# Essentials of Cybersecurity & Cyber Warfare — Lab Practicals (Windows & Linux)

# Practical 1 — Windows Security Log Analysis: Detecting Failed Logon / Brute-force Patterns

**Objective**: Use Windows Event Logs (Security) and PowerShell to identify failed logon attempts (Event ID 4625), extract source IP addresses, count attempts per IP, and export findings to CSV.

## Prerequisites:

• A Windows 10/11 or Windows Server lab VM (administrator privileges).
• PowerShell (built-in).
• A sample Security event log with failed logons
•

## Tools & Files:

• PowerShell (Run as Administrator)
• Event Viewer (GUI) - optional

## Step-by-step Tasks:

1. Open PowerShell as Administrator.

2. (Optional) If you have an exported EVTX file named security_sample.evtx in C:\labs, import it into a temporary log session so you do not modify the live log. Use wevtutil or the Event Viewer GUI to open the file. Example (Event Viewer): File → Open Saved Log → select the file.

3. Query Security events for failed logons (Event ID 4625) from the live Security log for the last 7 days and display the most important fields (TimeCreated and Message). Run:

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4625; StartTime=(Get-
Date).AddDays(-7)} |
```

4. Extract only the IP addresses and count occurrences to find top offenders. Run:

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4625; StartTime=(Get-
Date).AddDays(-7)} |
```

5. Export the raw events with selected fields to CSV for reporting:

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4625; StartTime=(Get-
Date).AddDays(-7)} |
```

6. OPTIONAL: If IP addresses are internal RFC1918 (e.g., 192.168.x.x) but you suspect external scanning via VPN or forwarded logs, cross-check firewall logs or perimeter devices.

7. Create a short remediation action list (block IPs, force password resets for accounts showing multiple failures, enable MFA).

## Solution / Expected Outputs (example):

Example of the PowerShell "Group-Object" output (counts are illustrative):

```
Count  Name
-----  ----
   45  203.0.113.45
   12  198.51.100.12
    9  10.0.0.12
    4  192.168.1.25
```

The exported CSV (failed_logons_report.csv) will contain rows with TimeCreated, Account, IpAddress,

# Practical 2 — Linux Network Traffic Analysis: Detecting Port Scans & SSH Brute-force

**Objective**: Capture and analyze network traffic to detect a TCP SYN port scan and multiple failed SSH connection attempts. Produce a report listing top source IP addresses and demonstrate blocking an attacking IP with iptables/ufw.

## Prerequisites:

• A Linux lab VM (Ubuntu/Debian/CentOS) with sudo privileges.
• tcpdump and tshark (Wireshark CLI) installed. If tshark is not installed, use tcpdump + tcpdump filters + awk.
• A second VM or attacker simulator (instructor-provided pcap) to generate benign attack traffic in the lab network.

## Tools & Files:

• tcpdump (capture) - sudo tcpdump
• tshark (analysis) - optional
• awk, sort, uniq - built-in utilities
• An offline pcap (optional) if you cannot generate traffic.

## Step-by-step Tasks:

1. Identify the network interface to capture (e.g., eth0). Run:
```
ip -brief a
```

2. Start a tcpdump capture to a pcap file (capture SYN packets only to reduce size):
```
sudo tcpdump -i eth0 'tcp[tcpflags] & tcp-syn != 0 and tcp[tcpflags] & tcp-ack == 0' -w /tmp/syn_on
```

3. Generate or wait for network activity (instructor may trigger a scan from attacker VM; otherwise use provided pcap). Capture for 1-2 minutes.

4. Stop capture (Ctrl+C) and display top source IPs that sent SYNs (quick analysis using tcpdump + awk):
```
sudo tcpdump -nn -r /tmp/syn_only.pcap -tt | awk '{print $3}' | sed 's/:\{0,1\}[0-9]*$//' | sort |
```

5. Use tshark to extract TCP flags and count SYNs per source IP (if tshark installed):
```
tshark -r /tmp/syn_only.pcap -T fields -e ip.src -e tcp.flags.syn | awk '$2=="1"{print $1}' | sort
```

6. To detect SSH brute-force (multiple failed TCP connections to port 22), filter the capture for port 22 and count connection attempts per IP:
```
sudo tcpdump -nn -r /tmp/syn_only.pcap 'port 22' | awk '{print $3}' | sed 's/:\{0,1\}[0-9]*$//' |
```

7. Create a short mitigation: block the top attacking IP (replace ):
```
sudo iptables -A INPUT -s <attacker-ip>
-j DROP # or on Ubuntu with ufw:
# sudo ufw deny from <attacker-ip> to any
```

8. OPTIONAL: Use fail2ban for automated blocking; show how to test and check status:
```
sudo systemctl status fail2ban
# or show jail status: sudo fail2ban-client status sshd
```

## Solution / Expected Outputs (example):

```
Sample tcpdump/awk output (illustrative):

   120  203.0.113.81
```

```
    34  198.51.100.9
    12  192.168.1.101

This shows 203.0.113.81 sent 120 SYN packets—likely a port scan or brute-force source.
After blocki

Example iptables command to list rules (verify block
applied): sudo iptables -L INPUT -n --line-numbers
```