Alex Arnold

Professor Torres

Date you turned it in

IT 0115-41197

Lab Number

Part A: [Netlab v3Links to an external site.](https://netlabve3.flc.losrios.edu/" \t "_blank) (Any of the MCSA labs)

* configure advanced audit policy in group policy

A screenshot of a computer

AI-generated content may be incorrect.

* Add two domain users, log on to the Windows Client with both users

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close up of a logo

AI-generated content may be incorrect.

* Add an admin user, log on to the Domain Controller as the new admin

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Attempt the wrong password for all users, and attempt to log on as the wrong user
* A screenshot of a phone

  AI-generated content may be incorrect.
* Reboot the server and the client, and log back on to both systems

A blue square with a black background

AI-generated content may be incorrect.

* (do you see some differences in your logs)
* 4624 successful login, 4625 failed login, 4647 user logoff, 1102 log cleared, 4719 audit policy changed, 6005 enent log service started (reboot)
* **Create a Windows PowerShell or batch script (enable the same policies that you enabled in the previous step)**

function Set-NetAccountPolicy {

param (

[string]$Description,

[string]$Command,

[string]$VerifyMatch

)

try {

Write-Host "`n$Description..." -ForegroundColor Cyan

Invoke-Expression $Command

Start-Sleep -Seconds 1

net accounts | Select-String $VerifyMatch

Write-Host "$Description successfully applied." -ForegroundColor Green

} catch {

Write-Host "Failed to apply: $Description. Error: $\_" -ForegroundColor Red

}

}

function Set-AuditPolicy {

param (

[string]$Subcategory,

[string]$Success = "enable",

[string]$Failure = $null

)

try {

Write-Host "`nConfiguring audit policy: $Subcategory..." -ForegroundColor Cyan

$cmd = "AuditPol /set /subcategory:`"$Subcategory`" /success:$Success"

if ($Failure) {

$cmd += " /failure:$Failure"

}

Invoke-Expression $cmd

Start-Sleep -Seconds 1

AuditPol /get /subcategory:"$Subcategory"

Write-Host "Audit policy '$Subcategory' successfully set." -ForegroundColor Green

} catch {

Write-Host "Failed to set audit policy '$Subcategory'. Error: $\_" -ForegroundColor Red

}

}

# --- Password Policy ---

Set-NetAccountPolicy "Setting password history to 24 passwords" 'net accounts /uniquepw:24' "Password history length"

Set-NetAccountPolicy "Setting maximum password age to 60 days" 'net accounts /maxpwage:60' "Maximum password age"

Set-NetAccountPolicy "Setting minimum password age to 1 day" 'net accounts /minpwage:1' "Minimum password age"

Set-NetAccountPolicy "Setting minimum password length to 14 characters" 'net accounts /minpwlen:14' "Minimum password length"

# --- Audit Policies ---

Set-AuditPolicy "Credential Validation" "enable" "enable"

Set-AuditPolicy "Kerberos Authentication Service" "enable" "enable"

Set-AuditPolicy "Kerberos Service Ticket Operations" "enable" "enable"

Set-AuditPolicy "Application Group Management" "enable" "enable"

Set-AuditPolicy "Computer Account Management" "enable"

Set-AuditPolicy "User Account Management" "enable" "enable"

# --- Account Lockout Policies ---

Set-NetAccountPolicy "Setting account lockout duration to 15 minutes" 'net accounts /lockoutduration:15' "Lockout duration"

Set-NetAccountPolicy "Setting account lockout threshold to 5 attempts" 'net accounts /lockoutthreshold:5' "Lockout threshold"

Set-NetAccountPolicy "Setting reset account lockout counter after 15 minutes" 'net accounts /lockoutwindow:15' "Lockout observation window"

* **Search the internet for a script, or create a script that will put all logs onto a file for analysis (a text file)**

# PowerShell Script: Export All Domain Controller Logs into One File

# =========================================================

# Set output file

$outputDir = "C:\Logs"

$outputFile = Join-Path $outputDir "DomainController\_AllLogs.txt"

# Create output directory if it doesn't exist

if (-not (Test-Path -Path $outputDir)) {

New-Item -ItemType Directory -Path $outputDir | Out-Null

}

# List of logs to pull from

$logNames = @(

"System",

"Application",

"Security",

"Directory Service",

"DNS Server",

"File Replication Service"

)

# Clear output file if it already exists

if (Test-Path -Path $outputFile) {

Clear-Content -Path $outputFile

}

# Collect and append all logs

foreach ($logName in $logNames) {

try {

Write-Host "Exporting log: $logName" -ForegroundColor Cyan

Add-Content -Path $outputFile -Value ("`n===== BEGIN LOG: $logName =====`n")

Get-WinEvent -LogName $logName -ErrorAction SilentlyContinue |

Select-Object TimeCreated, Id, LevelDisplayName, Message |

ForEach-Object {

$entry = "[{0}] [{1}] [{2}] {3}" -f $\_.TimeCreated, $\_.Id, $\_.LevelDisplayName, $\_.Message

