

STIG Implementation Report

- **Intern Credit Application For:** Bruce Thornton
Date: 11/12/2025
STIG Finding: WN11-CC-000170
 - **SRG:** [SRG-OS-000480-GPOS-00227](#)
Severity: medium
Vulnerability ID: V-253384 **CCI:** CCI-000366
-

1. Introduction

This report documents the process of identifying, remediating, and verifying the fix for a Windows 11 STIG compliance finding. The selected finding was: STIG ID: WN11-CC-000170 “The setting to allow Microsoft accounts to be optional for modern style apps must be enabled.”

2. Initial Scan Results

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000170
- Status: **Failed** (non-compliant)



Evidence: First identified the STIG:

<https://stigaview.com/products/win11/v2r2/WN11-CC-000170/>

Initial scan result:

The screenshot shows the Tenable Vulnerability Management interface. At the top, it says "Vulnerability Management" and "Scans > Scan Details > Audit Details". A banner at the top indicates "WN11-CC-000170 - The setting to allow Microsoft accounts to be optional for modern style apps must be enabled." The status is listed as "AUDIT FAILED". Below this, there are tabs for "Overview" and "Assets", with "Assets" being the active tab. A search bar shows "Search" and "1 Results". The main table has one result, with columns for "STATUS" (FAILED) and "NAME" (10.1.0.176). To the right of the table, there are sections for "Solution" (Configure the policy value for Computer Configuration >> Administrative Templates >> Windows Components >> App Runtime >> "Allow Microsoft accounts to be optional" to "Enabled"), "See Also" (link to https://dl.dod.cyber.mil/wp-content/uploads/stigs/zips/U_MS_Windows_11_V2R4_STIG.zip), and "Reference Information" (links to various standards like 800-171R3, 3.4.2, 800-53, CM-6b, CAT III, CN-L3, 8.1.10.6(d), CSF2.0, DE.CM-09, PR.PS-01, CCI, CCI-000366, CSF, PR.IP-1, DISA_BENCHMARK, and Microsoft_Windows_11_ST).

3. Manual Remediation Steps

Ran gpedit.msc:

Configure the policy value for Computer Configuration >> Administrative Templates >> Windows Components >> App Runtime >> "Allow Microsoft accounts to be optional" to "Enabled".

Run "gpupdate /force" and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000170
- Status: **Passed**

Evidence:

Vulns by Plugin Audits Vulns by Asset History

WN11-CC-000170 1 Results

0 Failed 0 Warning 1 Passed

1 Item

STATUS	NAME	FAMILY	COUNT
Passed	WN11-CC-000170 - The setting to allow Microsoft accounts to be...	Windows Compliance Checks	1

Scan Details

STATUS	Completed
START TIME	11/12/2025 at 2:19 PM
TEMPLATE	Advanced Network Scan
SCANNER	LOCAL-SCAN-ENGINE-01
TARGETS	10.1.0.176

4. Reintroduction of Finding (Manually Undo Test)

To demonstrate full control of the setting, the fix was undone:

- Disabled the setting. Open Group Policy Management “gpedit.msc” and followed the instructions for remediation from before and set it to the original setting: “Not Configured”
- Ran “gpupdate /force” and rescanned.
- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000170

Status: **Failed**, Non-Compliant

Evidence:

The screenshot shows the Tenable Vulnerability Management interface. At the top, it says "Vulnerability Management" and "Scans > Scan Details". The scan name is "Win11BruceNov12DS". On the left, there are tabs for "Vuuls by Plugin", "Audits" (which is selected), "Vuuls by Asset", and "History". The search bar shows "WN11-CC-000170". The results table has one row: "1 Failed", "0 Warning", "0 Passed". The table columns are "STATUS", "NAME", "FAMILY", and "COUNT". The "NAME" column contains the description of the vulnerability. To the right, there are sections for "Scan Details" (Status: Completed, Start Time: 11/12/2025 at 2:35 PM, Template: Advanced Network Scan, Scanner: LOCAL-SCAN-ENGINE-01) and "TARGETS" (IP: 10.1.0.176). On the far right, there are counts for Critical, High, Medium, and Low vulnerabilities.

