Report: Basic Auditing & Logging

1. Introduction

#

Auditing and logging are fundamental components of any secure Active Directory environment. By monitoring account activity such as logon attempts and account lockouts, organizations can detect suspicious behavior, enforce accountability, and comply with security policies. Proper audit configurations help administrators maintain visibility over user actions and system events that might indicate breaches or misuse.

This task focuses on configuring local audit policies and leveraging system tools to observe authentication-related events. By enabling the right audit settings and reviewing relevant logs, administrators gain critical insights into failed login attempts and locked-out user accounts, which are often early indicators of brute-force attacks or user errors.

2. Objective

The primary objective of this task is to implement and validate auditing mechanisms that monitor user authentication behavior within a Windows environment. By enabling audit policies through the Local Group Policy Editor, the system can begin logging key events such as successful and failed login attempts, account lockouts, and changes to user credentials. These logs provide critical visibility into how accounts are accessed and managed across the network, which is essential for identifying misuse, policy violations, or potential attack attempts.

In addition to configuring auditing, the task includes using built-in tools like **Event Viewer** and **PowerShell** to examine system logs and locate accounts that have been locked due to repeated login failures. This objective supports the broader goal of proactive threat detection and enhances administrative efficiency by allowing real-time response to suspicious activity. Ultimately, these auditing capabilities help strengthen the overall security posture of the domain while ensuring that activities remain transparent, traceable, and in line with best practices for compliance and accountability.

Key goals include:

- **Enable Account Logon Auditing**
  The primary goal was to activate auditing for account logon and logoff events via the Local Group Policy Editor. This helps track who is accessing the system and when. It's crucial for identifying unauthorized access attempts.

- **Monitor Failed Login Attempts**
  By reviewing audit logs, we aimed to identify incorrect login attempts, which may indicate brute-force attacks or forgotten passwords. This improves early detection of security incidents. It also helps reduce account lockouts.

- **Detect Locked-Out Accounts**
  Using PowerShell, we searched for accounts that were locked due to repeated login failures. This helps administrators respond quickly to potential misuse. It also supports efficient user account management.

- **Strengthen Security Visibility**
  Implementing auditing improves overall security visibility into user and system behavior. It ensures that every login attempt is logged and reviewable. This enhances accountability and traceability in the domain.

- **Support Compliance and Investigation**
  Audit logs serve as forensic evidence during internal investigations or external audits. They help ensure compliance with organizational and legal requirements. This makes auditing a key part of security governance.

3. Methodology

The methodology used to complete this task was divided into structured phases to ensure systematic implementation and verification of auditing settings. Each phase includes a set of steps with specific goals and outcomes, including the configuration of auditing policies, log review, and identification of locked-out accounts. Screenshots can be inserted at each step (as PoC - Proof of Concept) to visually confirm successful execution.

3.1 Enable Audit Policy for Account Logon Events

Navigation Path:

