SIEM Rules Detection Task

Yehia A. Mostafa

1) Task Description

Create detection rules for:

1. **User account and group enumeration**

   Example: `net user`, `net group`, `whoami /groups`

2. **User account creation using net command**

   Example: `net user attacker Passw0rd! /add`

3. **Privilege escalation by adding a new user to the Administrators group**

   Example: `net localgroup administrators attacker /add`

4. **Credential dumping by extracting the Security Account Manager (SAM)**

   Example: `reg save HKLM\SAM` or tools that attempt SAM database access

Bonus (Optional)

Create a **correlation rule** (EQL) to detect a brute force attack by correlating **multiple failed logons** followed by a **successful logon** from the same account and source within a short timeframe.

Reference

- https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/

2) Environment & Prep

- Increased **Ubuntu Server RAM to 8 GB** *(Server was irresponsive due to lack of resources)*.
- Installed **Sysmon** to capture process details (e.g., `process.args`) and ship them to Elastic:
  1. Downloaded **Sysmon64**

https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

2. Downloaded **Sysmon config** (XML) from GitHub (repo with exported config).
3. Installed with:

```
.\Sysmon64.exe -accepteula -i sysmonconfig-export.xml
```

4. Verified service:

```
Get-Service Sysmon64
```

5. In **Kibana → Fleet → Agents → Edit policy → Add Windows integration → Activate Sysmon logging**.

---

3) Practice Task discussed in session:

Created this PS Script

```
$TestUser = "elasticTestUser"
$Password = ConvertTo-SecureString "P@ssw0rd123!" -AsPlainText -Force

Write-Host "=== Task 1: Generating Windows Admin Events for Elastic ==="

# 1. User Creation
Write-Host "[+] Creating test user: $TestUser"
New-LocalUser -Name $TestUser -Password $Password -FullName "Elastic T
est User" -Description "Test user for Elastic log events"

# 2. User Deletion (create another temp user then delete)
$TempUser = "tempDeleteUser"
Write-Host "[+] Creating and deleting user: $TempUser"
New-LocalUser -Name $TempUser -Password $Password -FullName "Temp
Delete User" -Description "User to be deleted"
Remove-LocalUser -Name $TempUser
```

```
# 3. Lockout User (simulate failed logon attempts if policy allows)
Write-Host "[+] Simulating failed logins to lock out user: $TestUser"
for ($i=0; $i -lt 10; $i++) {
    Start-Process "powershell.exe" -ArgumentList "net use \\localhost\IPC$ /us
er:$env:COMPUTERNAME\$TestUser WrongPassword!" -WindowStyle Hidde
n
    Start-Sleep -Seconds 2
}

# 4. Privilege Escalation (add user to Administrators)
Write-Host "[+] Adding $TestUser to Administrators group"
Add-LocalGroupMember -Group "Administrators" -Member $TestUser

# 5. Disable User
Write-Host "[+] Disabling $TestUser"
Disable-LocalUser -Name $TestUser

# 6. Add New Policy (audit policy change as example)
Write-Host "[+] Modifying audit policy"
auditpol.exe /set /subcategory:"Logoff" /success:enable
```

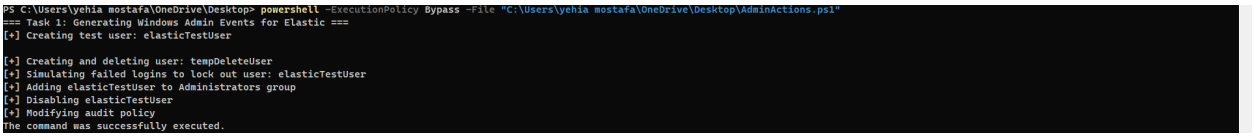
4) Test Activity Scripts (to trigger all detections)

4.1 AdminActions.ps1 (initial user/admin events script)

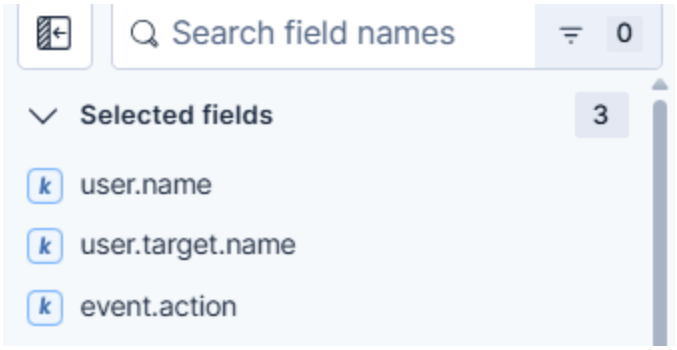
> If you see errors when running, you can use:
>
> `powershell -ExecutionPolicy Bypass -File "C:\Users\yehia mostafa\OneDrive\Desktop\AdminActions.ps1"`

