Penetration Testing Project — UTM Parrot OS (mayday.org)

Summary: This report documents a safe, reproducible penetration testing workflow using Parrot OS running in UTM (IP `192.168.64.2`). All active exploitation was performed **only** in an isolated lab (Metasploitable / Juice Shop). All checks against `mayday.org` were passive/non-intrusive. Screenshots will be pasted at the end of this document in the **Screenshots** section.

Table of Contents

---

1. Environment Setup & Verification

Goal: Demonstrate Parrot OS in UTM is installed, networked at `192.168.64.2`, and essential tools (Nmap, Wireshark, Metasploit) are present and functional.

Steps performed & evidence placeholders

1. UTM VM settings: verified adapter type and VM snapshot strategy.
   Command / UI action: Open UTM → Select Parrot VM → Settings → Network.
   Report explanation: VM configured using the adapter (chosen for isolation). A snapshot `pre-testing-YYYYMMDD`was created to allow rollback.

2. Parrot desktop & terminal: booted VM, opened terminal.
   Command / UI action: Start VM → open Terminal.
   Report explanation: Parrot OS booted normally; terminal available for testing.
3. Confirm IP address: `ip a` (or `ifconfig`).
   Command:

ip a

Report explanation: VM network interface shows IPv4 `192.168.64.2`, confirming connectivity within the lab subnet.

4. Verify tool availability (version checks):

nmap -v | head -n 3
wireshark --version
msfconsole --version

Report explanation: Nmap, Wireshark, and Metasploit responded to version queries indicating they are installed and runnable.

---

2. Lab Isolation & Snapshot (Containment)

Goal: Demonstrate snapshot created and explain network isolation choices.

Steps performed & evidence placeholders

1. Created snapshot `pre-testing-YYYYMMDD` in UTM.
   Report explanation: Snapshot created prior to any active tests to allow immediate rollback and containment.
2. Network isolation rationale: host-only / internal recommended for exploitation lab.
   Report explanation: Host-only/internal networks prevent accidental exposure of vulnerable VMs to the internet while allowing Parrot ↔ lab VM communication.

---

3. Passive Reconnaissance (mayday.org)

Goal: Gather public information about mayday.org using OSINT and passive tools only.

Commands run & concise findings

1. theHarvester (email/subdomains):

theharvester -d mayday.org -b bing -l 200

Explanation: Collected publicly indexed emails, subdomains, and hostnames. Paste the theHarvester output screenshot below.

2. Whois (registrar & registration data):

whois mayday.org | sed -n '1,40p'

Explanation: Captured registrar, registration/expiry dates, and name servers.

3. DNS (A/MX/NS records):

dig +short mayday.org any

dig +short MX mayday.org

Explanation: Retrieved authoritative IPs and mail exchangers (if publicly listed). If the site uses a CDN/WAF, results likely show CDN IPs (e.g., Cloudflare, Akamai).

4. Online mapping (DNSDumpster or similar):
   UI action: Run a DNSDumpster lookup for mayday.org and capture screenshot.
   Explanation: Visual map helps identify subdomains, hosting, and potential third-party services.

Security note: All actions above query publicly available information only and are non-intrusive.

---

4. Local Network Mapping (192.168.64.0/24)

Goal: Discover active hosts and identify lab machines (Metasploitable / Juice Shop).

Commands & observations

1. Ping sweep / ARP scan:

nmap -sn 192.168.64.0/24 -oN nmap_host_discovery_192.168.64.0_24.txt

Explanation: Identified active hosts. The Parrot VM (`192.168.64.2`) and lab VM (e.g., `192.168.64.3`) are listed. Saved raw output to file for appendix.

2. Note: If any unexpected hosts appear on the subnet, document them and investigate separately before proceeding with attacks.

---

5. Scanning & Enumeration

Important: For `mayday.org` we performed **only low-intensity, passive checks** (banner/cert). All intrusive port/UDP scans and enumeration were executed against the isolated lab VM only (e.g., `192.168.64.3`).

5A — Safe checks against mayday.org

1. Low-intensity service/version check (top 100 ports):

nmap -Pn --top-ports 100 --open --reason -sV --version-intensity 0 mayday.org -oN mayday_top100_$(date +%F).txt

Explanation: Limited probes to reduce risk of triggering protective measures; useful to see if any public service banners are exposed.

2. TLS/cert enumeration (HTTPS):

nmap -p 443 --script ssl-cert,ssl-enum-ciphers mayday.org -oN mayday_ssl_$(date +%F).txt

Explanation: Provides certificate issuer, expiry date, and cipher support—useful passive findings.

5B — Active enumeration on lab VM (192.168.64.3)

1. Full TCP port scan (lab only):

nmap -sS -p- -T4 -A 192.168.64.3 -oN lab_nmap_tcp_full_192.168.64.3.txt

Explanation: Discovers all open TCP ports and attempts service detection and OS fingerprinting.

2. UDP sample scan (lab only):

sudo nmap -sU -p 53,67,68,123,161 192.168.64.3 -oN lab_nmap_udp_192.168.64.3.txt

Explanation: UDP services can reveal additional attack surfaces; scanning is slower and noisier—kept local in lab.