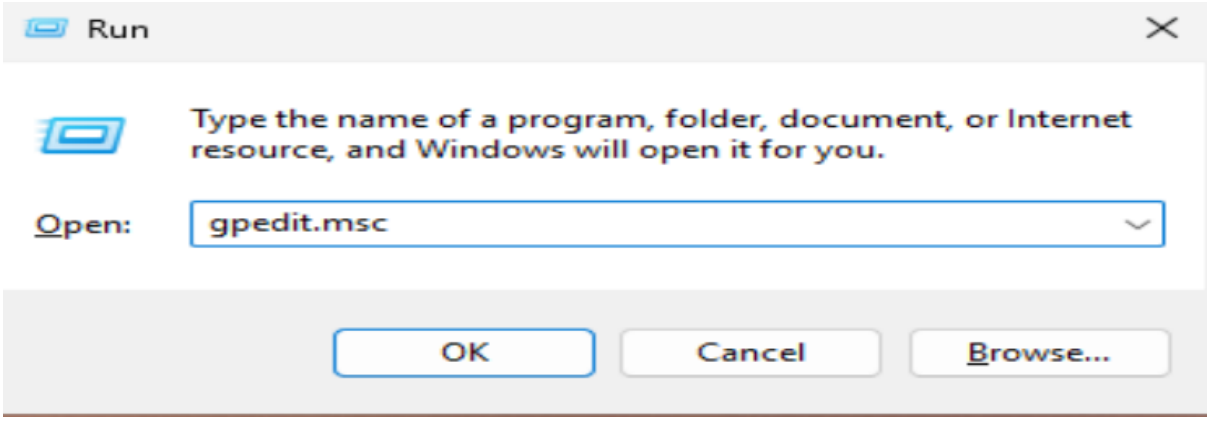
→ Start Menu → gpedit.msc

→ Local Computer Policy → Computer Configuration → Windows Settings

→ Security Settings → Local Policies → Audit Policy

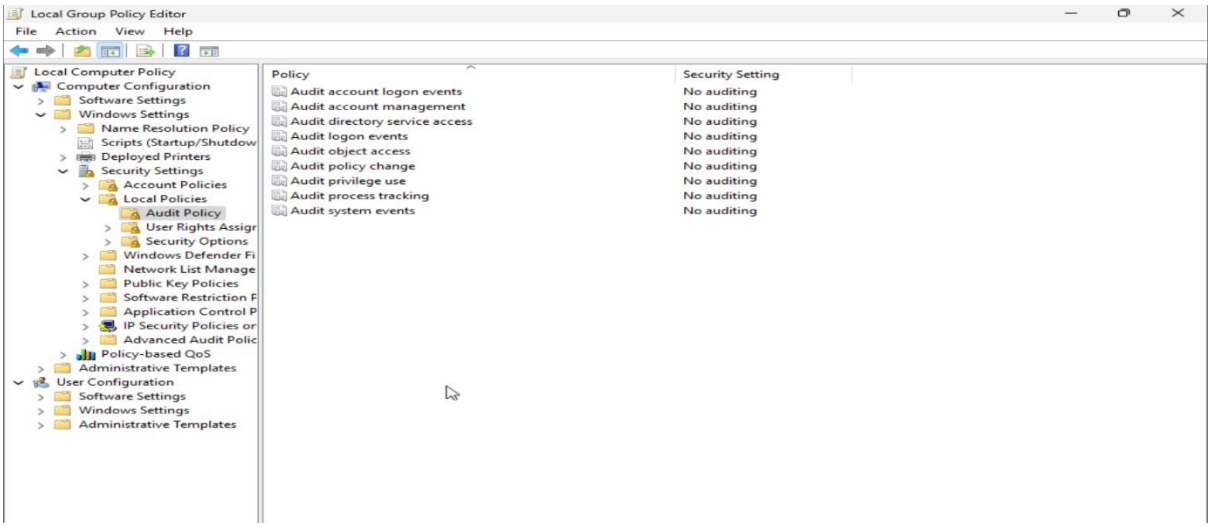
- **Open Local Group Policy Editor**
  Launch the Group Policy Editor by typing gpedit.msc in the Run dialog. This utility allows for configuring local security and audit policies.

  (Opening gpedit.msc.)



- **Navigate to Audit Policy Settings**
  Under Local Policies → Audit Policy, locate the "Audit logon events" option. This setting controls whether the system logs successful or failed login attempts.

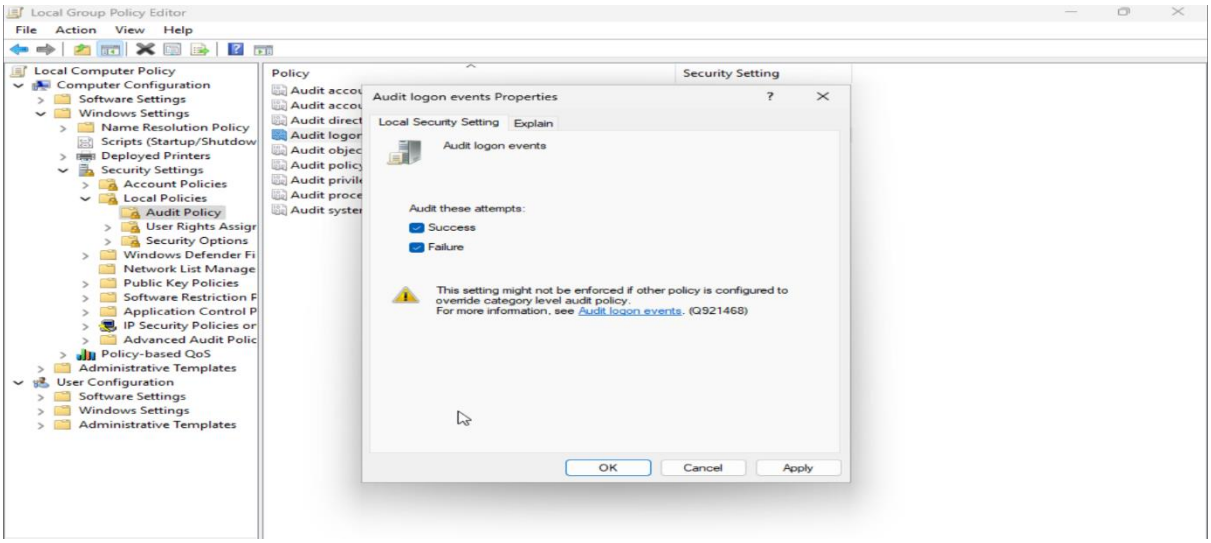
(Navigated to Audit Policy settings)



