VulnNet: dotpy — Industry-Style Penetration Testing Report

Client: TryHackMe (Training Environment)
Target: VulnNet: dotpy
Report Date: 21 January 2026
Testing Window: 13 December 2025 (single-day assessment; approx. 14:00–18:00 local time) **Version:** 1.3

Document Control

Version	Date	Author	Notes
1.1	21 January 2026	PREM	v1.1 — SSTI payload formatting + scope IP alignment + refined F-02 CVSS
1.2	21 January 2026	PREM	v1.2 — Added testing window, observed stack, evidence index
1.2.1	21 January 2026	PREM	v1.2.1 — Document control formatting + appendix spacing fix

---

1. Executive Summary

A penetration test was conducted against the **VulnNet: dotpy** target hosted in a training environment. The assessment identified a critical web application vulnerability allowing **Server-Side Template Injection (SSTI)** which was leveraged to achieve **remote code execution (RCE)** and obtain an initial shell on the system.

Post-exploitation activities revealed multiple privilege escalation paths, resulting in full **root-level compromise** of the host.

---

2. Scope

2.1 In Scope

- **Target Web Application:** `http://10.82.180.28:8080`
- **Target Host (Application Server):** `10.82.180.28`
- **Attacker Host (AttackBox/Kali):** `10.80.113.44`

- **Observed shell callback host:** `10.81.181.59`
- **Services Tested:**
  - TCP/8080 (HTTP – Werkzeug/Flask)

2.2 Out of Scope

- Denial of Service (DoS)
- Brute-force attacks against authentication
- Attacks outside the identified host

---

3. Methodology

The assessment followed a standard penetration testing workflow:

1. Reconnaissance and service discovery (Nmap)
2. Web content discovery (Gobuster)
3. Vulnerability verification (SSTI confirmation)
4. Exploitation (SSTI → RCE → reverse shell)
5. Post-exploitation enumeration
6. Privilege escalation (pip3 misuse + sudo SETENV Python execution)
7. Evidence collection and reporting

---

4. Risk Rating

Severity	Description
Critical	Full system compromise, remote unauthenticated exploitation, major impact
High	Privilege escalation or sensitive data exposure requiring some access
Medium	Exploitation possible with constraints; limited impact
Low	Minor information exposure or best-practice issues

---

5. Findings Summary

ID	Finding	Severity	Affected Component
F-01	Server-Side Template Injection (SSTI) → Remote Code Execution	Critical	Web App (Werkzeug/Flask over TCP/8080)
F-02	Sudo Misconfiguration: `pip3 install .` allowed as another user	High	Local privilege escalation to `system-adm`

ID	Finding	Severity	Affected Component
F-03	Sudo SETENV with Python script allows PYTHONPATH hijacking → Root	Critical	`/opt/backup.py` executed with sudo

---

6. Technical Findings

F-01: Server-Side Template Injection (SSTI) → Remote Code Execution

Severity: Critical
CWE: CWE-1336 (Improper Neutralization of Special Elements Used in a Template Engine)
CVSS v3.1: 9.8 (Critical)

CVSS Justification: Remote exploitable over HTTP with no authentication and no user interaction; leads to command execution and full compromise.
Vector: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`
Affected Service: HTTP (Werkzeug/Flask on TCP/8080)
Observed stack: Werkzeug/Flask (Python 3.6.9)
Attack Vector: Remote

Description

The target web application was vulnerable to **Server-Side Template Injection (SSTI)**. User-controlled input was evaluated by the server-side template engine, allowing execution of template expressions and ultimately enabling arbitrary command execution.

Evidence

Network scan (Nmap)

```
nmap -sV -sC -Pn 10.82.180.28 -oN services.txt
```

Evidence references

- Nmap service detection screenshot: `evidence/nmap/nmap_services.png`

Enumeration (Gobuster)

```
gobuster dir -u http://10.82.180.28:8080/ \
  -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt \
  -x php,js,bak,txt -b 403 -o gobuster.txt
```

Discovered paths (relevant):

- `/login` (200)
- `/register` (200)

Evidence references

- Gobuster enumeration screenshot: `evidence/web/gobuster.png`

SSTI confirmation Example payload used:

```
{{7+7}}
```

Evidence references

- SSTI confirmation (template evaluation result): `evidence/web/ssti_confirm_14.png`
- Burp request (baseline): `evidence/web/burp_baseline_request.png`
- Burp request (SSTI payload): `evidence/web/burp_ssti_payload_request.png`

Reverse shell (RCE) Listener on attacker:

```
nc -lvnp 4444
```

Evidence references

- Reverse shell received: `evidence/exploit/reverse_shell_received.png`

Impact

Successful exploitation provides attackers with:

- Remote command execution as the web service user
- Data theft and modification
- Full host compromise when chained with local privilege escalation

Remediation

- Avoid rendering user input directly into templates (e.g., do not pass untrusted input into `render_template_string`).
- Use strict allowlists and context-aware encoding.
- Consider sandboxed template rendering and disable dangerous template features.
- Implement a WAF as defense-in-depth (not as a primary control).