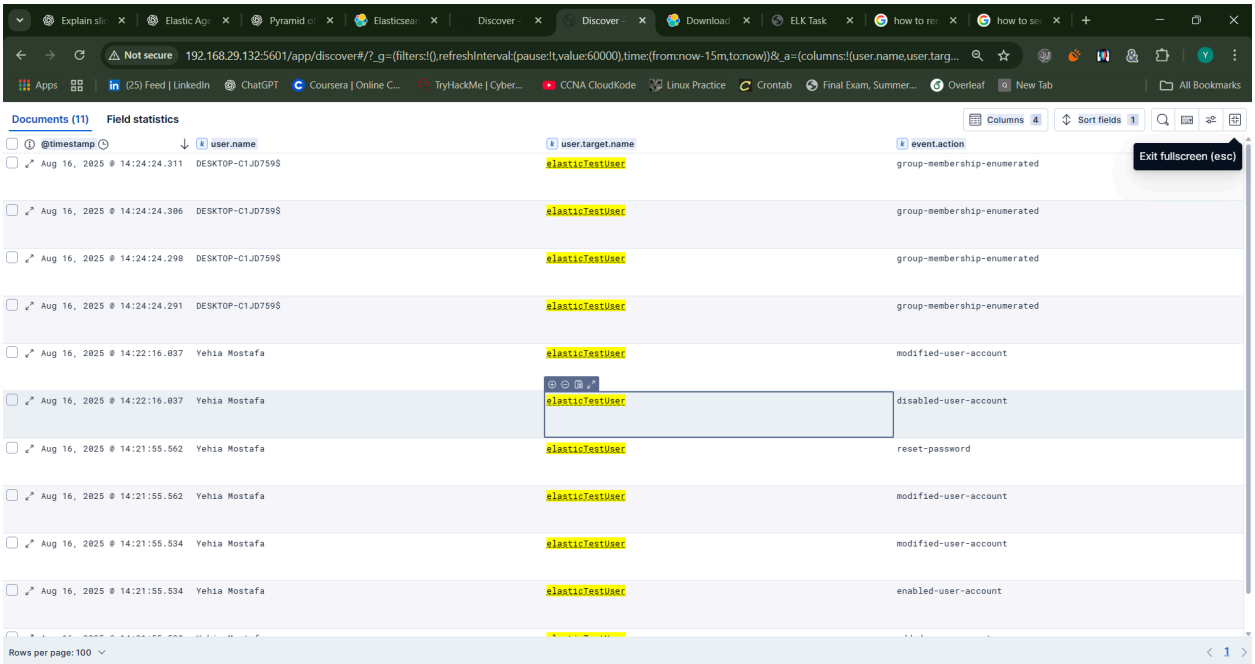
```
$TestUser = "elasticTestUser"
$Password = ConvertTo-SecureString "P@ssw0rd123!" -AsPlainText -Force
```

```
Write-Host "=== Task 1: Generating Windows Admin Events for Elastic ==="

# 1. User Creation
Write-Host "[+] Creating test user: $TestUser"
New-LocalUser -Name $TestUser -Password $Password -FullName "Elastic T
est User" -Description "Test user for Elastic log events"

# 2. User Deletion (create another temp user then delete)
$TempUser = "tempDeleteUser"
Write-Host "[+] Creating and deleting user: $TempUser"
New-LocalUser -Name $TempUser -Password $Password -FullName "Temp
Delete User" -Description "User to be deleted"
Remove-LocalUser -Name $TempUser

# 3. Lockout User (simulate failed logon attempts if policy allows)
Write-Host "[+] Simulating failed logins to lock out user: $TestUser"
for ($i=0; $i -lt 10; $i++) {
    Start-Process "powershell.exe" -ArgumentList "net use \\localhost\IPC$ /us
er:$env:COMPUTERNAME\$TestUser WrongPassword!" -WindowStyle Hidde
n
    Start-Sleep -Seconds 2
}

# 4. Privilege Escalation (add user to Administrators)
Write-Host "[+] Adding $TestUser to Administrators group"
Add-LocalGroupMember -Group "Administrators" -Member $TestUser

# 5. Disable User
Write-Host "[+] Disabling $TestUser"
Disable-LocalUser -Name $TestUser

# 6. Add New Policy (audit policy change as example)
Write-Host "[+] Modifying audit policy"
auditpol.exe /set /subcategory:"Logoff" /success:enable
```



- **Kibana → Discover** → selected relevant fields for visibility and filtering.
- Excluded empty rows for `user.name` or `user.target.name` (hover field → **minus**).
- Filtered by the created test user `elasticTestUser` using:
  `user.target.name: elastic*`





success — events visible and attributable.