- **Enable Logon Event Auditing**

  Double-click "Audit logon events" and check both Success and Failure. This ensures all login activity is recorded. Click Apply and OK.

(Audit logon events policy enabled)



3.2 Check Event Viewer for Failed Login Attempts

Navigation Path:

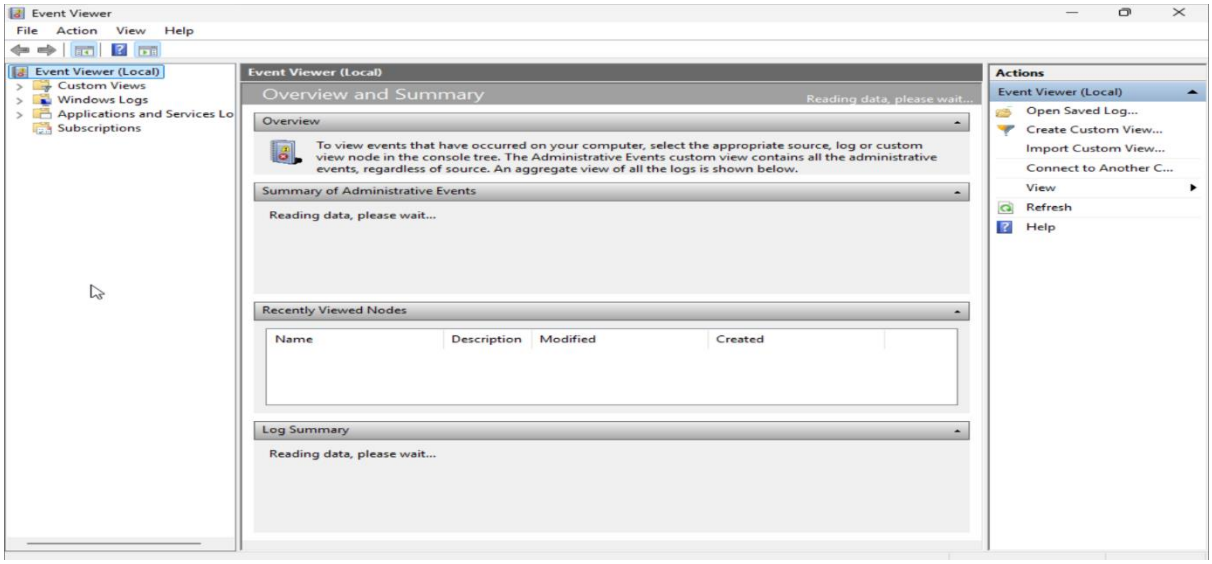
→ Start Menu → Event Viewer

→ Windows Logs → Security

- **Open Event Viewer**

  Start Event Viewer from the Start menu or Run dialog. This is used to view logs generated by the auditing system.

