Cyber Security Internship

(TryHackMe OWASP Top 10)

Harsh M Parikh

ph400764@gmail.com

A01 — Broken Access Control

Definition: Improper enforcement of user permissions allowing privilege escalation or unauthorized access.

Impact: Data leakage, privilege escalation, account takeover.

Lab / Practical notes

- Lab objective: identify endpoints that return data for other user IDs; attempt forced browsing to admin endpoints.
- Example reconnaissance commands (lab-only):
    - Enumerate endpoints in the app and test parameter tampering.
    - Use /users/{id} endpoints to check access control enforcement.

How it can be exploited:

- Modify a user id parameter to access another user's data (in lab).
- Use an authenticated low-privilege account and access an admin URL by changing path or ID.

Mitigation

- Enforce server-side authorization checks for every protected resource.
- Use a central access control library, avoid relying on client-side controls.
- Implement least privilege and role-based access checks on every request.

TryHackMe-style lab summary

- Steps: enumerate endpoints → login as normal user → change resource identifiers → observe response codes / content differences → document.
- Deliverable evidence: request/response snippets (redacted), explanation of check and fix.

A02 — Cryptographic Failures

Definition: Weak or misconfigured cryptography (e.g., weak algorithms, no TLS, poor key management).

ph400764@gmail.com

Impact: Data exposure, broken confidentiality, credential theft.

Lab / Practical notes

- Check for:
  - Use of HTTP rather than HTTPS.
  - Deprecated algorithms (MD5, SHA1) or weak ciphers.
  - Sensitive data stored in plaintext.

Mitigation

- Use TLS 1.2/1.3 only; enforce HSTS.
- Use well-vetted libraries and strong algorithms (AES-GCM, RSA-2048+/ECDHE).
- Proper key lifecycle and secret management (vaults, environment variables).

A03 — Injection (SQL, NoSQL, OS, LDAP)

Definition: Unsanitized input interpreted by an interpreter — leads to unauthorized commands or data.

Impact: Data exfiltration, remote command execution, bypass auth.

Lab / Practical notes

- Lab objective: demonstrate how user input can change backend queries (lab-only, authorized).
- Testing approach (lab-only): parameterize inputs, observe application errors, identify unsanitized fields.

Mitigation

- Use parameterized queries / prepared statements.
- Strong input validation and ORM-safe patterns.
- Least privilege DB users.

A04 — Insecure Design

Definition: Flaws introduced during design phase (missing threat modeling and secure defaults).

Impact: Broad — from authentication flaws to insecure data flows.

ph400764@gmail.com

Mitigation

- Threat modeling in design phase, adopt secure design patterns, default-deny.

A05 — Security Misconfiguration

Definition: Misconfigured servers, cloud services, frameworks, or default credentials left enabled.

Impact: Account takeover, data leakage, pivoting.

Lab / Practical notes

- Check publicly accessible services, directory listings, default pages, and permissive CORS.

Mitigation

- Harden images, remove default accounts, secure cloud buckets and services.

A06 — Vulnerable and Outdated Components

Definition: Use of components with known vulnerabilities (libraries, frameworks) without updates.

Impact: Remote code execution, data leaks.

Mitigation

- Maintain SBOM, use dependency scanners (OSS tools), run regular updates and patches.

A07 — Identification and Authentication Failures

Definition: Weak authentication primitives, broken session management.

Impact: Account takeover, privilege escalation.

Mitigation

- Implement MFA, protect session tokens (secure, HttpOnly), rotate tokens on sensitive operations.

ph400764@gmail.com

A08 — Software and Data Integrity Failures

Definition: Trusting unverified sources — e.g., CI/CD artifacts, unsigned packages.

Impact: Supply-chain compromise.

Mitigation

- Verify signatures, use reproducible builds, lock down build pipelines.

A09 — Security Logging and Monitoring Failures

Definition: Lack of sufficient logging, or missing alerting for suspicious activity.

Impact: Delayed detection of breaches.

Mitigation

- Centralized logs, alert on anomalies, retention policies, and incident response plans.

A10 — Server-Side Request Forgery (SSRF)

Definition: Server-side code fetches remote resources without validating URLs, allowing arbitrary server requests.

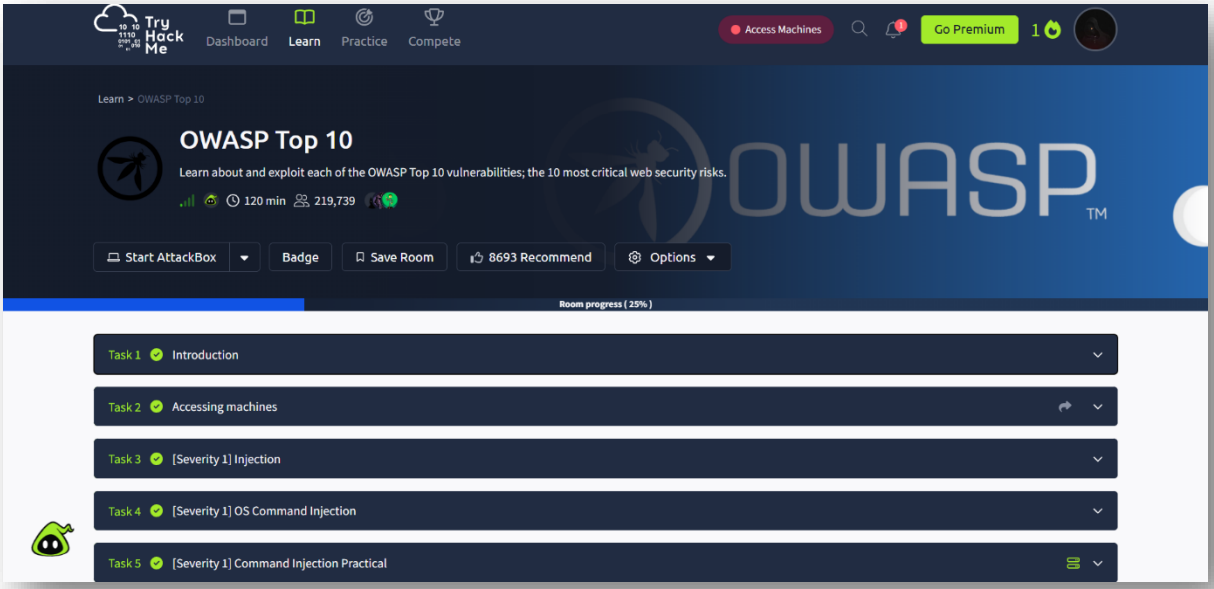
Impact: Access to sensitive internal services.

