

Penetration Testing

BackStory:

Below is a virtual machine (VM) from an incorporation that runs a critical service for the organization. This service handles cryptocurrency transactions, manages blockchain security, and stores confidential client data. The company prides itself on its cutting-edge security infrastructure, but recent security audits have raised concerns about potential vulnerabilities in its web applications and backend services.

Unbeknownst to this Incorporation., misconfigurations and an insecure web application have introduced security loopholes into the system. A vulnerability in the web application allows **Remote Code Execution (RCE)**, potentially exposing sensitive data and allowing attackers to gain unauthorized access to the system. Additionally, weak credentials and misconfigured services have created further openings for exploitation.

As a security researcher, your task is to assess the security of this incorporation server, identify its vulnerabilities, and attempt to exploit them to retrieve the hidden security flag:

`halborn{flag}`.

Objective

This lab aims to introduce students to remote Linux enumeration, vulnerability analysis, and exploitation techniques. Students will explore a given Ubuntu virtual machine (VM) from a separate attacking VM, identify open ports, discover vulnerabilities within the logic of a web application, and gain access to the web application.

Prerequisites

- Basic knowledge of the Linux command line
- Familiarity with networking concepts (e.g., SSH, ports, scanning)

- Understanding of common security vulnerabilities (e.g., misconfigured services, weak credentials, session hijack)
- Experience with penetration testing tools (e.g., Nmap, netdiscover, Burpsuite etc)

Lab Setup

1. Download and set up two VMs:
 - [Target VM \(Ubuntu\) with a vulnerable web application](#)
 - **Attacking VM (Kali Linux or another penetration testing distribution)**
2. Ensure both VMs are on the same network.
3. Identify the target VM's IP address.
4. Open a terminal on the attacking VM to begin the challenge.

The target VM contains exploitable and running services. You are expected to run the tools above to access the web application and find a flag in the exploited VM.

CHECKPOINTS AND TASKS

1: Hijack a Session to Gain Access to the Web Application


A. Enumerate the Target VM

- Identify the Target VM's IP address
- Scan for open ports and running services
- Analyze the result and determine potential attack vectors

B. Analyze the Web Application

- Identify the web application running on the target VM
- Obtain the source code to identify a logic flaw

Index of /

Name	Last modified	Size	Description
 cryptos.zip	2023-09-13 11:42	27K	

Apache/2.4.52 (Ubuntu) Server at 192.168.178.210 Port 80

Figure 1: Index Page of Web Server

- Inspect the source code to identify a potential implementation flaw
- A checkpoint to this can be found in the screenshot below

```
import time
import secrets
from flask import Flask, session, redirect, jsonify, request, make_response
from utils.db_connection import Connector
from utils.renderers import render_template
import jwt
from waitress import serve

app = Flask(__name__)
conn = Connector()
cursor = conn.initMySQL()
token_jwt = secrets.token_bytes(32)

@app.before_request
def checkBefore():
    global cursor
    global session
    cursor = conn.initMySQL()
    if session and "time" in session and time.time() - session['time'] > 7000: #2-hours
        session.pop("time", None)
    return redirect("/")
```

Figure 2: run.py

```
from random import choice

def token_hex(value):
    alphabet = 'abcdef0123456789'
    return ''.join(choice(alphabet) for _ in range(5))

def token_bytes(value):
    alphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
    return ''.join(choice(alphabet) for _ in range(value)).encode()
```