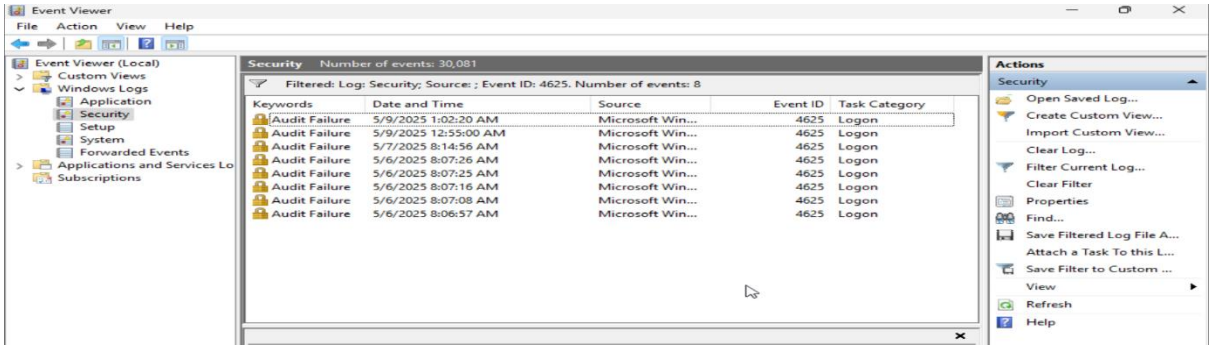
  (Event Viewer open on Security log.)



- **Filter for Failed Login Events (Event ID 4625)**

  Within the Security log, filter for **Event ID 4625**, which indicates a failed login attempt. This helps track brute-force attempts or incorrect credentials.

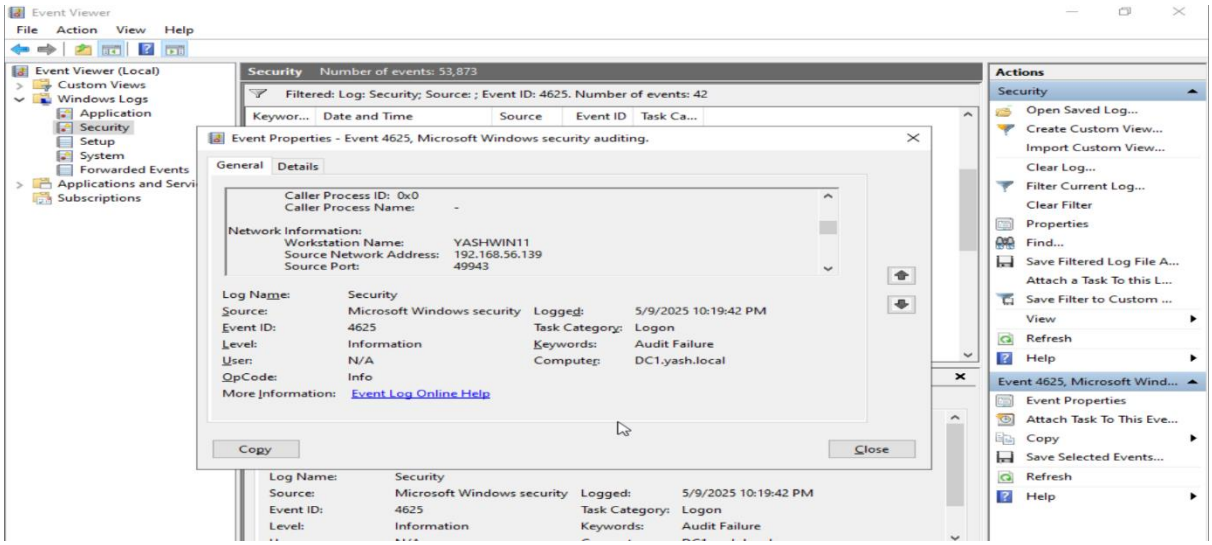
  (Filtered log showing Event ID 4625.)



- **Analyze the Event Details**

  Double-click the log entry to view detailed information, including username, IP address, and failure reason. This helps determine potential security risks.

(Detailed view of failed login event)



3.3 Detect Locked-Out Accounts Using PowerShell

Navigation Path:

→ Start Menu → gpedit.msc

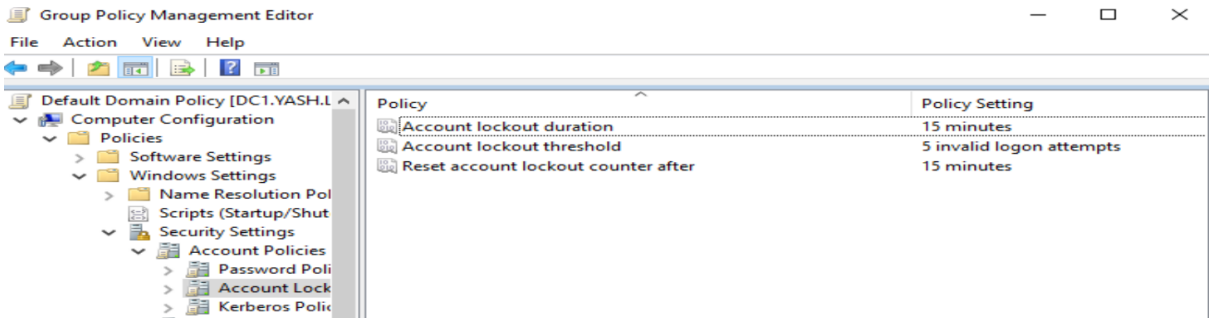
→ Local Computer Policy → Computer Configuration → Windows Settings