4.2 AdminActionsTest.ps1 (bulk trigger for the 4 rules + cleanup)

```
# ============================
# AdminActionsTest.ps1
# Script to trigger detection rules in Elastic via Windows Security Events
# ============================

Write-Host ">>> Starting detection trigger script..." -ForegroundColor Cyan

# 1. Recon / Enumeration (net.exe + whoami.exe)
Write-Host ">>> Running enumeration commands..." -ForegroundColor Yello
w
Start-Process -FilePath "net.exe" -ArgumentList "user" -NoNewWindow -Wait
Start-Process -FilePath "net.exe" -ArgumentList "group" -NoNewWindow -Wa
it
Start-Process -FilePath "whoami.exe" -ArgumentList "/groups" -NoNewWind
ow -Wait

# 2. User Creation (Event ID 4720)
Write-Host ">>> Creating test user 'elasticTestUser'..." -ForegroundColor Yell
ow
net user elasticTestUser Passw0rd! /add

# 3. Privilege Escalation (Add to Administrators group, Event ID 4732)
Write-Host ">>> Adding test user 'elasticTestUser' to Administrators group..."
-ForegroundColor Yellow
net localgroup Administrators elasticTestUser /add

# 4. Credential Dumping Attempt (reg.exe saving SAM hive)
Write-Host ">>> Attempting to dump SAM hive (this may be blocked)..." -Fore
groundColor Yellow
try {
    Start-Process -FilePath "reg.exe" -ArgumentList "save HKLM\SAM C:\Temp
\SAM_dump.save" -NoNewWindow -Wait
} catch {
    Write-Host ">>> Failed to dump SAM hive (expected if no permissions)." -F
oregroundColor Red
```

```
}

# ============================
# CLEANUP SECTION
# ============================

Write-Host ">>> Cleaning up test user and reverting changes..." -Foreground
Color Cyan

# Remove user from Administrators group
net localgroup Administrators elasticTestUser /delete

# Delete user account
net user elasticTestUser /delete

# Remove dump file if it exists
if (Test-Path "C:\Temp\SAM_dump.save") {
    Remove-Item "C:\Temp\SAM_dump.save" -Force
    Write-Host ">>> Removed SAM hive dump file." -ForegroundColor Yellow
}

Write-Host ">>> Cleanup complete. Test finished." -ForegroundColor Green
```

5) Detection Use Cases (Rules)

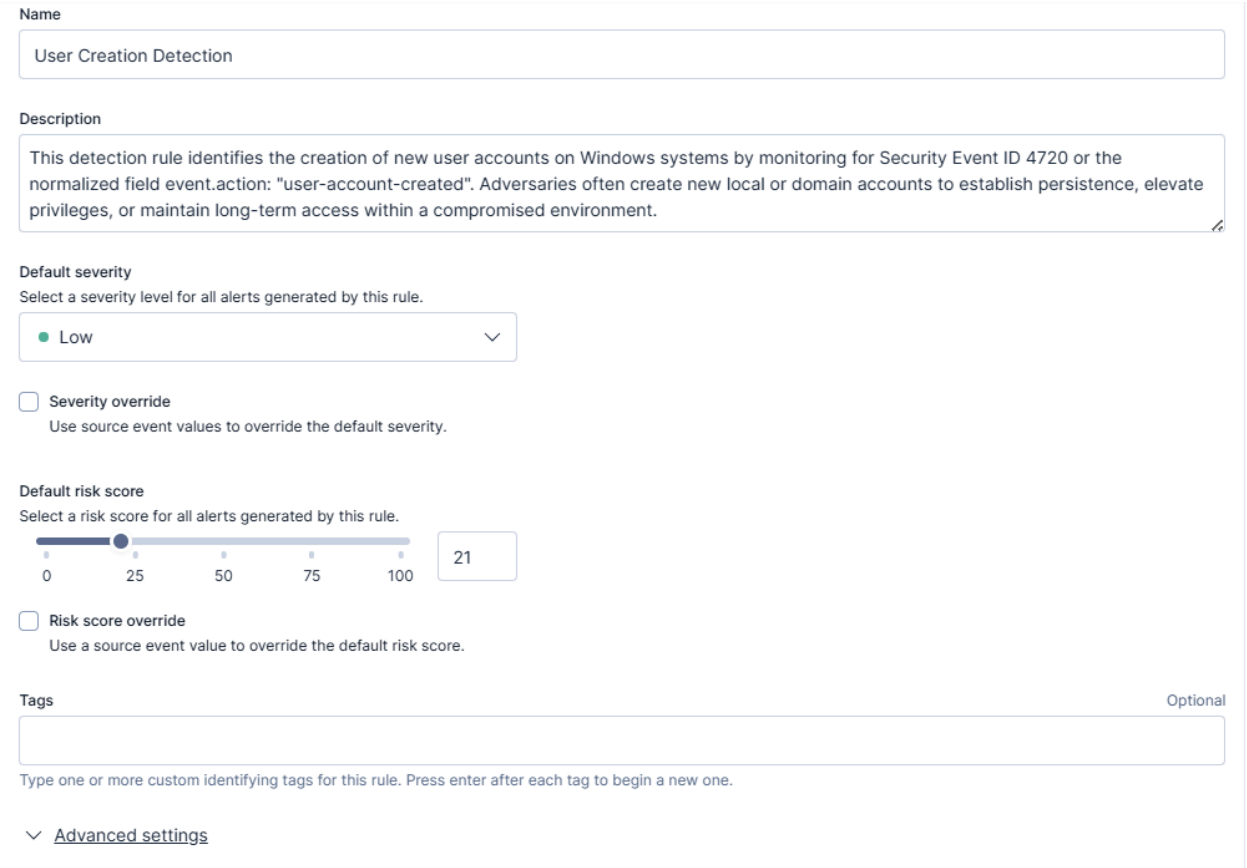
5.1 User Account Creation (net user)