Add-Content -Path $outputFile -Value $entry

}

Add-Content -Path $outputFile -Value ("`n===== END LOG: $logName =====`n")

}

catch {

Write-Host "Failed to export $logName: $\_" -ForegroundColor Red

}

}

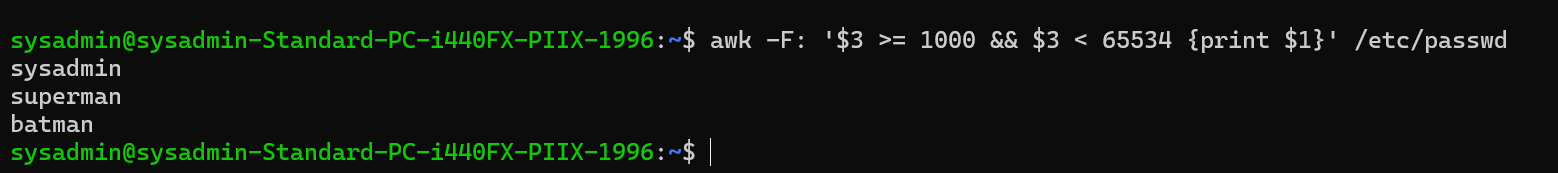
Write-Host "`nAll logs have been combined into $outputFile" -ForegroundColor Green

A screenshot of a computer

AI-generated content may be incorrect.

Part B: [Netlab v2Links to an external site.](https://netlabve2.flc.losrios.edu/" \t "_blank): NISGTS Linux+ Series 1 (any of the labs) (I used my own VM)

* Add two users to the Ubuntu box and log on as both users. Also, try the wrong creds, and try to log on as the wrong user.

A screenshot of a computer

AI-generated content may be incorrect.

* **Create a Linux bash script to copy all log files to one file for analysis**

#!/bin/bash

# Output file

output\_file="/home/sysadmin/Desktop/combined\_logs\_$(date +%F\_%T).log"

# Directories to search logs in

log\_dirs=("/var/log")

# Create or empty the output file

> "$output\_file"

# Loop through log directories and append logs

for dir in "${log\_dirs[@]}"; do

find "$dir" -type f -name "\*.log" -o -name "\*.gz" -o -name "\*.1" 2>/dev/null | while read -r log\_file; do

echo "===== $log\_file =====" >> "$output\_file"

# Decompress if .gz

if [[ "$log\_file" == \*.gz ]]; then

zcat "$log\_file" >> "$output\_file" 2>/dev/null

else

cat "$log\_file" >> "$output\_file" 2>/dev/null

fi

echo -e "\n\n" >> "$output\_file"

done

done

echo "Logs collected in: $output\_file"

A white square with a black text on it

AI-generated content may be incorrect.

Part C:Netlab v2 for Linux, v3 for Windows

* Now that you have your logs for both Windows and Linux. We need to parse the logs. Answer the below questions:
  + On your Windows log file
    - via CLI or PowerShell
      * List any errors
        + Events dropped by transport
      * List any warnings
      * Failed logon
      * System reboot unexpected
      * List authenticated users
        + Batman
        + Superman
        + Wonderwoman
      * List blocked users
      * Batman
      * Superman
      * thefalsh
  + In your Linux log file
    - via CLI or with a Bash script

#!/bin/bash

# Path to combined log file (from collect\_logs.sh)

combined\_log="/home/sysadmin/Desktop/combined\_logs\_\*.log"

# Use the most recent combined log

latest\_log=$(ls -t $combined\_log 2>/dev/null | head -n 1)

# Output file

output\_file="/home/sysadmin/Desktop/combined\_log\_analysis\_$(date +%F\_%T).log"

> "$output\_file"

if [[ -z "$latest\_log" ]]; then

echo "No combined log file found in /tmp. Please run collect\_logs.sh first."

exit 1

fi

echo "Analyzing: $latest\_log" >> "$output\_file"

echo "=== Network Card Details (from current system) ===" >> "$output\_file"

ip a >> "$output\_file" 2>&1

echo -e "\n\n" >> "$output\_file"

echo "=== Authenticated Users (Accepted logins) ===" >> "$output\_file"

grep -i "Accepted" "$latest\_log" >> "$output\_file"

echo -e "\n\n" >> "$output\_file"

echo "=== Blocked Users (Failed logins) ===" >> "$output\_file"

grep -i "Failed" "$latest\_log" >> "$output\_file"

echo -e "\n\n" >> "$output\_file"

echo "Analysis complete. Output saved to: $output\_file"

echo "Log analysis complete. Output saved to: $output\_file"

* + - * List network card details

A screenshot of a computer program

AI-generated content may be incorrect.

* + - * List authenticated users

A close-up of a number

AI-generated content may be incorrect.

* + - * List blocked users

A screenshot of a computer code

AI-generated content may be incorrect.  
  
A screenshot of a computer

AI-generated content may be incorrect.

It took a few attempts to get it correct