→ Security Settings → Account Policies → Account Lockout Policy

- **Apply Lockout Duration and Reset Counter**

  After setting the threshold, configure Account Lockout Duration and Reset Account Lockout Counter After. These values define how long the lockout lasts and when failed attempts reset.

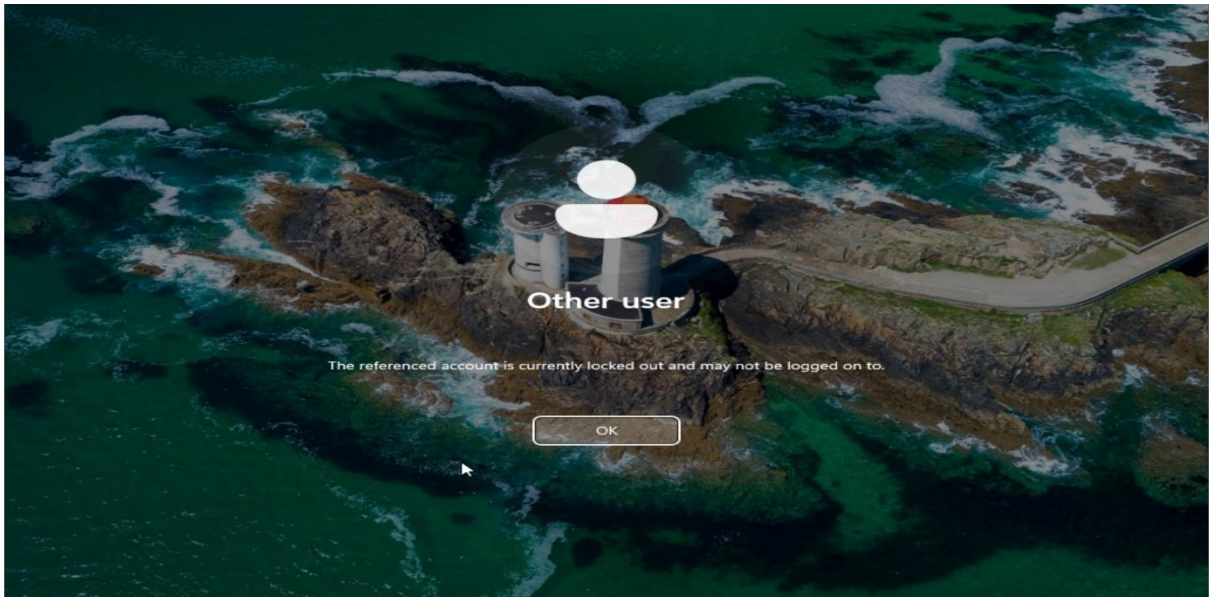
(Lockout duration and reset values configured.)



- **Trigger Lockout by Failed Logins**

  Intentionally enter incorrect passwords for a user account 5 times to trigger the lockout. A message will confirm the account is locked. This simulates real-world brute-force or user error.

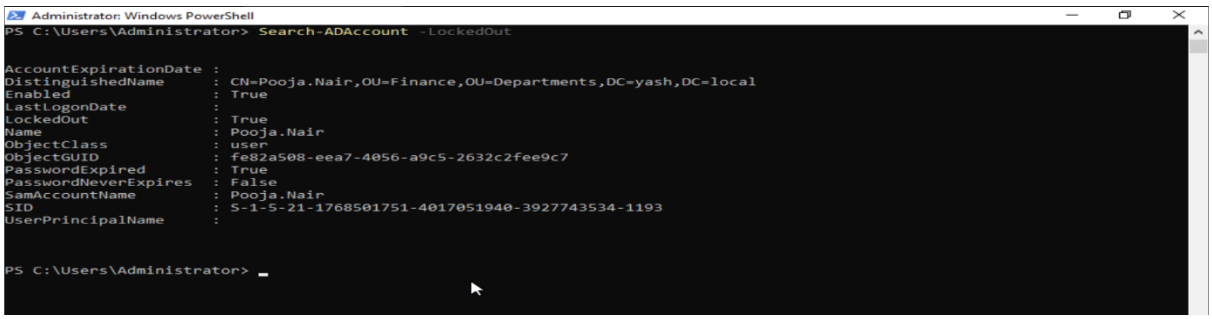
  (Account lockout message on login screen.)



