STIG Implementation Report

- **Intern Credit Application For:** Bruce Thornton
  Date: 11/2/2025
  STIG Finding: WN11-SO-000025
- **SRG:** SRG-OS-000480-GPOS-00227
  Severity: medium
  Vulnerability ID: V-253436 **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-SO-000025 “The built-in guest account must be renamed.”

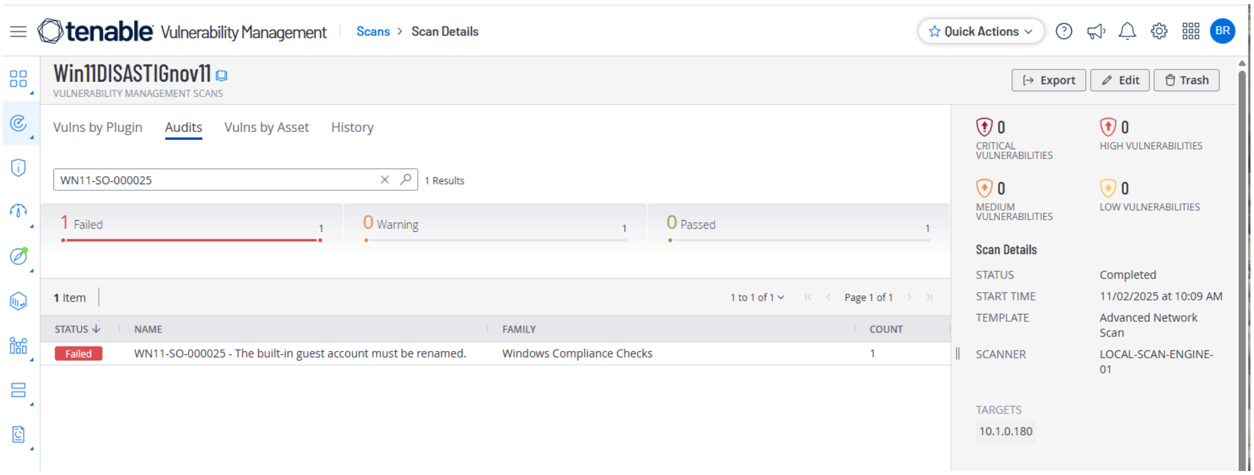
2. Initial Scan Results

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-SO-000025
- Status: **Failed** (non-compliant)

📎 **Evidence:** First identified the STIG:

https://stigaview.com/products/win11/v1r6/WN11-SO-000025/

Initial scan result:



Win11DISASTIGnov11
VULNERABILITY MANAGEMENT SCANS

WN11-SO-000025 — 1 Results

1 Failed 1 | 0 Warning 1 | 0 Passed 1

1 Item — 1 to 1 of 1 — Page 1 of 1

STATUS	NAME	FAMILY	COUNT
Failed	WN11-SO-000025 - The built-in guest account must be renamed.	Windows Compliance Checks	1

0 CRITICAL VULNERABILITIES
0 HIGH VULNERABILITIES
0 MEDIUM VULNERABILITIES
0 LOW VULNERABILITIES

Scan Details

STATUS	Completed
START TIME	11/02/2025 at 10:09 AM
TEMPLATE	Advanced Network Scan
SCANNER	LOCAL-SCAN-ENGINE-01