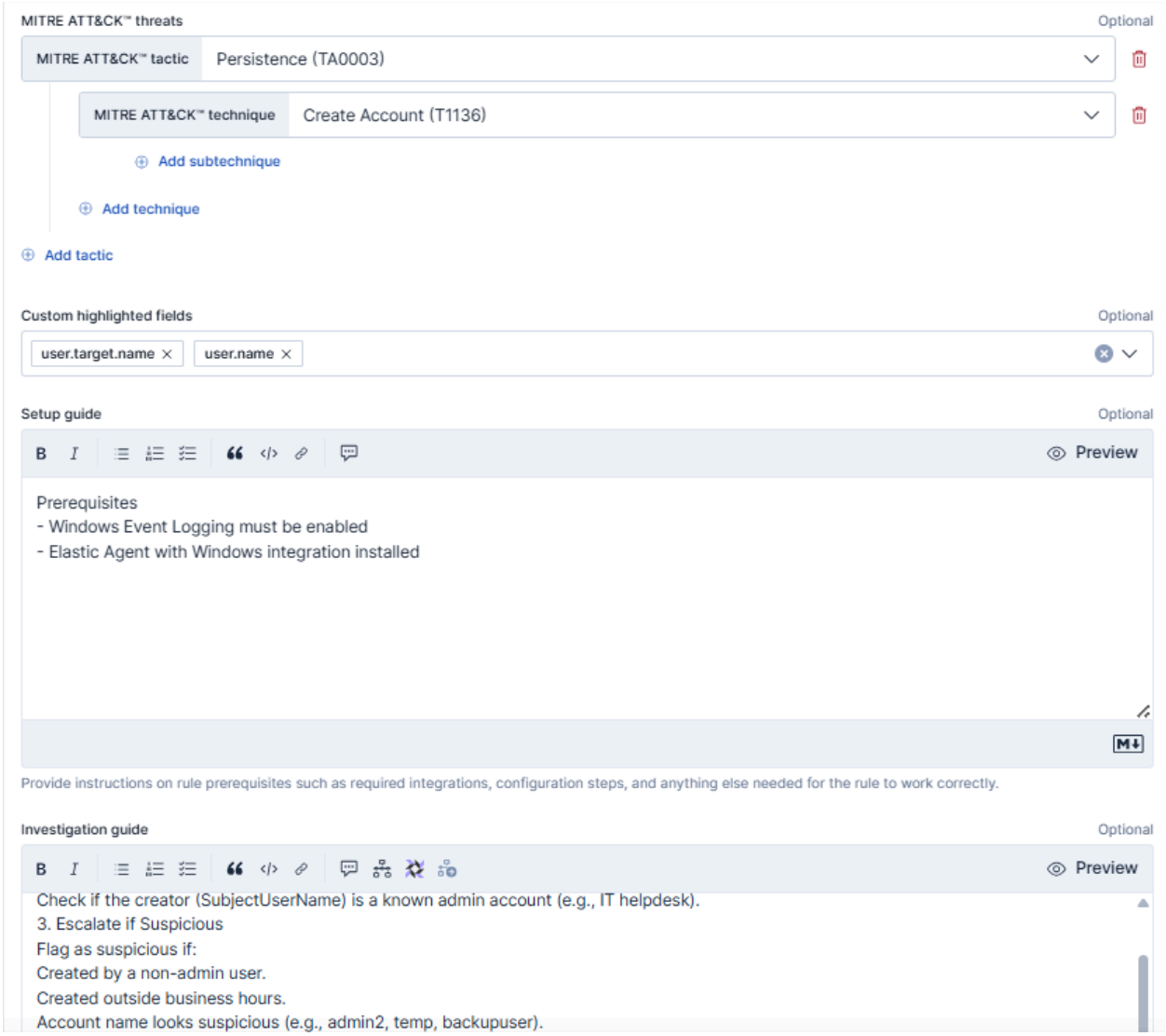
Rule Name

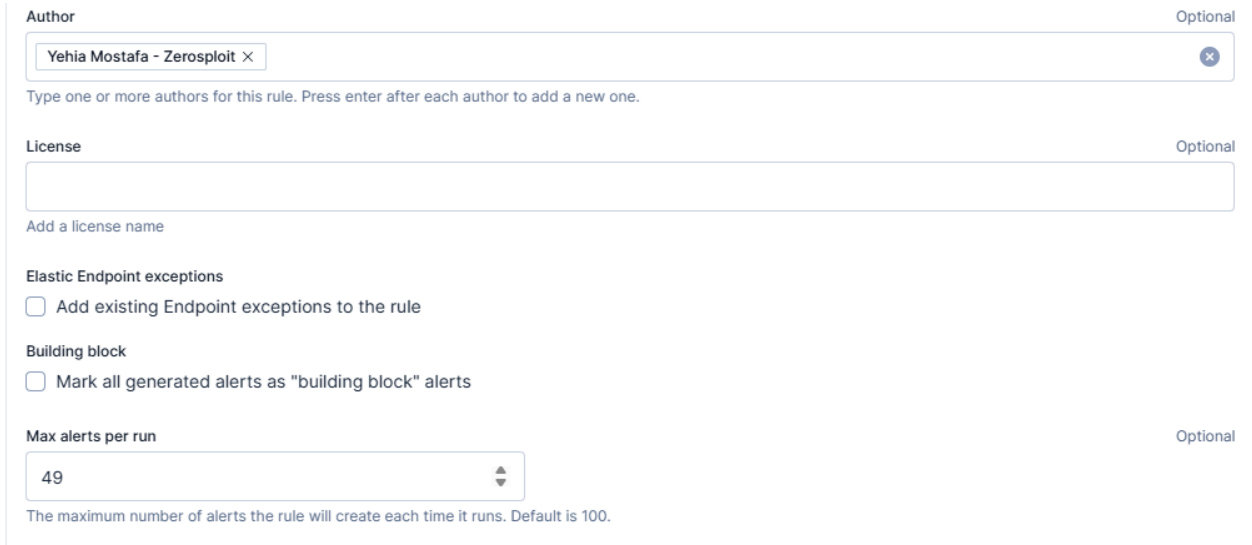
Windows – User Account Created