Figure 3: Secrets.py

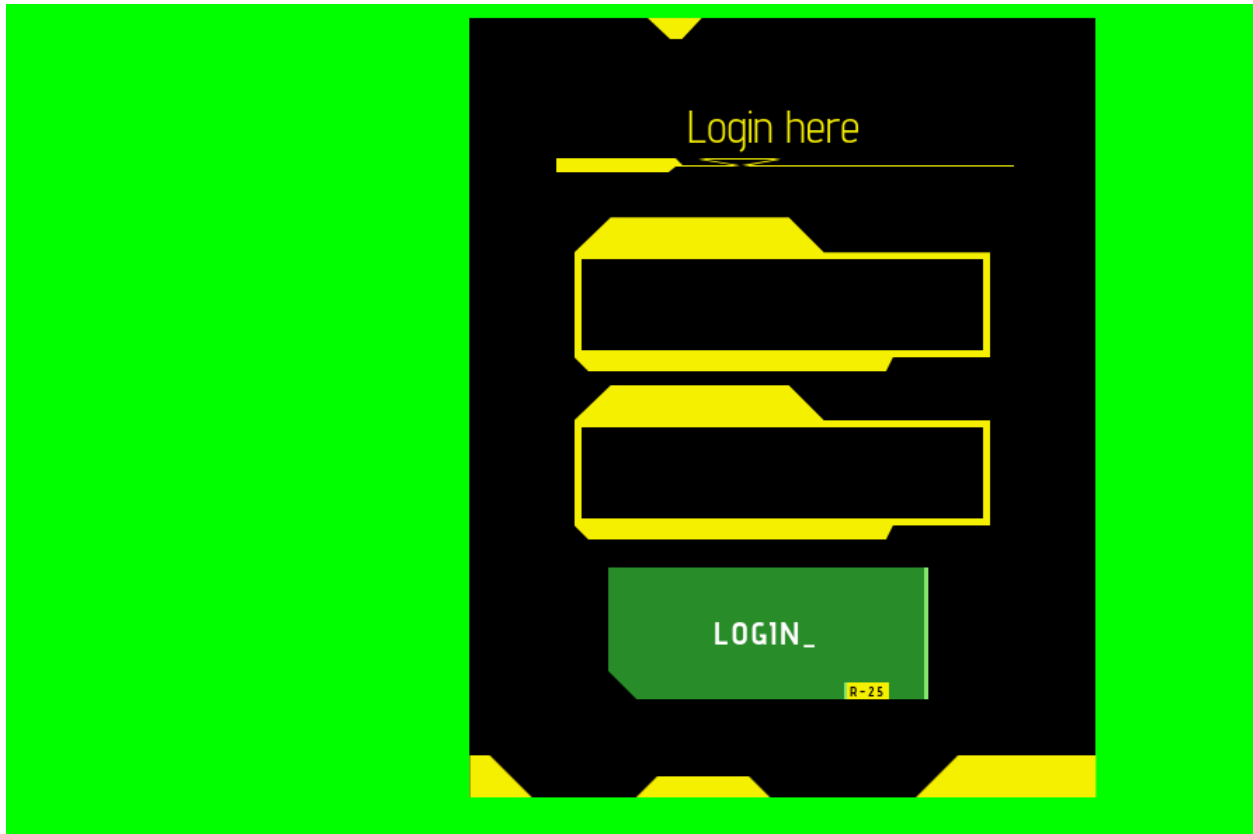


Figure 4: Login Page of Web Application

C. Use Exploitation Tools to Gain Access to the Web Application

- After identifying the Implementation, check endpoints
- Use various exploitation Tools to gain access

D. Verification of Gaining Access

- Gaining Access implies you can access other endpoints of the web application
- A checkpoint is shown in the snapshot below

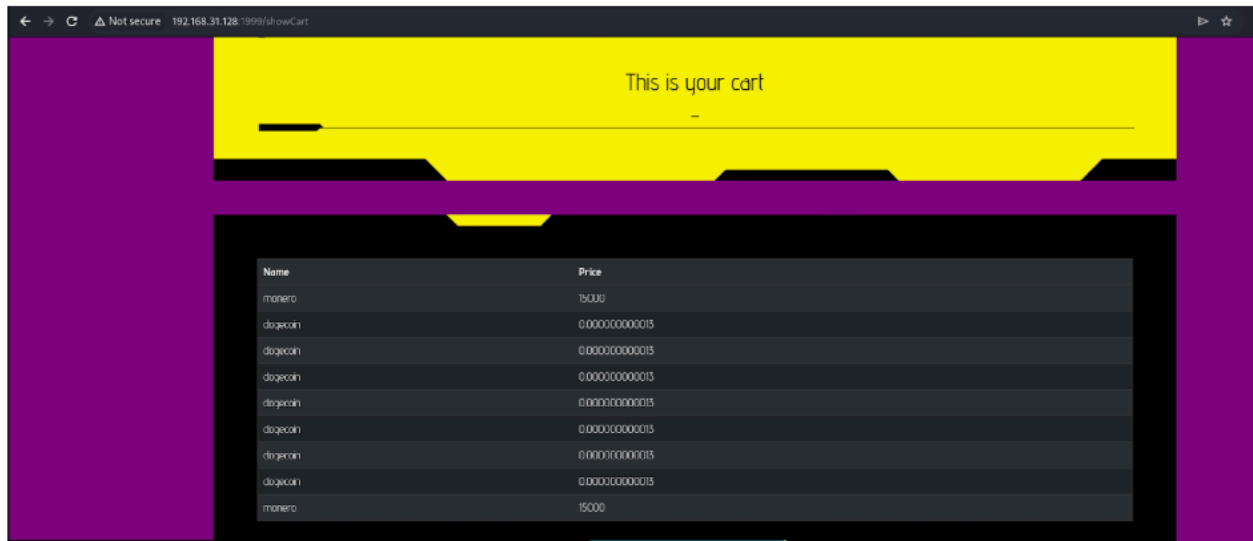


Figure 5: Access to showCart Endpoint

Bonus Task

2: Privilege Escalation to find Flag

In the Ubuntu VM, other services running are vulnerable to web applications with vulnerabilities that can be exploited, including **Remote Code Execution (RCE)**.

Using your understanding of exploiting, obtain **two (2)** flags in the format `halborn{flag}`