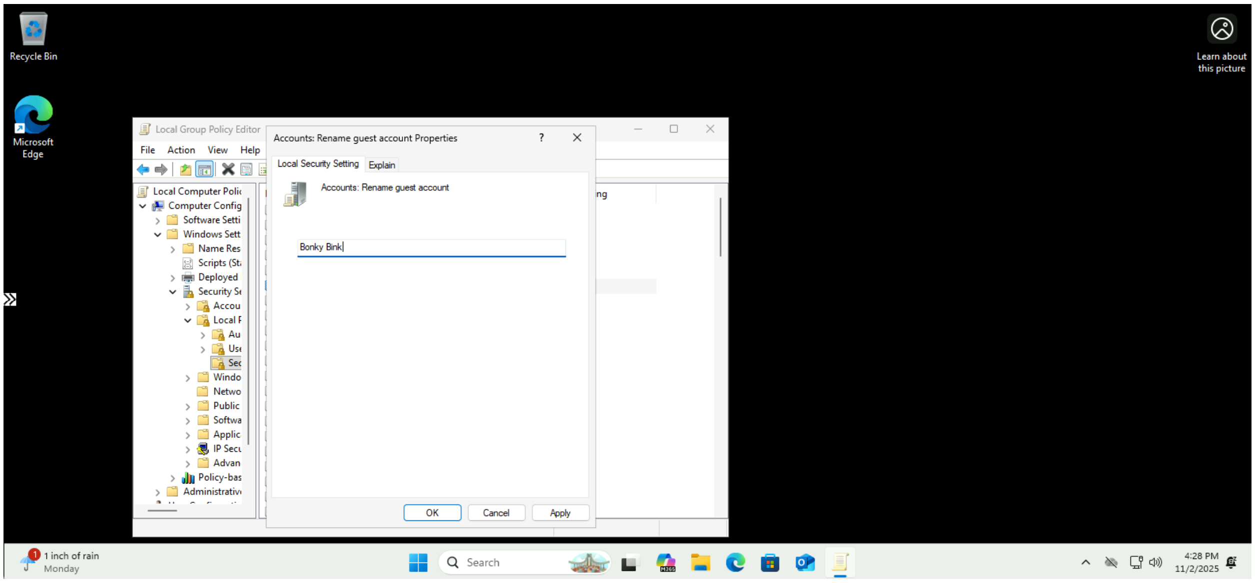
TARGETS
10.1.0.180

---

3. Manual Remediation Steps

Run “gpedit.msc”

Configure the policy value for Computer Configuration >> Windows Settings >> Security Settings >> Local Policies >> Security Options >> "Accounts: Rename guest account" to a name other than "Guest".

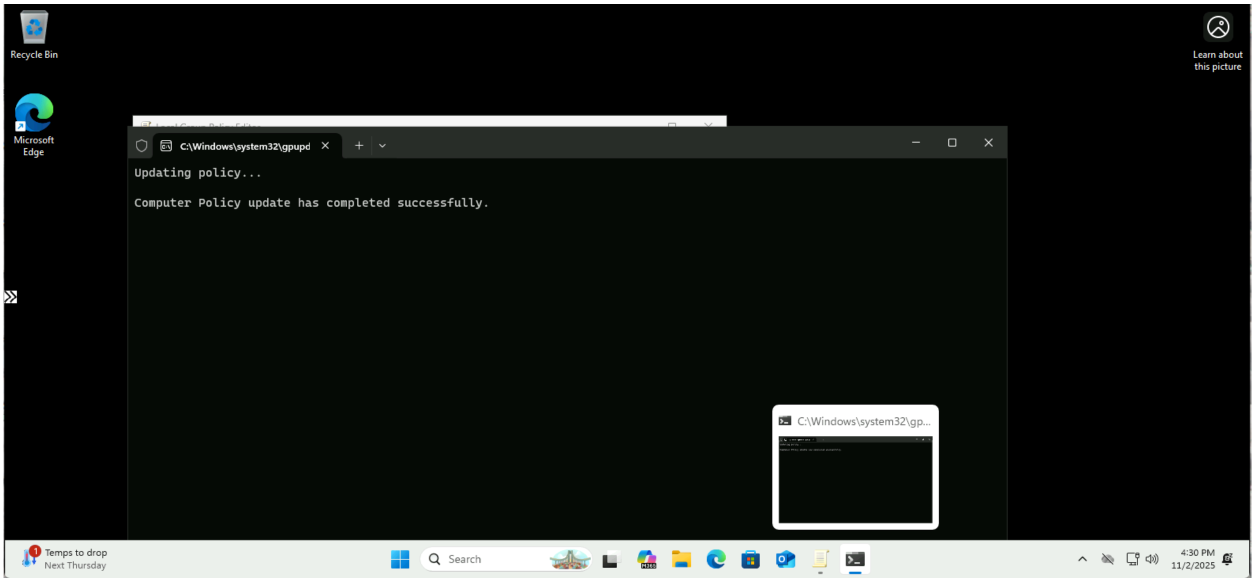


Accounts: Rename guest account Properties

Accounts: Rename guest account

Bonky Bink

Because the “Rename” isn’t specified, I changed it to “Bonky Bink.”