Detection Logic (KQL)

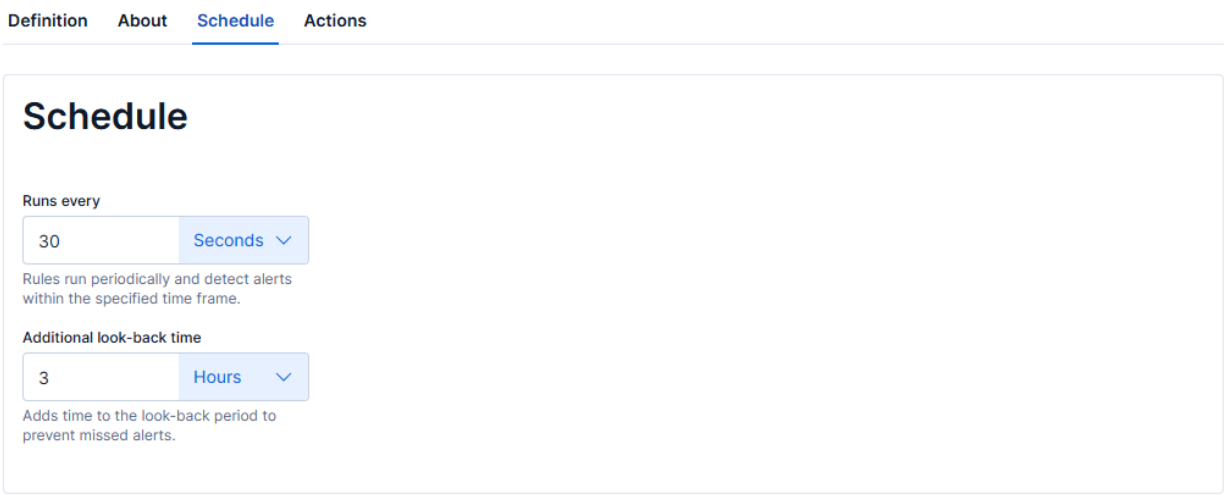
```
event.action: "user-account-created" or event.code: "4720"
```