F-02: Sudo Misconfiguration — pip3 install allowed as another user

Severity: High
CWE: CWE-269 (Improper Privilege Management)
CVSS v3.1: 7.1 (High)

CVSS Justification: Requires local access (foothold) and allows escalation to a privileged local account (`system-adm`). Impact is high for confidentiality and integrity; availability impact is limited without chaining additional issues.
Vector: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L`
Affected Host: Linux
Attack Vector: Local (requires foothold)

Description

After gaining an initial foothold, the `web` user was able to leverage a sudo misconfiguration permitting execution of:

```
sudo -u system-adm /usr/bin/pip3 install .
```

This allows installation of a malicious Python package and execution of arbitrary code as the `system-adm` user.

Evidence

Evidence references

- setup.py downloaded: `evidence/priv-esc/setup_py_download.png`
- setup.py contents + pip install execution: `evidence/priv-esc/pip3_install_privesc.png`
- user flag obtained as system-adm: `evidence/flags/user_flag.png`

Impact

- Privilege escalation from `web` to `system-adm`
- Lateral movement and further compromise

Remediation

- Remove sudo permissions for package managers such as `pip3`.
- Restrict sudo rules to explicit binaries and fixed arguments.
- Require password for privileged operations.
- Use Python virtual environments for package installation.

F-03: Sudo SETENV Python Execution → PYTHONPATH Hijacking (Root)

Severity: Critical
CWE: CWE-269 (Improper Privilege Management); CWE-250 (Execution with Unnecessary Privileges)
CVSS v3.1: 9.8 (Critical)

CVSS Justification: Local vector but leads to full root compromise via misconfigured sudo allowing environment manipulation.
Vector: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H`
Affected Component: `/opt/backup.py`
Attack Vector: Local (requires foothold)

Description

The `system-adm` user had sudo privileges to run the following command without a password and with environment variables preserved:

```
sudo -l
# (ALL) SETENV: NOPASSWD: /usr/bin/python3 /opt/backup.py
```

By abusing `PYTHONPATH`, a malicious module could be loaded by the privileged script, resulting in code execution as root.

Evidence

Evidence references

- sudo -l output + PYTHONPATH abuse + root flag proof: `evidence/priv-esc/sudo_py_backup_root.pn`

Impact

- Full root compromise
- Complete loss of Confidentiality, Integrity, Availability

Remediation

- Remove `SETENV` from sudo rules unless absolutely required.
- Add `env_reset` and explicitly whitelist safe environment variables.
- Run privileged scripts with `python3 -I` (isolated mode) or with hardened import handling.
- Restrict sudo command execution to root-owned scripts with secure permissions.

---

7. Attack Chain Summary

1. Nmap identified web service on TCP/8080 (Werkzeug/Flask)

2. Gobuster discovered `/login` and `/register`
3. SSTI confirmed via `{{7+7}}` template evaluation
4. SSTI exploited to gain RCE and reverse shell
5. Privilege escalation to `system-adm` using sudo `pip3 install .`
6. Root access gained using sudo `SETENV` with PYTHONPATH hijacking

---

8. Conclusion

The VulnNet: dotpy target was fully compromised due to a combination of a critical **SSTI vulnerability** and multiple **privilege escalation misconfigurations**. Addressing the web input rendering flaw and hardening sudo rules would significantly reduce the attack surface and prevent full system compromise.

Appendix A – Tools Used

- Nmap
- Gobuster
- Burp Suite
- Netcat
- wget
- Manual Linux enumeration commands

Appendix B – Evidence Index

The following evidence artifacts are referenced throughout this report and are intended to be included in the accompanying repository:

- `evidence/nmap/nmap_services.png`
- `evidence/web/gobuster.png`
- `evidence/web/ssti_confirm_14.png`
- `evidence/web/burp_baseline_request.png`
- `evidence/web/burp_ssti_payload_request.png`
- `evidence/exploit/reverse_shell_received.png`
- `evidence/priv-esc/setup_py_download.png`
- `evidence/priv-esc/pip3_install_privesc.png`
- `evidence/flags/user_flag.png`
- `evidence/priv-esc/sudo_py_backup_root.png`