- **Verify Locked-Out Accounts via PowerShell**

  Displays all accounts currently locked out due to policy. It confirms that the lockout policy is working as intended.

  (PowerShell output showing locked-out user.)



4. Results And Findings

After performing the auditing and account lockout configuration tasks, the system behavior was monitored using Event Viewer and PowerShell. The audit logs successfully captured both successful and failed login attempts. Lockout policies were tested and validated with controlled incorrect login attempts.

- **Audit Policy Successfully Enabled**
  The "Audit logon events" policy was correctly configured for both Success and Failure. This resulted in detailed logs being generated for each login attempt. The audit trail provided clear visibility into authentication activity.

- **Event ID 4625 Detected**
  Failed login attempts were recorded with Event ID 4625 in the Security log. These logs displayed critical information such as username, logon type, and failure reason. It confirmed the audit configuration was effective.

- **Lockout Policy Applied Correctly**
  The Account Lockout Threshold was set to 5 attempts and successfully triggered after simulated failed logins. This enforced the intended account security behavior. The user was locked out after exceeding the threshold.

- **PowerShell Detected Locked-Out Accounts**
  Using Search-ADAccount -LockedOut, the test user appeared as locked. This confirmed that domain controllers recognized and logged the lockout. It validated both auditing and policy enforcement.

- **Logs Exported and Documented**
  Audit logs were saved and included in the documentation. These files can be used for compliance, investigations, or future reviews. They also demonstrate full traceability of the process.

5. Recommendations

Based on the configuration and test results, several improvements and best practices are suggested to further strengthen the system's auditing and authentication posture. These recommendations aim to enhance detection, prevention, and accountability across the domain.

- **Apply Audit Policy at Domain Level**
  Instead of local policy (gpedit.msc), enforce audit settings via Group Policy Objects (GPOs) for consistency. This ensures all domain-joined systems follow the same audit standard. Centralized control also simplifies management.

- **Enable Advanced Audit Policies**
  Use advanced audit configuration for more granular event control (e.g., logon type, privilege use). This provides better insights into different authentication scenarios. It enhances forensic capabilities.

- **Set Realistic Lockout Durations**
  Fine-tune lockout duration and reset time to balance security and usability. Overly strict values may cause frustration or service disruptions. A duration of 15–30 minutes is commonly effective.

- **Regularly Monitor Security Logs**
  Schedule periodic reviews of Event Viewer logs or set up alerting systems. Proactive monitoring helps detect brute-force attacks or internal misuse early. This reduces incident response time.

- **Train Admins on Log Analysis**
  Ensure IT staff are trained to interpret event IDs and audit logs. Understanding patterns and anomalies improves response accuracy. It also supports faster investigations.

6. Conclusion

The tasks carried out in this exercise demonstrate a successful and structured approach to implementing key security controls within a Windows environment. Enabling auditing for account logon events ensures that both successful and failed authentication attempts are logged, creating a reliable audit trail critical for detecting misuse, brute-force attempts, or insider threats. The use of Event Viewer to analyze failed logon attempts, particularly with Event ID 4625, confirmed that the audit policies were applied correctly and provided valuable insights into user activity.

In parallel, the configuration of the account lockout policy strengthened access control by enforcing a threshold for invalid login attempts, which helps protect against common password-guessing and brute-force attacks. The process of triggering and detecting a locked-out account using PowerShell (Search-ADAccount -LockedOut) validated that domain policies were functioning correctly and that administrative tools can effectively identify and respond to account-related issues. These combined efforts not only reinforce user account protection but also support compliance, monitoring, and investigative readiness. Overall, the activity successfully established a more secure and auditable authentication environment, laying the groundwork for more advanced security practices in the future.