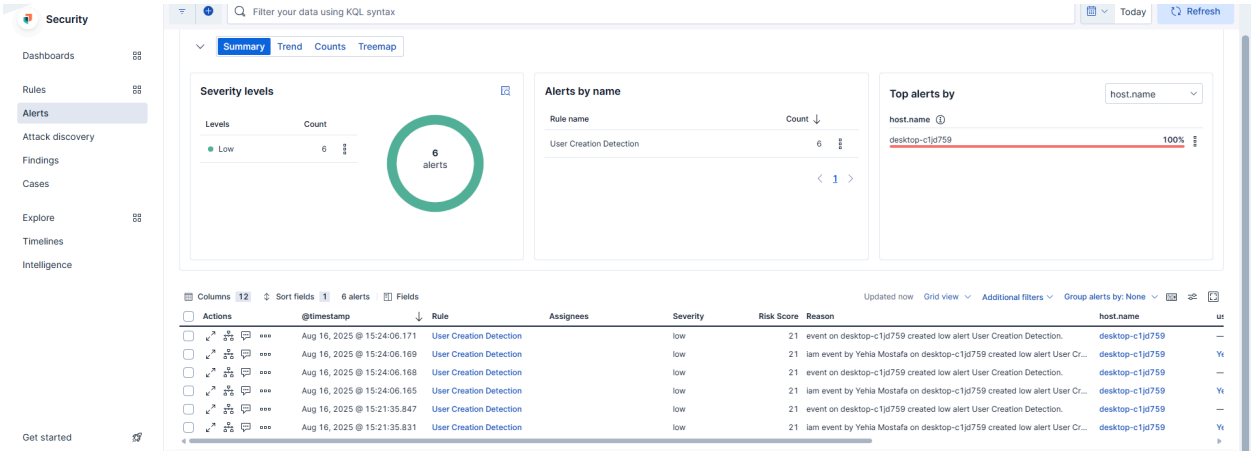

MITRE ATT&CK™ threats
Optional
MITRE ATT&CK™ tactic
Persistence (TA0003)
MITRE ATT&CK™ technique
Create Account (T1136)
Add subtechnique
Add technique
Add tactic
Custom highlighted fields
Optional
user.target.name
user.name
Setup guide
Optional
Preview
Prerequisites
- Windows Event Logging must be enabled
- Elastic Agent with Windows integration installed
Provide instructions on rule prerequisites such as required integrations, configuration steps, and anything else needed for the rule to work correctly.
Investigation guide
Optional
Preview
Check if the creator (SubjectUserName) is a known admin account (e.g., IT helpdesk).
3. Escalate if Suspicious
Flag as suspicious if:
Created by a non-admin user.
Created outside business hours.
Account name looks suspicious (e.g., admin2, temp, backupuser).


Author
Optional
Yehia Mostafa - Zerosploit
Type one or more authors for this rule. Press enter after each author to add a new one.
License
Optional
Add a license name
Elastic Endpoint exceptions
Add existing Endpoint exceptions to the rule
Building block
Mark all generated alerts as "building block" alerts
Max alerts per run
Optional
49
The maximum number of alerts the rule will create each time it runs. Default is 100.

The lookback time is the time it rewinds and starts checking since.



Security → Alerts



success.

5.2 User Account & Group Enumeration (Recon)

Using Same Configurations as mentioned above.

Detection Logic (KQL)

```
process.name:("net.exe" or "whoami.exe") and process.args:("user"or "grou
```

```
p" or "/groups")
```

5.3 Privilege Escalation (added to Administrators)

Detection Logic (KQL)

```
event.action: "added-to-group" or event.code: "4732"
```

5.4 Credential Dumping Attempt (SAM)

Detection Logic (KQL)

```
process.name: "reg.exe"
and process.args: ("save" and "HKLM\\SAM")
```

Results for the bulk test:

success.

