Scenario 5: Command & Control (C2) Beaconing Behavior (T1071/T1008)

Lab Setup

- **Victim Machine**: Windows 10
- **Attacker Machine**: Kali Linux (or simulate attacker domain: attacker.com)
- **SIEM Platform**: Elastic Security (Elastic Cloud)
- **Network Monitoring**: Packetbeat or Zeek (optional for HTTP monitoring)

Attack Simulation

Create C2 Beaconing Behavior

- On Victim (Windows 10), simulate periodic beacon using:

powershell

```
while ($true) { curl http://attacker.com/ping; Start-Sleep -Seconds 60 }
```

- Or use Task Scheduler to automate beacon every 60 seconds:

powershell

```
schtasks /create /sc minute /mo 1 /tn "C2Beacon" /tr "curl http://attacker.com/ping"
```

- This simulates periodic HTTP beaconing to attacker-controlled server.

Detection Strategy

A. **Monitor Rare Outbound Domains**

- Collect outbound HTTP/HTTPS logs:
  - Use **Packetbeat** or **Firewall Logs** to capture DNS and HTTP requests.
  - Collect via **Filebeat → Elastic Security**.

B. **Frequency-Based Correlation**

- Detect frequent connections to the same domain/IP at regular intervals.

Elastic KQL Query Example:

kql

```
url.domain: "attacker.com"

| stats count() by url.domain, date_histogram(field="@timestamp", fixed_interval="1m")

| where count >= 1
```

- Look for domains contacted every 60 seconds consistently.

Alternative (For rare domains):

kql

```
url.domain: *

| stats count() by url.domain

| where count < 10
```

- Combine both queries to detect beaconing to rare domains with regular interval access.

Optional: Visualize Beacon Pattern

- Use Elastic **Visualizations → Line Chart** with:
  - X-axis: Timestamp
  - Y-axis: Count of HTTP requests to domain
  - Filter: url.domain: "attacker.com"
- Regular spikes every 60 seconds indicate beaconing.

Summary

- ✅ Simulated periodic beacon using curl every 60 seconds
- ✅ Collected outbound HTTP logs into Elastic
- ✅ Detected beacon behavior via frequency-based correlation