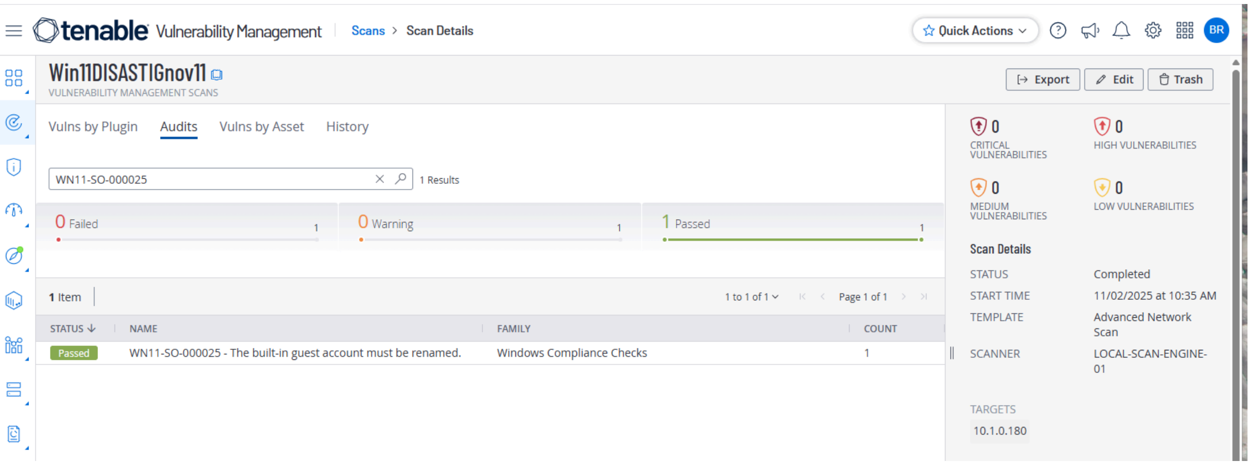
```
Updating policy...

Computer Policy update has completed successfully.
```

Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-SO-000025
- Status: **Passed**

Evidence:



Win11DISASTIGnov11
VULNERABILITY MANAGEMENT SCANS

Search: WN11-SO-000025 — 1 Results

0 Failed 1 | 0 Warning 1 | 1 Passed 1

1 Item

STATUS	NAME	FAMILY	COUNT
Passed	WN11-SO-000025 - The built-in guest account must be renamed.	Windows Compliance Checks	1

0 CRITICAL VULNERABILITIES
0 HIGH VULNERABILITIES
0 MEDIUM VULNERABILITIES
0 LOW VULNERABILITIES

Scan Details

STATUS	Completed
START TIME	11/02/2025 at 10:35 AM
TEMPLATE	Advanced Network Scan
SCANNER	LOCAL-SCAN-ENGINE-01

