

Forum:[Hack Forums](#)

Thread: [Exploiting and Securing Remote Code Execution Vulnerabilities: A Practical Ethical Ha](#)

JoseMar - January 11, 2025, 11:17 pm

Exploiting and Securing Remote Code Execution Vulnerabilities: A Practical Ethical Hacking Guide

RCE, or Remote Code Execution, is a vulnerability that allows an attacker to run arbitrary code on a remote system. This typically occurs when an application fails to properly validate user input, allowing malicious code to execute. RCE is one of the most critical vulnerabilities because it can grant attackers full control over a target system.

How RCE Works

Input Injection: The attacker sends malicious input to an application, often through web forms, APIs, or uploaded files.

Execution Context: The input is processed unsafely, often directly passed to system commands, libraries, or interpreters like Python or PHP.

Code Execution: The system executes the injected commands with the privileges of the application.

Examples of RCE Vulnerabilities

1: Command Injection

Code

```
import os
```

```
user_input = input("Enter filename: ")
```

```
os.system(f"cat {user_input}")
```

If a user enters filename; rm -rf /, the system executes both cat filename and the destructive command rm -rf /.

2: Deserialization Vulnerabilities Applications that deserialize untrusted data can allow attackers to include malicious serialized objects. For example, in Python:

Code

```
import pickle
```

```
data = input("Enter serialized data: ")
```

```
obj = pickle.loads(data)
```

A malicious serialized object could execute arbitrary commands.

3: Web Application RCE A web app with insecure file upload may allow attackers to upload a malicious PHP file:

Code

```
<?php
```

```
echo shell_exec($_GET['cmd']);
```

?>

The attacker can then run commands by visiting `example.com/upload/malicious.php?cmd=ls`

Preventing RCE

Input Validation: Sanitize and validate user inputs to ensure only safe data is processed.

Least Privilege: Run applications with minimal privileges to limit the impact of an RCE.

Escape Shell Commands: Use safe methods like `subprocess.run` with a list of arguments instead of string concatenation in Python.

Use Secure Libraries: Avoid unsafe functions like `eval()` and `exec()`

Ethical Practice

When practicing RCE exploitation in your labs:

Always ensure the target systems are explicitly set up for testing.

Follow legal and ethical guidelines.

Focus on learning how to secure systems against RCE after exploiting them.

Example: Command Injection in a Python Web Application

Imagine a web application where users can input a domain name to check its DNS records. Here's an insecure implementation:

Vulnerable Code

```
Code from flask import Flask, request
import os
```

```
app = Flask(__name__)
```

```
@app.route('/dns_lookup', methods=['GET'])
```

```
def dns_lookup():
```

```
    domain = request.args.get('domain')
```

```
    # Vulnerable: User input is directly used in a system command
```

```
    result = os.popen(f"nslookup {domain}").read()
```

```
    return f"<pre>{result}</pre>"
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True)
```

Exploiting the Vulnerability

Normal InputURL: `http://localhost:5000/dns_lookup?domain=example.com`

Result: The DNS records for example.com are returned.

Malicious InputURL: `http://localhost:5000/dns_lookup?domain=example.com; cat /etc/passwd`

Result: The application executes both `nslookup example.com` and `cat /etc/passwd`. The attacker sees the system's `/etc/passwd` file, which contains sensitive information.

Securing the Application

Sanitize Input Use input validation to ensure only valid domain names are accepted. For example:

Code import re

```
def is_valid_domain(domain):
    pattern = r"^[a-zA-Z0-9.-]+$"
    return bool(re.match(pattern, domain))
```

Avoid System Commands Instead of using os.popen or os.system, use libraries that interact directly with DNS servers

Code import dns.resolver

```
@app.route('/dns_lookup', methods=['GET'])
def dns_lookup():
    domain = request.args.get('domain')
    if not is_valid_domain(domain):
        return "Invalid domain!", 400
    try:
        answers = dns.resolver.resolve(domain, 'A')
        result = "\n".join([str(ip) for ip in answers])
        return f"<pre>{result}</pre>"
    except Exception as e:
        return f"Error: {str(e)}", 500
```

Run with Least Privileges Configure the web server to run as a non-root user, limiting the damage if an RCE is exploited.

Ethical Lab Exercise

Setup

Create the vulnerable web app.

Run it in an isolated environment like Docker or a VM.

Exploit

Access the /dns_lookup endpoint.

Test various payloads, such as:

example.com; ls

example.com; cat /etc/passwd

Patch

Apply the fixes above.

Verify that malicious inputs no longer execute system commands.

Report

Document your findings, including:

The vulnerable code.

Exploitation steps.

How you secured the application.

Note: The example provided focuses on command injection, which is just one type of RCE vulnerability. Remote Code Execution can occur in various contexts, including:

Deserialization Vulnerabilities: Exploiting poorly implemented object deserialization to execute arbitrary code.

Insecure File Uploads: Uploading malicious scripts or executables that the server executes.

Web Shells: Exploiting RCE to plant persistent backdoors on the server.

Library or Dependency Exploits: Leveraging vulnerabilities in third-party software to execute code.

Each type of RCE has unique attack vectors and prevention techniques, making it important to understand the broader scope of RCE vulnerabilities in ethical hacking.

7 - January 11, 2025, 11:20 pm

Excellent tutorial. Exactly what this forum need IMO

Amazing job

Nightmare - January 12, 2025, 3:34 am

Nice my brother,keep it up!!Love your tutorials!???

JoseMar - January 15, 2025, 2:00 pm

(Jan 12, 2025 12:20 AM)7 Wrote: [\[CITE\]](#) Excellent tutorial. Exactly what this forum need IMO
Amazing job

(Jan 12, 2025 04:34 AM)-Nightingale- Wrote: [\[CITE\]](#) Nice my brother,keep it up!!Love your tutorials!???

Appreciate it guys! ?