DEPI Graduation Project - Lifecycle Phases and Detailed Scenario Reports

Note on safety and scope: This document avoids providing real malware binaries, exploit shellcode, or step-by-step instructions that would enable unauthorized remote code execution or distribution of ransomware. I cannot provide or link to real ransomware samples or provide code to produce working shellcode—this would be unsafe and disallowed. Instead the document includes safe, non-destructive simulation methods, complete forensic reporting templates, MITRE ATT&CK mappings, and detailed guidance for conducting the exercises in an air-gapped lab under institutional approval.

Project Structure: Lifecycle Phases (no weekly constraints)

Phase 1 — Preparation & Lab Setup
• Define IRT roles and responsibilities and finalize IR policy and communications plan.
• Build the lab: recommended VMs (Windows Server AD, 1–2 Windows clients, Linux mail/web server, attacker VM, analysis VM, forensic collector).
• Take baseline snapshots and configure logging (Sysmon, PowerShell logging, auditd, tcpdump). Secure the forensic evidence store and define chain-of-custody procedures.

Phase 2 — Simulation & Attack Execution (controlled, safe)
• Run phishing campaigns using GoPhish (hosted on attacker/host machine) targeting consenting test accounts only.
• Simulate payload behavior using non-destructive scripts (examples included) or Atomic Red Team emulations. Do NOT deploy real shellcode or unauthorized RCE payloads.
• Collect volatile and persistent evidence (memory dumps, disk images, event logs, PCAPs).

Phase 3 — Containment, Eradication & Recovery
• Analyze evidence to identify scope and root cause (Volatility, Autopsy, Plaso).
• Contain compromised systems (isolate VMs, disable accounts), eradicate artifacts, and recover from snapshots/backups.
• Record all actions in incident log; update playbooks.

Phase 4 — Post-Incident Analysis & Reporting
• Conduct lessons learned, produce executive and technical reports, and present findings.
• Update policies, controls, and detection coverage based on lessons.

Phishing Scenario — Full Report (GoPhish)

Overview:
This phishing exercise uses GoPhish running on the attacker/host machine and targets a single, consented Windows 10 VM in the isolated lab network. The goal is to exercise detection, evidence collection and analysis workflows when a user interacts with a

phishing landing page and (safely) submits test credentials or triggers a benign payload.

GoPhish Campaign Configuration

1. Install & host GoPhish on the attacker host or an isolated Linux VM. Ensure it has no Internet access and can only reach lab VMs.
2. Create an email template that closely resembles a legitimate internal message (use test-only recipient). Use placeholders for recipient.
3. Create a landing page in GoPhish that logs POSTs to a local file and displays an informative confirmation page (do NOT collect real credentials). The page should not automatically execute binaries.
4. Create a sending profile that points to your internal MailHog/local SMTP instance.
5. Launch the campaign to the consenting test account and monitor server logs and GoPhish logs.

Safe Payload Options (Allowed)

I cannot provide or enable real exploit shellcode or working remote code execution payloads. Below are safe alternatives you can use to simulate post-phish attacker behavior:

- **Credential capture only**: store posted form data (test strings) in a log file. This exercises detection of data exfil.
- **Non-destructive action**: serve a benign PowerShell script that copies test files to a separate folder (simulation of file tampering) — the script should never delete or encrypt originals.
- **Atomic Red Team emulations**: run benign atomic tests that emulate TTPs (e.g., PowerShell execution) without delivering malware.

If your supervisor/institution grants explicit approval for exploit-level testing inside an air-gapped environment, all such actions must be documented and supervised. I cannot provide exploit code or shellcode-building commands.

Phishing Evidence Collection Checklist

• Start network capture (tcpdump) on gateway: capture entire window of campaign.
• Export GoPhish campaign logs and MailHog SMTP logs.
• On victim VM: export Windows Event Logs (wevtutil), collect browser artifacts (history/cache), and capture RAM (FTK Imager / DumpIt) immediately after interaction.
• Create forensic disk image (FTK Imager or dd) and compute MD5/SHA256 hashes.
• Preserve copies of landing page HTML and any logged form-post files.

Forensic Analysis Steps

1. Memory analysis (Volatility 3): run PsList, CmdLine, Netstat plugins to find suspicious processes and network connections.
   Examples (Volatility 3):

- windows.pslist.PsList
- windows.cmdline.CmdLine
- windows.netstat.Netstat

2. Disk triage (Autopsy): ingest disk image, run keyword search for the phishing domain, and build timeline.
3. Timeline (Plaso/log2timeline -> psort): generate CSV of events and correlate email delivery, click time, and payload execution time.
4. IOC generation: extract the landing page URL, sender address, payload file names, and any process hashes.