TARGETS
10.1.0.180

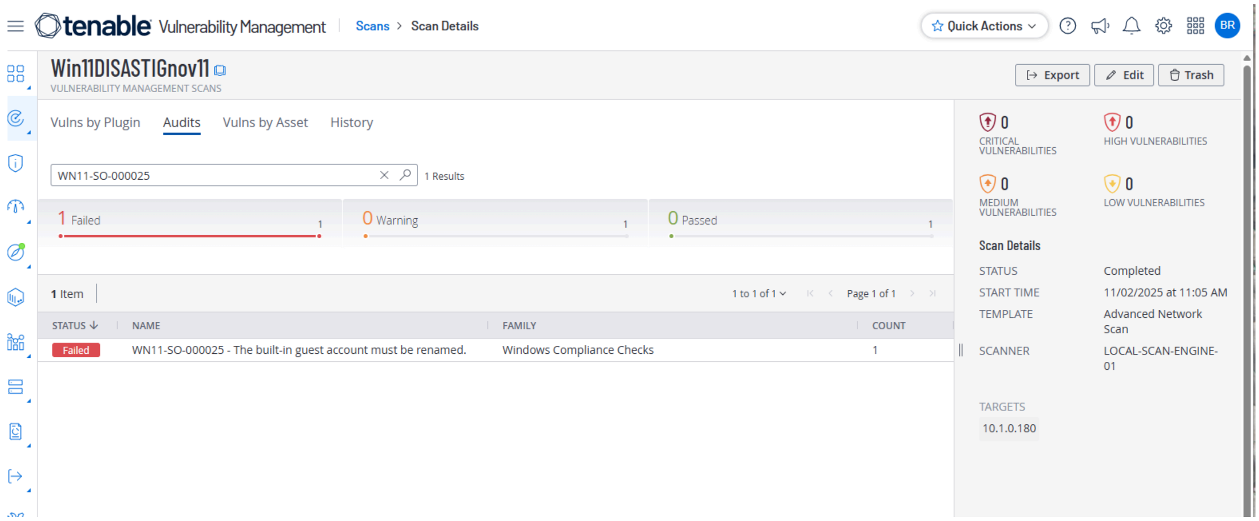
4. Reintroduction of Finding (Manually Undo Test)

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: “Guest”
- Ran “`gpupdate /force`” and rescanned.

Status: **Failed**, Non-Compliant

Evidence:



Win11DISASTIGnov11
VULNERABILITY MANAGEMENT SCANS

WN11-SO-000025 — 1 Results

1 Failed 1 | 0 Warning 1 | 0 Passed 1

1 Item — 1 to 1 of 1 — Page 1 of 1

STATUS	NAME	FAMILY	COUNT
Failed	WN11-SO-000025 - The built-in guest account must be renamed.	Windows Compliance Checks	1

0 CRITICAL VULNERABILITIES	0 HIGH VULNERABILITIES
0 MEDIUM VULNERABILITIES	0 LOW VULNERABILITIES

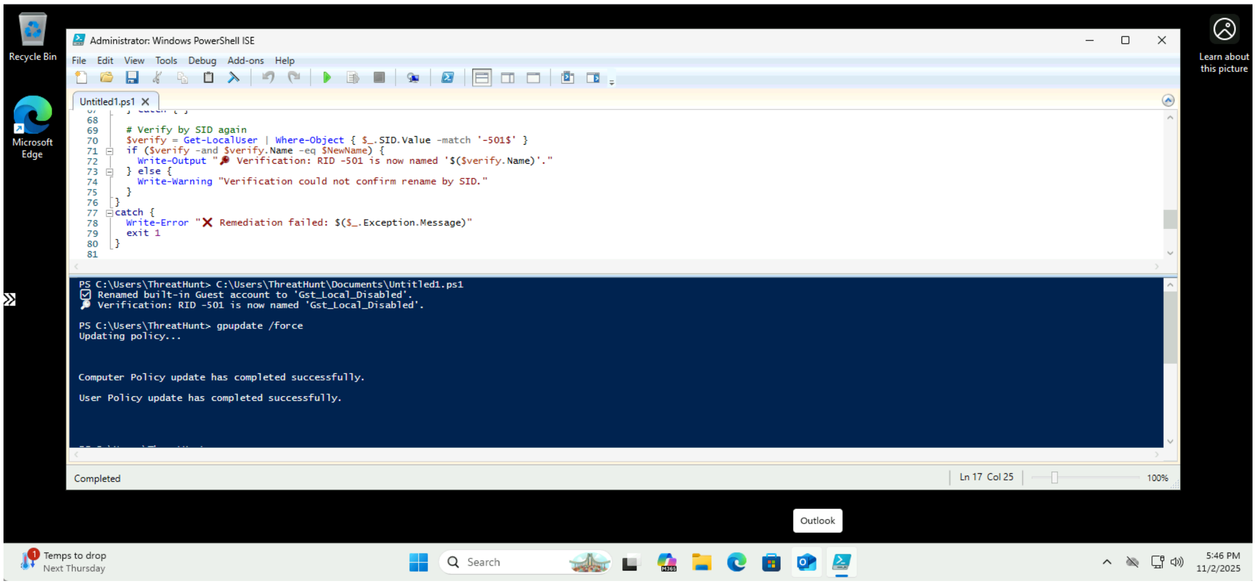
Scan Details

STATUS	Completed
START TIME	11/02/2025 at 11:05 AM
TEMPLATE	Advanced Network Scan
SCANNER	LOCAL-SCAN-ENGINE-01

