Brute Force room

Created by	Ⓜ Manikandan N
Created time	@December 24, 2025 1:25 PM
Last edited by	Ⓜ Manikandan N
Last updated time	@December 24, 2025 1:27 PM
Category	blue team room what i learned and completed

✅ Brute Force Room – What *I* Did 🛡️

✔️ Audit Failure counting 🔍

- I **did not search for the literal word** `Audit Failure`
- I counted events using the phrase:

```
An account failed to logon
```

- Because this phrase represents **Windows Event ID 4625**
- This correctly identifies **Audit Failure events**

✔️ Username identification 👤

- Looked for the repeated **local account name**
- Same username appeared across multiple failed logons
- This showed the account being **targeted**

✔️ Event ID 🧾

- Found **Event ID 4625**

- This confirms:
    - Logon failure
    - Audit Failure
    - Authentication attack

✔️ Source IP 🌐

- Extracted the IP that appeared repeatedly
- Same IP across many failures = attacker

✔️ Country of attacker 🌍

- Used:

    ```
    curl ipinfo.io/<IP>
    ```

- Found country based on **IP ownership**
- Understood this is **GeoIP enrichment**, not physical proof

---

✔️ Source Port Range 🔢

- Logs showed:

    ```
    Source Port: <number>
    ```

- Extracted **all source ports**
- Found:
    - Lowest port → `49162`
    - Highest port → `65534`

✅ Final answer format:

```
49162-65534
```

⚠️ What I Struggled With (REAL STRUGGLES) 🤯

1️⃣ “Audit Failure” confusion

- Initially thought logs must contain the exact words
- Learned that **Audit Failure = failed security action**
- Keyword ≠ meaning

2️⃣ Regex `{1,5}` and spaces

- Didn’t understand why regex wasn’t matching
- Learned:
  - Ports have max **5 digits**
  - Logs can have **any number of spaces**
  - Regex must be flexible

3️⃣ Why two `grep` commands

- Thought one `grep` should be enough
- Learned:
  - First `grep` → filter correct lines
  - Second `grep` → extract only the number
- This is **normal SOC workflow**

4️⃣ GeoIP misunderstanding

- Tried to infer country from authentication fields
- Learned:
  - Country ONLY comes from IP
  - Authentication info ≠ network info

🧠 Small → Big Things I Need to Remember 📈

🔷 Small (Basics)

- `grep` = search
- `i` = ignore case
- `o` = only matching text
- `E` = extended regex

🔷 Medium (Log Analysis)

- `{1,5}` → valid port digits
- `[[:space:]]*` → any spacing (even zero)
- `sort -n` → numeric sort
- `head -1` → lowest value
- `tail -1` → highest value

🔷 Big (SOC Thinking) 🧠

- One failure = noise
- Patterns = attack
- Same IP + many failures = brute force
- Wide ephemeral port range = automation
- GeoIP = context, not evidence

🧪 Exact Commands I Used (AND WHY) 💻

Count Audit Failures

```
grep -i"An account failed to log on" logs.txt |wc -l
```

👉 Counts Event ID 4625 failures

Extract Source Ports

```
grep -oE'Source Port:[[:space:]]*[0-9]{1,5}' logs.txt | grep -oE'[0-9]{1,5}'
```

👉 First filter → then extract number

Find Lowest Source Port

```
grep -oE'Source Port:[[:space:]]*[0-9]{1,5}' logs.txt | grep -oE'[0-9]{1,5}' |sort -n |head -1
```

Find Highest Source Port

```
grep -oE'Source Port:[[:space:]]*[0-9]{1,5}' logs.txt | grep -oE'[0-9]{1,5}' |sort -n |tail -1
```

GeoIP Lookup 🌍

```
curl ipinfo.io/<IP>
```

🧠 Final SOC Conclusion (MY WORDS) 🧑‍💻

> I identified multiple Windows audit failure events (Event ID 4625) targeting a local account from a single external IP. The repeated failures and use of a wide ephemeral source port range (49162–65534) indicate an automated brute-force login attack.