MITRE ATT&CK Mapping (Phishing)

Primary technique(s):
- T1566: Phishing (initial access)
- T1204.002: User Execution: Malicious Link
- If PowerShell/script execution simulated: T1059.001 (PowerShell)

Phishing - Example Incident Report (Template)

Incident ID: PHISH-2025-001
Summary: Test user clicked a phishing link hosted via GoPhish; posted test credentials and triggered a benign simulation script that copied files to a test folder.
Impact: No real credentials or production data affected. Simulated RTA used only test data.
Evidence Collected: GoPhish logs, MailHog logs, web server access.log, victim memory image, disk image, PCAP.
Timeline (high level):
- 10:02:35 SMTP: campaign email delivered (MailHog)
- 10:05:12 Victim browser: GET landing page (webserver access.log)
- 10:05:30 Victim browser: POST form (landing page log)
- 10:05:45 RAM capture begun (FTK Imager)

Findings & IOCs: landing.example.lab, attacker@test.lab, simulated payload filename: sim_copy.ps1, hash: <sha256>
Mitigation: block landing.example.lab in virtual router, add IOC to detection rules, run endpoint scan for sim_copy.ps1, reset any test credentials.

Ransomware Scenario — Simulation and Full Report

Important safety note: I cannot provide links to or distribute real ransomware samples or binaries. Providing malware or download links that enable deployment would be unsafe and is disallowed. Instead, this section describes a comprehensive, realistic ransomware

simulation using safe, reversible tools and scripts (e.g., RanSim or a reversible encryption script), and includes a full forensic report template, MITRE mapping, and IOCs examples.

Safe Ransomware Simulation Options

1) RanSim / Ransomware simulators (non-destructive): these tools simulate the behaviour of ransomware (file modification, ransom note) without actually encrypting or destroying originals. Use them in an isolated lab.
2) Reversible Encryption Script (example included): create copies of target files into an encrypted_copy folder with the extension .encrypted while preserving originals. The script must support a decrypt mode for full recovery.
3) Atomic Red Team techniques for T1486 emulation: run behavior emulations that create the same telemetry as real ransomware without malicious payloads.

Reversible Ransomware Simulation Script (Concept)

A safe reversible simulation (PowerShell pseudo-code) was included earlier in your materials and should be used instead of real ransomware. It performs COPY operations and creates a ransom note. Always test on synthetic files and not on real user data.

Ransomware Evidence Collection Checklist

• Begin network capture (tcpdump) and ensure timestamps are synchronized across hosts.
• When encryption begins: capture memory image, running processes, and open network sockets.
• After activity: image disks, export Windows Event Logs, Sysmon logs, and collect ransom notes and modified file lists.
• Compute hashes for all evidence and record chain-of-custody.

Forensic Analysis Steps (Ransomware)

1. Use Autopsy to identify modified files, extract ransom note, and list filenames/timestamps.
2. Use Volatility to find process that performed file I/O and any loaded modules or network connections.
3. Correlate events in Plaso timeline to determine initial access vector and time-to-encrypt.
4. Produce IOCs: file extension patterns, ransom note text, file hashes, and any C2 IPs (if simulated).

MITRE ATT&CK Mapping (Ransomware)

Typical techniques:
- T1486: Data Encrypted for Impact

- T1566: Phishing (frequent initial vector)
- T1059: Command and Scripting Interpreter (PowerShell/Bash)
- T1078: Valid Accounts (lateral movement)

Map specific behaviors from your simulation to these techniques in your final technical report.

Sample Ransomware Incident Report (Template)

Incident ID: RANSIM-2025-001
Summary: Simulated ransomware executed in lab on VM 'Win10-Target' copying test files to *.encrypted and creating READ_ME.txt.
Impact: Test dataset files were modified (copies created). No real production data affected. Systems isolated and restored from snapshots.
Evidence collected: memory image, disk image, Sysmon logs, PCAP, ransom note file, file modification list.
Timeline (high level):
- 14:10:05: Simulation script executed (attacker VM)
- 14:10:17: First file modified (Autopsy)
- 14:12:30: RAM capture started
Findings & IOCs: simulated extension .encrypted, ransom note signature 'READ_ME - lab simulation', script name: ransim_safe.ps1, hash: <sha256>
Mitigation & Recovery: Isolated VM, restored from snapshot, removed simulation artifacts, updated playbook to detect .encrypted pattern and ransom note text.

Appendices & Templates

Appendix A — Chain of Custody template (example):
• Evidence ID | Description | Collected By | Date/Time | Hash (MD5/SHA256) | Storage Location | Notes

Appendix B — Incident Log / Timeline Table (example headers):
• Timestamp | Source (email/web/vm) | Event Description | Evidence File | Collected By

Appendix C — Contacts & Escalation (example):
• Incident Manager: Name, phone, email
  Legal Liaison: Name, phone, email
  Communications Lead: Name, phone, email

Appendix D — Recommended Tools & Configs:
• Volatility 3, Autopsy, Plaso, FTK Imager, tcpdump, Wireshark, GoPhish, MailHog, Sysinternals, REMnux.