3. Service enumeration examples (SSH, HTTP):

```
nmap -sV -p 22 --script ssh2-enum-algos 192.168.64.3
nmap -sV -p 80 --script http-title,http-server-header 192.168.64.3
```

Explanation: Banner info and HTTP headers give valuable version data for vulnerability identification.

---

6. Vulnerability Scanning (Nessus) — Lab Only

Goal: Run Nessus Essentials against local lab VM and collect evidence of findings.

Workflow & evidence

1. Install Nessus and access the web UI at `https://localhost:8834/`.
2. Create Basic Network Scan targeting `192.168.64.3`.
3. Run the scan, wait for completion, and review results.

What to include in report: screenshot of Nessus dashboard, PDF export of the report (top findings). For three chosen vulnerabilities, include plugin output, CVE references, and suggested remediation steps.

---

7. Vulnerability Analysis (3 selected findings)

For each selected vulnerability include: Name, Source (Nessus/Nmap), Risk Rating, Affected Service/Port, 1-sentence Description, Evidence (screenshot filename), False-Positive checks, and Mitigation.

Example template (repeat for 3 vulns):

- **Name:** vsftpd v2.3.4 backdoor
- **Source:** Nmap / Nessus
- **Risk:** High
- **Service/Port:** FTP / 21

- **Description:** Backdoor in specific vsftpd builds allows remote code execution.
- **Evidence:** `6_nessus_vsftpd_v234_backdoor.png` (screenshot of Nessus plugin output)
- **False-positive analysis:** Verified by running safe banner checks and confirming server version via `nmap -sV` and `ftp` banner.
- **Mitigation:** Upgrade vsftpd to patched version or disable FTP and use SFTP/SSH.

(Repeat for two more findings — choose ones from Nessus results such as outdated Apache, weak ciphers, or default credentials.)

---

8. Basic Exploitation (Lab Only) — vsftpd demo

Important: This entire section was performed only against the isolated Metasploitable lab VM. Never exploit public systems without explicit permission.

Commands & actions

1. Start Metasploit: `msfconsole`
   Evidence: screenshot of msfconsole prompt `msf >`.
2. Search & select exploit:

search vsftpd
use exploit/unix/ftp/vsftpd_234_backdoor
show options
set RHOST 192.168.64.3
set RPORT 21

Explanation: Configures exploit to target lab VM.

3. Run exploit:

run

Evidence & explanation: Meterpreter shell or reverse shell displayed. Run `id` or `sysinfo` and capture minimal evidence. Immediately stop further actions and revert VM to snapshot.

4. Cleanup: revert snapshot `pre-testing-YYYYMMDD` via UTM to remove artifacts.

---

9. Web Application Testing

Goal: Passive checks against mayday.org; active testing only on local Juice Shop or DVWA.

Passive scan — mayday.org (ZAP passive only)

1. OWASP ZAP: Open ZAP → Enter `https://mayday.org` in the Sites pane → run Passive Scan.
   Evidence: screenshot of passive alerts (security headers, cookies flags, robots.txt findings).
   Explanation: Passive scan identifies configuration issues without sending intrusive payloads.

Active scan — local Juice Shop (example)

1. Launch Juice Shop:

docker run -d -p 3000:3000 bkimminich/juice-shop

2. Open `http://localhost:3000` and confirm app is reachable (screenshot).
3. In ZAP, run active scan against `http://localhost:3000` and capture alerts.

Findings to document: for each vulnerability (e.g., reflected XSS, SQLi) include the request, payload, response screenshot, and remediation.

---

10. Evidence packaging & screenshot list

How to paste screenshots: Paste each screenshot image file below the appropriate caption so reviewers can match them to the steps. Name files using the convention listed.

Screenshot filenames (paste in this order)

1. `1_UTM_settings.png` — UTM VM list + Network settings
2. `1_Parrot_desktop_terminal.png` — Parrot desktop with terminal
3. `1_Parrot_ip_192.168.64.2.png` — `ip a` showing 192.168.64.2
4. `1_tool_versions_nmap_wireshark_msf.png` — tool version outputs
5. `2_snapshot_pre-testing-YYYYMMDD.png` — snapshot created in UTM
6. `3_theHarvester_mayday.png` — theHarvester output

7. `3_whois_dig_mayday.png` — whois and dig outputs
8. `3_dnsdumpster_mayday.png` — DNSDumpster output (if used)
9. `4_nmap_host_discovery.png` — nmap -sn 192.168.64.0/24 output
10. `5_mayday_top100_scan.png` — low-intensity nmap output for mayday.org
11. `5_mayday_ssl.png` — nmap ssl-cert output for mayday.org
12. `5_lab_nmap_tcp_192.168.64.3.png` — full TCP nmap on lab VM
13. `5_lab_nmap_udp_192.168.64.3.png` — UDP scan output
14. `5_service_enum_ssh_http.png` — service enumeration command outputs
15. `6_nessus_dashboard.png` — Nessus dashboard showing scans
16. `6_nessus_top_findings.pdf` — Exported Nessus PDF (attach file)
17. `7_vuln1_nessus_vsftpd.png` — screen