TARGETS
10.1.0.180

5. Remediation with PowerShell Script

Save as: `Remediate-WN11-SO-000025.ps1` and run **as Administrator** utilizing PowerShell ISE:



Recycle Bin
Microsoft Edge
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1
Learn about this picture

```
    # Verify by SID again
    $verify = Get-LocalUser | Where-Object { $_.SID.Value -match '-501$' }
    if ($verify -and $verify.Name -eq $NewName) {
      Write-Output "🔎 Verification: RID -501 is now named '$($verify.Name)'."
    } else {
      Write-Warning "Verification could not confirm rename by SID."
    }
}
catch {
    Write-Error "❌ Remediation failed: $($_.Exception.Message)"
    exit 1
}
```

```
PS C:\Users\ThreatHunt> C:\Users\ThreatHunt\Documents\Untitled1.ps1
☑ Renamed built-in Guest account to 'Gst_Local_Disabled'.
🔎 Verification: RID -501 is now named 'Gst_Local_Disabled'.

PS C:\Users\ThreatHunt> gpupdate /force
Updating policy...



Computer Policy update has completed successfully.

User Policy update has completed successfully.
```

Script:

<#
.SYNOPSIS
  Remediates STIG WN11-SO-000025:
  Renames the built-in Guest account (RID ...-501).

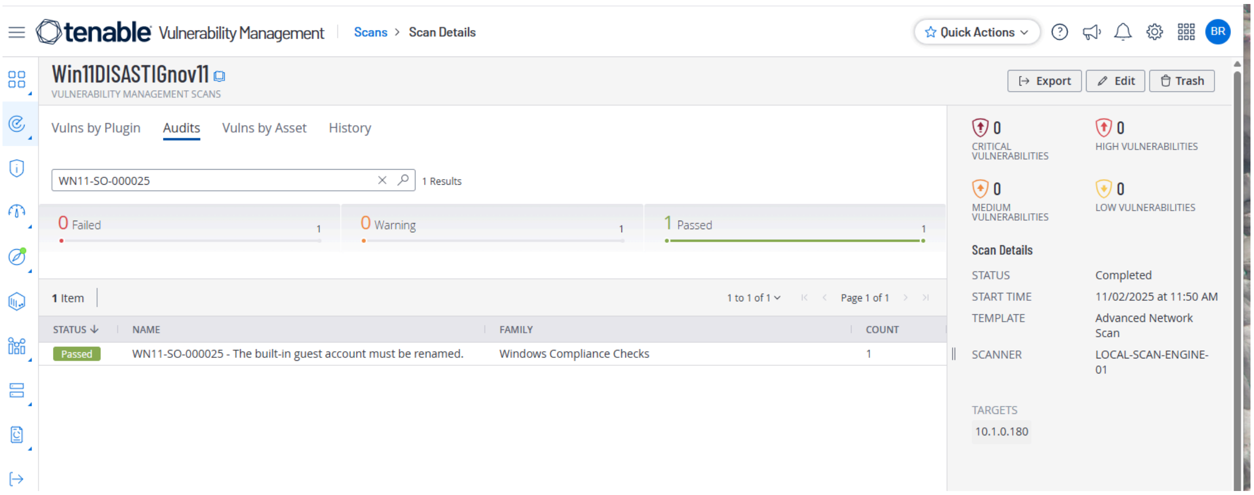
.DESCRIPTION
  Locates the local account whose SID ends with -501 and renames it to a specified value.
  If the account is already renamed (not "Guest"), the script confirms compliance and exits.