5. Remediation with PowerShell Script

Save as: Remediate-WN11-CC-000170.ps1 and run as **Administrator** utilizing PowerShell ISE:

The screenshot shows the Windows PowerShell ISE window titled "Administrator: Windows PowerShell ISE". The script file "Remediate-WN11-CC-000170.ps1" is open, containing the following PowerShell code:

```
34 # Evidence object
35 $pscustomobject@{
36     ComputerName = $env:COMPUTERNAME
37     STIG_ID = $stigid
38     SettingName = "Allow Microsoft accounts to be optional (App Runtime)"
39     RegistryPath = "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System"
40     ValueName = $valueName
41     RequiredValue = $desiredValue
42     ActualValue = $actualValue
43     ComplianceStatus = $compliant
44     ActionTaken = $action
45     Timestamp = $getdate
46 } | Format-List
```

The output pane shows the results of running the script:

```
ComputerName : Win11VMBruce
STIG_ID : WN11-CC-000170
SettingName : Allow Microsoft accounts to be optional (App Runtime)
RegistryPath : HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
ValueName : MSADOptional
RequiredValue : 1
ActualValue : 1
ComplianceStatus : Compliant
ActionTaken : Created MSAOptional and set to 1.
Timestamp : 11/12/2025 8:50:46 PM
```

A status bar at the bottom indicates "Completed". A small screenshot of the Windows Taskbar is visible in the bottom right corner.

Script Used:

```
<#
STIG ID : WN11-CC-000170
Title : The setting to allow Microsoft accounts to be optional for modern style apps must be
enabled.
```

Check/Fix mapping:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
Value: MSAOptional (REG_DWORD) = 1 # Enabled
#>

$regPath    = 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System'
$valueName   = 'MSAOptional'
$desiredValue = 1
$stigId     = 'WN11-CC-000170'

# Ensure key exists
if (-not (Test-Path $regPath)) { New-Item -Path $regPath -Force | Out-Null }

# Set value if needed
$current = Get-ItemProperty -Path $regPath -Name $valueName -ErrorAction SilentlyContinue
if ($null -eq $current) {
    New-ItemProperty -Path $regPath -Name $valueName -PropertyType DWord -Value
$desiredValue -Force | Out-Null
    $action = "Created $valueName and set to $desiredValue."
} elseif ($current.$valueName -ne $desiredValue) {
    Set-ItemProperty -Path $regPath -Name $valueName -Value $desiredValue -Type DWord
    $action = "Updated $valueName from $($current.$valueName) to $desiredValue."
} else {
    $action = "$valueName already set to $desiredValue. No change needed."
}

# Verify
$actual = (Get-ItemProperty -Path $regPath -Name $valueName -ErrorAction
SilentlyContinue).$valueName
$compliant = if ($actual -eq $desiredValue) {'Compliant'} else {'Non-Compliant'}

# Evidence object
[pscustomobject]@{
    ComputerName    = $env:COMPUTERNAME
    STIG_ID         = $stigId
    SettingName     = 'Allow Microsoft accounts to be optional (App Runtime)'
    RegistryPath    = 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System'
    ValueName       = $valueName
    RequiredValue   = $desiredValue
```

```

ActualValue      = $actual
ComplianceStatus = $compliant
ActionTaken      = $action
Timestamp        = Get-Date
} | Format-List *

```

Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000170
- Status: **Passed**

Evidence:

The screenshot shows the Tenable Vulnerability Management interface. At the top, it says "Scans > Scan Details". Below that, the scan name is "Win11BruceNov12DS". The audit tab is selected, showing 1 result for "WN11-CC-000170". The finding details are as follows:

STATUS	NAME	FAMILY	COUNT
Passed	WN11-CC-000170 - The setting to allow Microsoft accounts to be...	Windows Compliance Checks	1

On the right side, there is a summary of vulnerabilities:

- 0 Critical Vulnerabilities
- 0 High Vulnerabilities
- 0 Medium Vulnerabilities
- 0 Low Vulnerabilities

Scan Details:

STATUS	Completed
START TIME	11/12/2025 at 2:57 PM
TEMPLATE	Advanced Network Scan
SCANNER	LOCAL-SCAN-ENGINE-01
TARGETS	10.1.0.176

6. Conclusion

The finding **WN11-CC-000170** was successfully:

- Detected in an initial Tenable STIG Audit scan,

- Remediated manually,
- Verified through a second scan,
- Undone and confirmed as vulnerable again,
- Finally re-applied through PowerShell automation, and validated with a third scan.

This demonstrates the ability to manage Windows STIG compliance both manually and through PowerShell automation.