1. Advanced Exploitation Lab

Activities:

- Exploited DVWA *File Upload* vulnerability to gain remote shell.
- Automated login & upload using a Python PoC (customized for CSRF bypass & Low Security mode).
- Captured a **Meterpreter session** via Metasploit listener.

Tools:

- **Metasploit** – payload generation & reverse shell handler
- **Python** – PoC automation (`dvwa_upload_poc.py`)
- **Exploit-DB** – reference for upload PoCs

Tasks:

- Chain exploit from **file upload vulnerability → RCE (Meterpreter)**
- Customize Python PoC (CSRF token handling, auto trigger)
- Document results (logs, payloads, impact, remediation)

Brief:

Exploit Chain Log:

Exploit ID	Description	Target IP	Status	Payload
004	File Upload → RCE Chain	192.168.0.107	Success	php/meterpreter/reverse_tcp

Customization:

We modified the **Exploit-DB PoC** for DVWA file upload to handle DVWA's login flow and missing CSRF tokens in *Low Security mode*. Added logic to:

- Auto-login with supplied credentials.
- Parse upload response to extract file path.
- Trigger uploaded PHP payload automatically.

50-word Summary of Changes:
The PoC script was modified to work with DVWA by automating login with username/password, handling optional CSRF tokens, uploading custom PHP reverse shell payloads, and triggering them automatically. Error handling was added for DVWA security modes, ensuring reliable exploitation. This customization allowed smooth integration with Metasploit for session capture.

Report

Title: Chained Exploit on Web Server

Findings:

- Vulnerability: File Upload → Remote Code Execution
- Host: 192.168.0.107
- Payload: php/meterpreter/reverse_tcp
- Reference: [Exploit inspired by Exploit-DB uploads PoCs]

Remediation:

- Implement strict file type validation (magic bytes, MIME check).
- Store uploads outside the web root.
- Apply security patches & harden PHP configuration.

Escalation Email (100 words):

Dear Development Team,

During penetration testing, we identified a critical file upload vulnerability in DVWA (host: 192.168.0.107). The application accepts PHP scripts without validation, allowing attackers to execute arbitrary code remotely. Using a crafted payload, we obtained a Meterpreter session on the server, confirming full compromise. Immediate remediation is required: enforce file validation, restrict upload directories, and update server configurations. Please prioritize this issue as it represents a direct path to Remote Code Execution (RCE). We strongly recommend deploying fixes in the next patch cycle and testing thoroughly.

Regards,
VAPT Security Analyst