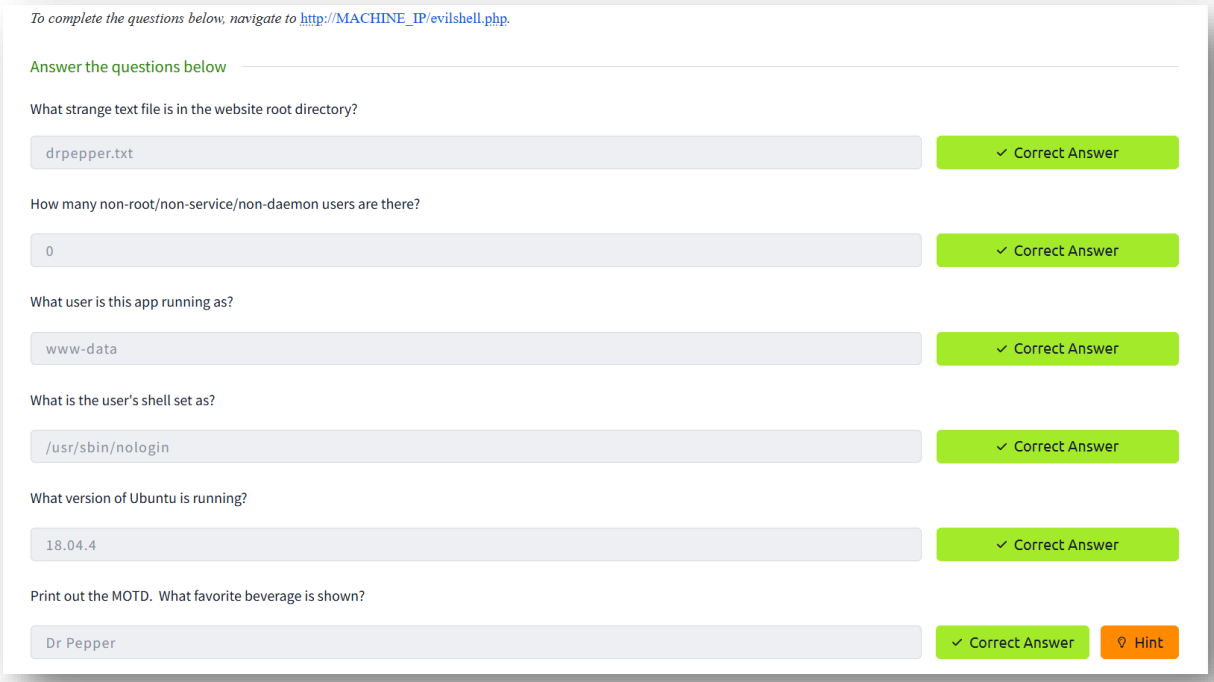
Mitigation

- Whitelist external endpoints, restrict outbound access, validate and sanitize URLs.

ph400764@gmail.com

Screenshot :


Try Hack Me
Dashboard
Learn
Practice
Compete
Access Machines
Go Premium
Learn > OWASP Top 10
OWASP Top 10
Learn about and exploit each of the OWASP Top 10 vulnerabilities; the 10 most critical web security risks.
120 min
219,739
OWASP
TM
Start AttackBox
Badge
Save Room
8693 Recommend
Options
Room progress (25%)
Task 1 Introduction
Task 2 Accessing machines
Task 3 [Severity 1] Injection
Task 4 [Severity 1] OS Command Injection
Task 5 [Severity 1] Command Injection Practical


To complete the questions below, navigate to http://MACHINE_IP/evilshell.php.
Answer the questions below
What strange text file is in the website root directory?
drpepper.txt
Correct Answer
How many non-root/non-service/non-daemon users are there?
0
Correct Answer
What user is this app running as?
www-data
Correct Answer
What is the user's shell set as?
/usr/sbin/nologin
Correct Answer
What version of Ubuntu is running?
18.04.4
Correct Answer
Print out the MOTD. What favorite beverage is shown?
Dr Pepper
Correct Answer
Hint

Conclusion:

Through the OWASP Top 10 labs, I strengthened my understanding of modern web vulnerabilities and hands-on exploitation techniques. Each lab enhanced my practical security testing and mitigation skills.

ph400764@gmail.com