name.html

Description: Local file inclusion allows the threat actor to see the content of files that are available in the system or the web server.

Likelihood: I put it as a high likelihood because LFI is a common vulnerability that occurs in the webserver. The vulnerability in my webserver is easy to exploit with known exploits or payloads that are easy to use.

Impact: This is high because with this vulnerability an attacker can see information about the webserver's code. Most importantly the web server is running as root. The attacker can easily see the files that are only allowed to root users. Although the user might fully compromise the system, they can do a great amount of recon by utilizing this vulnerability.

ASVS (the requirements that might not be met):

- Level 1:
  - V1.2 Security Architecture: LFI can indicate weaknesses in the security architecture, allowing unauthorized access to files. Failure to prevent LFI may result in a non-compliance with this requirement.
  - V2.1 Data Classification (Level 1): Unauthorized access through LFI may lead to exposure of sensitive data, violating data classification requirements.
  - V9.1 Security Feature Flags (Level 1): Configuration weaknesses leading to LFI may affect the efficacy of security feature flags.

CVSS Options:

Exploitability Metrics:
Attack Vector: Network

- The attacker just needs internet access and can easily access the web server and exploit vulnerability over the internet.

Attack Complexity: Low

- The attacker can easily exploit this vulnerability with known payloads that are out there, and by looking at proof of concepts. The payload is very simple to test with.

Privilege Required: None

- The attacker does not need any privilege to exploit this vulnerability

User Interaction: None

- The reason is that this is a server side vulnerability and it only sees the content of files that on the web server

Scope: Unchanged

- The reason is that the impacted component is the same machine. The webserver and affected machines are the same component.

Impact Metrics:
Confidentiality impact: High

- It is a big risk. The reason is that it reveals the content of any files in the webserver machine. The web server runs as root.

Integrity Impact: None

- It does not change of any files content or any data

Availability impact: None

- It does not affect any availability of any files or information

<table>
<tr><th colspan="3">Reflected Cross-site scripting (XSS)</th></tr>
<tr><td><b>Risk:</b> Critical</td><td><b>Impact</b>: medium</td><td><b>Likelihood</b>: Critical</td></tr>
<tr><td><b>CVSS SCORE:</b> 7.1
<b>CWE:</b> 28.9
- Base Finding: 66
- Attack Surface: 0.985
- Environmental: 0.4445
CWSS Calculator</td><td colspan="2"><b>CVSS_String:</b>
<b>CVSS-3.1:AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N</b>
<b>CWE_String:</b>
TI:M,0.6/AP:N,0.1/AL:NA,1/IC:N,1/FC:T,1/RP:N,1/RL:NA,1/AV:I,1/AS:NA,1/IN:T,0.9/SC:A,1/BI:L,0.3/DI:H,1/EX:H,1/EC:M,0.7/P:H,0.9/</td></tr>
<tr><td colspan="3"><b>OWASP Top 10:</b> Injection</td></tr>
<tr><td colspan="3"><b>Component(s)</b>: /name.html</td></tr>
</table>

Description: In this vulnerability, the threat actor can use it to execute javascript code in the browser and the code can be malicious to steal a person's session cookie.

Impact: The attacker can use phishing email to get a person to click on a link causing execution of malicious javascript code in the victim's machine. It doesn't really affect the web server machine.

Likelihood: The exploitation is simple and easy to find out. WIth payloads that are publicly available it can be exploited and used for malicious gains.

ASVS (the requirements that might not be met):

- Level 1:
    - V1.2 Security Architecture (Level 1): Reflected XSS may indicate weaknesses in the security architecture, as it allows untrusted data to be executed in the user's browser. Failing to prevent Reflected XSS could result in non-compliance with this requirement.
    - V4.1 Session Token Protection (Level 1): Reflected XSS could compromise session security. Ensuring protection against XSS is vital to prevent unauthorized access to user sessions.
    - V6.1 Input Data Validation (Level 1): Failing to validate and sanitize input data may lead to Reflected XSS vulnerabilities. Non-compliance with this requirement could result in unvalidated data entering the application.

CVSS Options

Exploitability Metrics:

Attack Vector: Network

- The attacker just needs internet access and can easily access the web server and exploit vulnerability over the internet.

Attack Complexity: Low

- The attacker can easily exploit this vulnerability with known payloads that are out there, and by looking at proof of concepts. The payload is very simple to test with.

Privilege Required: None

- The attacker does not need any privilege to exploit this vulnerability

User Interaction: Required

- Requires the victim to click on a malicious link and through the malicious code it can run potentially system commands in the victim's machine.

Scope: Unchanged

- The reason is that the impacted component is the same machine. The webserver and affected machines are the same component.

Impact Metrics:
Confidentiality impact: High

- Can be used to see the victims data or session token

Integrity Impact: None

- It does not change what is being seen in victims browser

Availability impact: None

- It does not affect any availability of any information or data. Maybe what the user sees.

Command Injection

Risk: High	**Impact**: Critical	**Likelihood**: low

CVSS SCORE: 7.1 **CWE:** 83.3 - Base Finding: 98 - Attack Surface: 1 - Environmental: 0.85 CWSS Calculator	**CVSS_String:** **CVSS-3.1:AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H** **CWE_String:** TI:C,1/AP:A,1/AL:S,0.9/IC:N,1/FC:T,1/RP:N,1/RL:A,1/AV:I,1/AS:NA,1/IN:NA,1/SC:A,1/BI:C,1/DI:L,0.2/EX:H,1/EC:N,1/P:C,0.8/

OWASP Top 10: Injection

Component(s): Not any particular site, the payload can be replaced with the whole HTTP request before it is sent to the web server.

Description: With this vulnerability the threat actor can execute system commands in the web server system.

Impact: with this vulnerability is critical because with the attacker can run system commands and gain full access to the system and the machine

Likelihood: However, the way this command is found is very unlikely, it requires the attacker to send just the payload instead of the whole HTTP request.

ASVS (the requirements that might not be met):

- Level 1:
    - V1.2 Security Architecture (Level 1): Command Injection may indicate weaknesses in the security architecture, allowing unauthorized execution of commands. Failing to prevent Command Injection could result in non-compliance with this

requirement.

- V2.1 Data Classification (Level 1): Command Injection could lead to unauthorized access or modification of sensitive data, violating data classification requirements.
- V9.1 Security Feature Flags (Level 1): Configuration weaknesses leading to LFI may affect the efficacy of security feature flags.

CVSS Options:

Exploitability Metrics:

Attack Vector: Network

- The attacker just needs internet access and can easily access the web server and exploit vulnerability over the internet.

Attack Complexity: High

- The attack payload is not out there. There is output from the command that is being run that is shown to the client.

Privilege Required: None

- The attacker does not need any privilege to exploit this vulnerability

User Interaction: None

- The reason is that this is a server side vulnerability and does not require user interaction

Scope: Unchanged

- The reason is that the impacted component is the same machine. The webserver and affected machines are the same component.

Impact Metrics:

Confidentiality impact: High

- You can see all the file in the machine and it’s content

Integrity Impact: High

- You can change the data in all the files in the system. And changes passwords

Availability impact: High

- You have full access to the machine with root privilege and you can make information or files or restrict authorized users from seeing information in the system.