Task: 01

Incident Response Simulation

Task: Conduct an incident response simulation to handle a security breach.

CODE:-

```
# Incident Response Simulation in Colab (Safe & Dummy Data)

import pandas as pd

# ------------------------------
# 1. Simulated Authentication Logs
# ------------------------------
auth_logs = pd.DataFrame([
    {"timestamp": "2025-08-30 10:15:00", "user": "alice", "action":
"login_fail", "source_ip": "192.168.1.15"},
    {"timestamp": "2025-08-30 10:15:05", "user": "alice", "action":
"login_fail", "source_ip": "192.168.1.15"},
    {"timestamp": "2025-08-30 10:15:10", "user": "alice", "action":
"login_success", "source_ip": "203.0.113.45"}, # suspicious IP
    {"timestamp": "2025-08-30 10:20:00", "user": "alice", "action":
"file_access", "file": "payroll.xlsx"}
])

print("=== Authentication Logs ===")
print(auth_logs)

# ------------------------------
# 2. Simulated Process Logs
# ------------------------------
process_logs = pd.DataFrame([
    {"timestamp": "2025-08-30 10:25:00", "host": "HR-PC01", "process":
"explorer.exe"},
    {"timestamp": "2025-08-30 10:25:05", "host": "HR-PC01", "process":
"powershell.exe -enc ZWNobyAnSGFja2VkISc="}, # suspicious
    {"timestamp": "2025-08-30 10:26:00", "host": "HR-PC01", "process":
"ransom_note.txt"} # indicator of ransomware
])

print("\n=== Process Logs ===")
print(process_logs)

# ------------------------------
# 3. Simulated Network Logs
```

```
# ----------------------------
network_logs = pd.DataFrame([
    {"timestamp": "2025-08-30 10:30:00", "host": "HR-PC01", "dest_ip":
"198.51.100.77", "port": 4444}, # C2 traffic
    {"timestamp": "2025-08-30 10:31:00", "host": "HR-PC01", "dest_ip":
"8.8.8.8", "port": 53} # normal DNS
])

print("\n=== Network Logs ===")
print(network_logs)

# ----------------------------
# 4. Basic Analysis
# ----------------------------
print("\n=== Analysis Results ===")
suspicious_ip = auth_logs[auth_logs["source_ip"] == "203.0.113.45"]
if not suspicious_ip.empty:
    print("⚠ ALERT: Login from unusual IP detected")

suspicious_proc =
process_logs[process_logs["process"].str.contains("powershell|ransom_note"
, case=False, na=False)]
if not suspicious_proc.empty:
    print("⚠ ALERT: Suspicious process activity detected")

c2_traffic = network_logs[network_logs["dest_ip"] == "198.51.100.77"]
if not c2_traffic.empty:
    print("⚠ ALERT: Possible C2 communication detected")

print("\nIncident Simulation Complete — investigate above alerts.")
```

OUTPUT:-

```
=== Authentication Logs ===
             timestamp   user         action     source_ip          file
0  2025-08-30 10:15:00  alice     login_fail  192.168.1.15           NaN
1  2025-08-30 10:15:05  alice     login_fail  192.168.1.15           NaN
2  2025-08-30 10:15:10  alice  login_success  203.0.113.45           NaN
3  2025-08-30 10:20:00  alice    file_access           NaN  payroll.xlsx

=== Process Logs ===
             timestamp     host                                  process
0  2025-08-30 10:25:00  HR-PC01                                explorer.exe
1  2025-08-30 10:25:05  HR-PC01  powershell.exe -enc ZWNobyAnSGFja2VkISc=
2  2025-08-30 10:26:00  HR-PC01                            ransom_note.txt
```

```
=== Network Logs ===
             timestamp     host        dest_ip  port
0  2025-08-30 10:30:00  HR-PC01  198.51.100.77  4444
1  2025-08-30 10:31:00  HR-PC01        8.8.8.8    53

=== Analysis Results ===
⚠ ALERT: Login from unusual IP detected
⚠ ALERT: Suspicious process activity detected
⚠ ALERT: Possible C2 communication detected

Incident Simulation Complete — investigate above alerts.
```

What this does:

- Creates **fake logs** (auth, process, network).
- Detects **red flags** (suspicious IP login, PowerShell execution, ransomware note, C2 traffic).
- Prints **alerts** like an IR analyst would see.

Next steps you can add:

- Add timelines (merge logs chronologically).
- Extract **IOCs** (IPs, process names).
- Create a **report dataframe** summarizing findings.
- Export results to CSV for your IR “report.