6) Bonus: Brute Force Detection (Correlation)

> According to CrowdStrike, ***typical brute force attacks can attempt many guesses per second.***
>
> ***Implemented detection focuses on clear failed-attempt bursts.***

Correlation Rule (EQL)

```
sequence by host.id, source.ip, user.name with maxspan=300s
  [authentication where winlog.event_id == "4625" and event.outcome == "fail
ure"] with runs=10
```

- **Meaning**: For the same `host.id` + `source.ip` + `user.name`, alert when there are **≥10** failed logons (`4625`) within **300s**.

---

8) Hydra Test Setup (Windows SSH & Firewall)

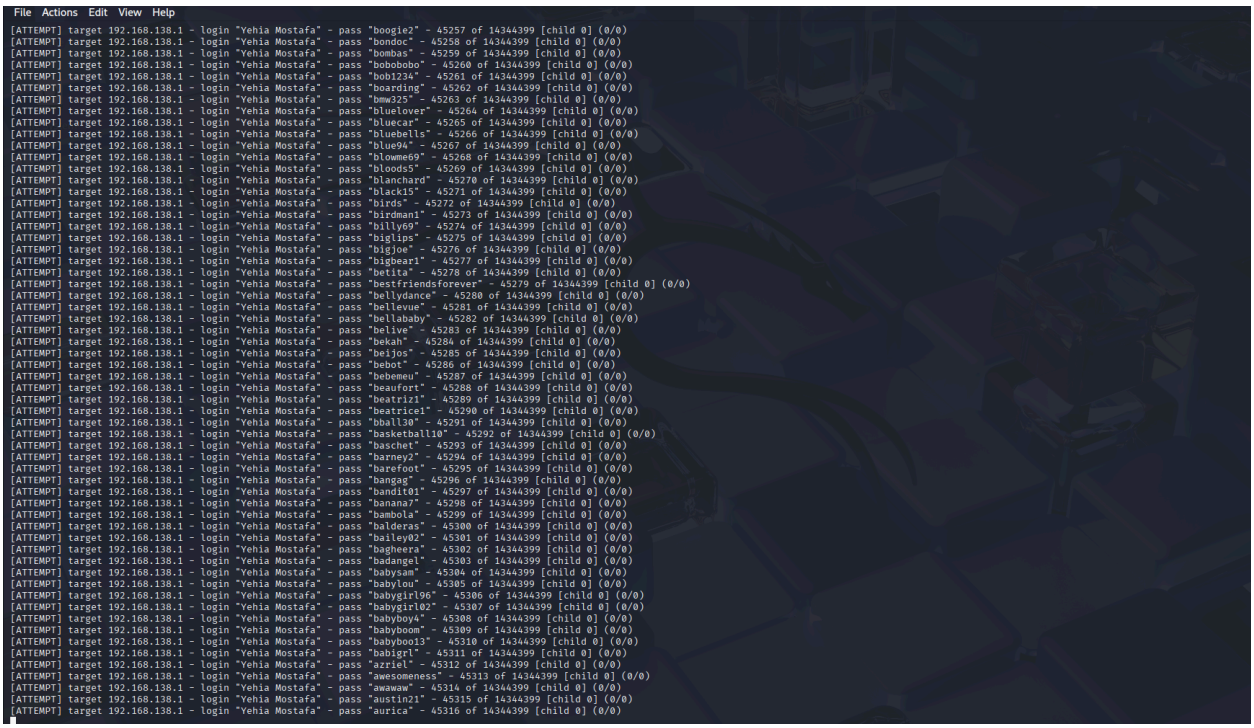
To brute force against the Windows host via SSH:

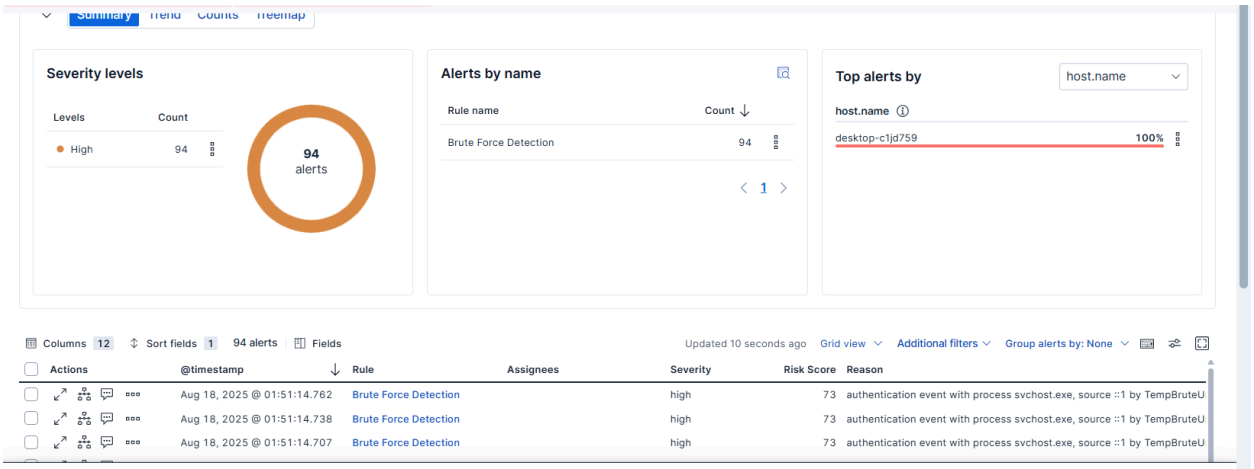
Install OpenSSH Server on Windows host

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

# Start the service
Start-Service sshd

# Start on boot
Set-Service -Name sshd -StartupType 'Automatic'

# Allow SSH in the firewall
New-NetFirewallRule -Name sshd -DisplayName "OpenSSH Server" -Protocol
TCP -LocalPort 22 -Action Allow -Direction Inbound
```

Hydra command (from Kali/Ubuntu)

```
hydra -l "Yehia Mostafa" -P /usr/share/wordlists/rockyou.txt ssh://192.168.138.1 -t 1 -V
```





success.

---

9) EQL Field Mapping Conflict – Problem & Workaround

Observed error:

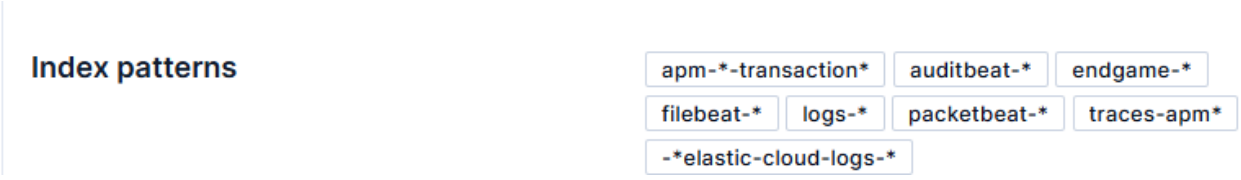
```
Found 2 problems line 2:25: Cannot use field [winlog.event_id] due to ambiguities being mapped as
 [2] incompatible types: [text] in [winlogbeat-2025.08.14, winlogbeat-2025.08.16, winlogbeat-2025.08.17],
  [keyword] in [.ds-logs-system.application-default-2025.08.11-000001, .ds-logs-system.security-default-2025.08.11-000001,
  .ds-logs-system.system-default-2025.08.11-000001, .ds-logs-windows.powershell-default-2025.08.17-000001,
  .ds-logs-windows.powershell_operational-default-2025.08.17-000001, .ds-logs-windows.sysmon_operational-default-2025.08.17-000001] line 2:55:
  Cannot use field [event.outcome] due to ambiguities being mapped as [2] incompatible types:
   [text] in [winlogbeat-2025.08.14, winlogbeat-2025.08.16,
  winlogbeat-2025.08.17], [keyword] in [.ds-logs-system.auth-default-2025.08.11-000001, .ds-logs-system.security-default-2025.08.11-000001,
  .ds-logs-windows.powershell-default-2025.08.17-
  000001, .ds-logs-windows.powershell_operational-default-2025.08.17-000001, .ds-logs-windows.sysmon_operational-default-2025.08.17-000001]
```

Root cause

- Mixed mappings across indices: some map `winlog.event_id` / `event.outcome` as **text**, others as **keyword**. EQL needs consistency.

Temporary workaround applied

- **Unlinked** problematic indices from the rule so the query runs only against indices with consistent (keyword) mappings.



What “unlinking” did

- Rule stopped querying the broken index pattern → **errors disappeared**.

- **Trade-off**: events in unlinked indices are **ignored** (no detections from them).

Future correct approach (planned)

1. Reindex into a new index (e.g., `winlogbeat-fixed`) with **proper mappings** (`keyword` for `winlog.event_id`, `event.outcome`, etc.).
2. Delete or stop using the broken index (`winlogbeat-2025.08.17`).
3. Point rules only at the **fixed index** or a pattern including only fixed indices.

---

10) Cleanup & System State

- Removed temporary test users and group membership changes in the script (cleanup section).
- Verified only intended local users remain (kept system accounts disabled).