.PARAMETER NewName
  The new name for the built-in Guest account (default: "Gst_Local_Disabled").

.NOTES
  - Run as Administrator.
  - Works on standalone and domain-joined systems (targets LOCAL SAM).
  - Keep the name ≤ 20 characters (SAM limit) and avoid leading/trailing spaces.
#>

```
[CmdletBinding(SupportsShouldProcess=$true)]
param(
  [Parameter(Mandatory=$false)]
  [ValidateLength(1,20)]
  [ValidatePattern('^[^\s].*[^\s]$')]
  [string]$NewName = 'Gst_Local_Disabled'
)

# Require admin
if (-not ([Security.Principal.WindowsPrincipal] [Security.Principal.WindowsIdentity]::GetCurrent()
  ).IsInRole([Security.Principal.WindowsBuiltinRole] "Administrator")) {
  Write-Error "Run this script as Administrator."
  exit 1
}

# Ensure LocalAccounts module is available (PowerShell 5+)
if (-not (Get-Command Get-LocalUser -ErrorAction SilentlyContinue)) {
  Write-Error "Get-LocalUser / Rename-LocalUser cmdlets not found (PowerShell 5+ required)."
  exit 1
}

try {
  # Find the built-in Guest by RID -501
  $guestAcct = Get-LocalUser | Where-Object { $_.SID.Value -match '-501$' }

  if (-not $guestAcct) {
    throw "Built-in Guest account (RID -501) not found."
  }

  # Already renamed?
  if ($guestAcct.Name -ne 'Guest') {
    Write-Output "✅ Guest account already renamed: '$($guestAcct.Name)'. No action needed."
    return
  }

  # Check for name collision
  if (Get-LocalUser -Name $NewName -ErrorAction SilentlyContinue) {
    throw "The name '$NewName' is already in use. Choose a different NewName."
  }

  if ($PSCmdlet.ShouldProcess("Guest (RID -501)", "Rename to '$NewName'")) {
    Rename-LocalUser -Name 'Guest' -NewName $NewName
    Write-Output "✅ Renamed built-in Guest account to '$NewName'."
  }
  # Optional: keep Guest disabled (many environments also require it disabled)
  try {
    Disable-LocalUser -Name $NewName -ErrorAction SilentlyContinue | Out-Null
  } catch { }

  # Verify by SID again
  $verify = Get-LocalUser | Where-Object { $_.SID.Value -match '-501$' }
  if ($verify -and $verify.Name -eq $NewName) {
    Write-Output "🔎 Verification: RID -501 is now named '$($verify.Name)'."
  } else {
    Write-Warning "Verification could not confirm rename by SID."
  }
}
catch {
  Write-Error "❌ Remediation failed: $($_.Exception.Message)"
  exit 1
}
```

Evidence:



Win11DISASTIGnov11

VULNERABILITY MANAGEMENT SCANS

WN11-SO-000025 1 Results

0 Failed 1 | 0 Warning 1 | 1 Passed 1

1 Item | 1 to 1 of 1 | Page 1 of 1

STATUS	NAME	FAMILY	COUNT
Passed	WN11-SO-000025 - The built-in guest account must be renamed.	Windows Compliance Checks	1

0 CRITICAL VULNERABILITIES
0 HIGH VULNERABILITIES
0 MEDIUM VULNERABILITIES
0 LOW VULNERABILITIES

Scan Details

STATUS Completed
START TIME 11/02/2025 at 11:50 AM
TEMPLATE Advanced Network Scan
SCANNER LOCAL-SCAN-ENGINE-01

TARGETS
10.1.0.180

What it changes: Renames the local account whose SID ends in -501 (built-in Guest).

Why SID-based: Ensures you target the built-in Guest even if it was previously renamed.

Scanner expectations: Tenable typically checks that the RID -501 account's name ≠ “Guest.”

Optional hardening: Many STIG baselines also keep the Guest disabled; the script tries to disable after rename (no harm if already disabled).

6. Conclusion

The finding **WN11-SO-000025** 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.