

Sri Lanka Institute of Information Technology



Year 2 semester 2

IT23360600

G. P. I. Perera

SLIIT KANDY UNI

BUG BOUNTY REPORT 8

Web Security – IE2062

B.Sc. (Hons) in information Technology Specializing in Cyber Security

1. Requirement gathering and analysis

Selected sub domain	render.com
Hackerone URL	https://hackerone.com/render/
IP address	216.24.57.1

Subdomain list

```
(Lynx@vbox)-[~]
$ subfinder -d render.com

projectdiscovery.io

[INF] Current subfinder version v2.6.0 (outdated)
[INF] Loading provider config from /home/lynx/.config/subfinder/provider-config.yaml
[INF] Enumerating subdomains for render.com
aws-us-west-2-4-postgres.render.com
bid468w.21532664m.render.com
haha-15z1.app.render.com
replica-cyan.oregon-postgres.render.com
replica-cyan.aws-ap-southeast-1-1-postgres.staging.render.com
singapore-keyvalue.staging.render.com
gcp-us-west1-1-proxy-f.render.com
h48-fork.app.render.com
aws-us-east-2-1-postgres.render.com
ohio-postgres.render.com
community.render.com
lmnl-life.app.render.com
reo.render.com
render.app.render.com
render-test.app.render.com
aws-eu-central-1-1-proxy.staging.render.com
aws-us-west-2-3-redis.render.com
image-registry.aws-us-west-2-2.internal.staging.render.com
coreysery-io.app.render.com
pilot-web.app.render.com
test.app.render.com
aws-ap-southeast-1-1-postgres.render.com
simr.app.render.com
slab.app.render.com
scim.render.com
aws-us-east-2-1-redis.staging.render.com
temporal.render.com
e2e-test-server-vjo9.app.render.com
example.app.render.com
gqlgen-fork.app.render.com
home.render.com
posthog.render.com
redis-oregon.localhost.render.com
replica-cyan.aws-us-west-2-1-postgres.render.com
aws-us-west-2-1-redis.render.com
aws-us-east-1-1-redis.staging.render.com
image-registry.gcp-europe-west3-1.internal.staging.render.com
```

Firewall detection:

```
(lynx@vbox)-[~]
$ wafw00f https://render.com
  Application Error Disclosure
  Content Security Policy (CSP) Header Not Set (427)
  Cross-Domain Misconfiguration
  Big Redirect Detected (Potential Sensitive Information Leak)
  Cross-Domain JavaScript Source File Inclusion (4)
  Information Disclosure - Debug Error Messages (4)
  Private Information Disclosure (2)
  Strict-Transport-Security Header Not Set (71)
  Timing Attack Disclosure - Unix (3)
  X-Content-Type-Options Header Missing (680)
  ~ WAFW00F : v2.3.1 ~
  The Web Application Firewall Fingerprinting Toolkit
  Information Disclosure - Suspicious Comments (569)
[*] Checking https://render.com (28)
[+] The site https://render.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
  Retrieved from cache (72)
```

Nmap scan:

```
(lynx@vbox)-[~]
$ nmap render.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-27 22:16 EDT
Nmap scan report for render.com (216.24.57.1)
Host is up (0.11s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https
8080/tcp   open  http-proxy
8443/tcp   open  https-alt
Nmap done: 1 IP address (1 host up) scanned in 27.91 seconds
```

Active scan result from OWASP ZAP:

The screenshot displays the Burp Suite interface during an automated scan. The left sidebar shows the 'Sites' tab with 'Contexts' and 'Default Context'. The main panel is titled 'Automated Scan' and contains the following elements:

- URL to attack:**
- Use traditional spider:** ☒
- Use ajax spider:** with
- Buttons:**
- Progress:** Using traditional spider to discover the content

The bottom sidebar shows the 'Alerts' tab with a list of 15 alerts. The bottom status bar shows 'Alerts 0 3 7 5 Main Proxy: localhost:8080'.

The screenshot shows the Burp Suite interface with the Alerts tab active. The Alerts list on the left contains 15 items, with 'Application Error Disclosure' selected. The details pane on the right shows the following information:

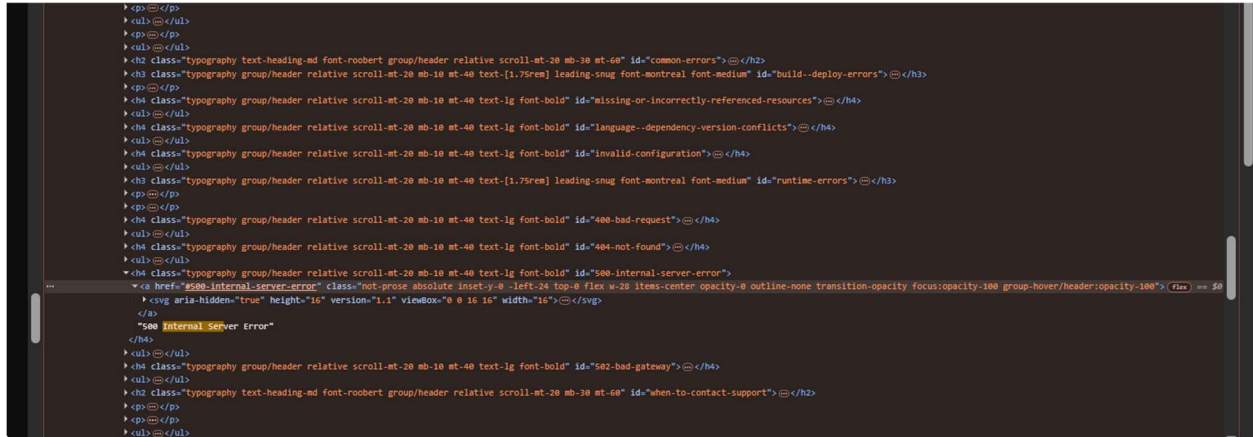
- URL:** <https://render.com/docs/troubleshooting-deploys>
- Risk:** Medium
- Confidence:** Medium
- Parameter:** Medium
- Attack:**
- Evidence:** Internal Server Error
- CWE ID:** 550
- WASC ID:** 13
- Source:** Passive (90022 - Application Error Disclosure)
- Input Vector:**
- Description:** This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.
- Other Info:**

The bottom status bar shows 'Alerts' with 15 items, '3' main proxy, 'localhost:8080', and 'Current Status' with various icons.

Nikto scan:

[illegible]

Developer tools:



2. Report Details

1. Vulnerability Title: Application Error Disclosure - Generic 500 with no details

2. Vulnerability Description:

The application discloses internal error information when unexpected input is provided, resulting in a 500 Internal Server Error.

While the generic error page does not visibly show details to the user, upon inspecting the raw HTML and server responses, hidden error messages are exposed.

These messages reveal backend technologies and internal application structure, potentially aiding further attacks.

Impact:

- Reveals internal file paths and server-side code structures.
- Disclosed backend technologies (e.g., PHP, MySQL, Apache).
- Increases the attack surface for further exploits such as:
 - SQL Injection
 - Local File Inclusion (LFI)
 - Remote Code Execution (RCE)
- Provides an attacker with insight into the system's inner workings, making targeted attacks easier.

4.Affected Components:

1. <https://render.com/docs/troubleshooting-deploys>

5.Impact Assessment:

OWASP analysis:

Risk level	Meduim
Confidence	Medium

6. Steps to reproduce –

- **On owasp zap –**
Start the application, input target URL and run an automated scan.
Observe alerts.
- **Developer tools –**
Right-click anywhere on the page → **Inspect** (or press Ctrl + Shift + I or F12). This opens **Developer Tools**. Search and see the information disclosure in the script.


7. Proposed mitigation or fix

- Implement proper error handling and return **generic error messages** to users.
- Configure the server and application to disable detailed error reporting in production environments.
- Sanitize and validate all user input to prevent application crashes.
- Monitor and log detailed errors internally, but do not expose them to users.

8. Submission


#3115493

☆

Application error disclosure 

[ADD HACKER SUMMARY](#)

TIMELINE · EXPORT



lynx_jr2002 submitted a report to [Render](#). [\(Edit information\)](#)

a few seconds ago


The application discloses internal error information when unexpected input is provided, resulting in a 500 Internal Server Error. While the generic error page does not visibly show details to the user, upon inspecting the raw HTML and server responses, hidden error messages are exposed.

These messages reveal backend technologies and internal application structure, potentially aiding further attacks.


- On owasp zap –
Start the application, input target URL and run an automated scan.
Observe alerts.
- Developer tools –
Right-click anywhere on the page → Inspect (or press Ctrl + Shift + I or F12). This opens Developer Tools. Search and see the information disclosure in the script.

Impact

- Reveals internal file paths and server-side code structures.
- Disclosed backend technologies (e.g., PHP, MySQL, Apache).
- Increases the attack surface for further exploits such as:
 - SQL Injection
 - Local File Inclusion (LFI)
 - Remote Code Execution (RCE)
- Provides an attacker with insight into the system's inner workings, making targeted attacks easier.

 1 attachment

F4293081: [Screenshot_2025-04-28_074700.png](#)



Add comment

Request Mediation

Leave a comment to all participants