

STIG Implementation Report

- **Intern Credit Application For:** Bruce Thornton
Date: 11/11/2025
STIG Finding: WN11-CC-000391
 - **SRG:** [SRG-OS-000185-GPOS-00079](#)
Severity: medium
Vulnerability ID: V-256893 **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-000391 “Internet Explorer must be disabled for Windows 11.”

2. Initial Scan Results

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000391
- Status: **Failed** (non-compliant)



Evidence: First identified the STIG:

<https://stigaview.com/products/win11/v1r5/WN11-CC-000391/>

Initial scan result:

The screenshot shows the Tenable Vulnerability Management interface. At the top, it displays the navigation path: Vulnerability Management > Scans > Scan Details > Audit Details. The main title is "WN11-CC-000391 - Internet Explorer must be disabled for Windows 11." Below this, there are two tabs: "Overview" and "Assets", with "Assets" being the active tab. A search bar and a results count of "1 Results" are also present. The main content area shows a single result row with columns for STATUS (FAILED) and NAME (10.1.0.118). On the right side, there is a "Solution" section with a detailed description for Windows 11 semi-annual channel users. Below the solution is a "See Also" link to a specific STIG document. Further down is a "Reference Information" section containing various policy IDs and their corresponding details.

3. Manual Remediation Steps

For Windows 11 semi-annual channel, remove or disable the IE11 application.

To disable IE11 as a standalone browser:

Run “gpedit.msc”

Set the policy value for "Computer Configuration/Administrative Templates/Windows Components/Internet Explorer/Disable Internet Explorer 11 as a standalone browser" to "Enabled" with the option value set to "Never".

Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000391
- Status: **Passed**

Evidence:

The screenshot shows the Tenable Vulnerability Management interface. The top navigation bar includes 'Scans' and 'Scan Details'. The main content area displays a scan titled 'Win11DSBruceNov11'. The 'Audits' tab is selected, showing 0 Failed, 0 Warning, and 1 Passed results. A table below lists the single passed finding: 'WN11-CC-000391 - Internet Explorer must be disabled for Wind...' under the 'Windows Compliance Checks' family. The right side of the screen provides 'Scan Details' including status (Completed), start time (11/11/2025 at 10:48 AM), template (Advanced Network Scan), scanner (LOCAL-SCAN-ENGINE-01), and target (10.1.0.118).

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-000391

Status: **Failed**, Non-Compliant

Evidence:

The screenshot shows the Tenable Vulnerability Management interface. The scan name is "Win11DSBruceNov11". The status bar indicates "1 Results". The results table shows one item: "WN11-CC-000391 - Internet Explorer must be disabled for Wind..." under the "Windows Compliance Checks" family. The status is "Failed". The right panel displays "Scan Details" including the status "Completed", start time "11/11/2025 at 11:20 AM", template "Advanced Network Scan", scanner "LOCAL-SCAN-ENGINE-01", and target "10.1.0.118".

5. Remediation with PowerShell Script

Save as: Remediate-WN11-CC-000391.ps1 and run **as Administrator** utilizing PowerShell ISE:

The screenshot shows the Windows PowerShell ISE window. The script file "Remediate-WN11-CC-000391.ps1" contains the following PowerShell code:

```
1 $stig
2 $STIG_ID = 'WN11-CC-000391'
3 $Title = 'Internet Explorer must be disabled for Windows 11.'
4
5 Technical implementation for compliance:
6 1. IE11 feature removed or disabled (where present).
7 2. Policy 'Disable Internet Explorer 11 as a standalone browser' set to 'Enabled'
8 with option 'Never', which corresponds to the registry value:
9 [HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Main\NotifyDisableIEOptions] (REG_DWORD) = 0
10 #>
11 $stigid = 'WN11-CC-000391'
12
13 #>
```

The output pane shows the execution results:

```
PS C:\Users\ThreatHunt> C:\Users\ThreatHunt\Desktop\Remediate-WN11-CC-000391.ps1
```

ComputerName	STIG_ID	SettingName	RegistryPath	KeyValue	RequiredValue	ActualValue	IE_Feature_Action	Remarks	ComplianceStatus	Timestamp
Win11DSBruce111	WN11-CC-000391	Disable Internet Explorer 11 as a standalone browser	HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Internet Explorer\Main	NotifyDisableIEOptions	0	0	Not applicable (feature not found).	Required value was tested. NotifyDisableIEOptions and set to 0.	Compliant	11/11/2025 6:10:59 PM

The status bar at the bottom indicates "Completed".

Script Used:

<#

STIG ID : WN11-CC-000391

Title : Internet Explorer must be disabled for Windows 11.

Technical implementation for compliance:

1. IE11 feature removed or disabled (where present).
2. Policy 'Disable Internet Explorer 11 as a standalone browser' set to 'Enabled'

with option 'Never', which corresponds to the registry value:

HKLM\SOFTWARE\Policies\Microsoft\Internet Explorer>Main\NotifyDisableIEOptions
(REG_DWORD) = 0

#>

```
$stigId = 'WN11-CC-000391'
```

```
# -----
```

```
# 1) Disable IE11 feature if present
```

```
# -----
```

```
$featureName = 'Internet-Explorer-Optional-amd64'
```

```
$feature = Get-WindowsOptionalFeature -Online -FeatureName $featureName -ErrorAction  
SilentlyContinue
```

```
$featureAction = 'Not applicable (feature not found).'
```

```
if ($feature) {
```

```
    if ($feature.State -ne 'Disabled') {
```

```
        Disable-WindowsOptionalFeature -Online -FeatureName $featureName -NoRestart  
-ErrorAction SilentlyContinue | Out-Null
```

```
        $featureAction = "Feature '$featureName' was enabled and has been disabled."
```

```
    } else {
```

```
        $featureAction = "Feature '$featureName' was already disabled."
```

```
    }
```

```
}
```

```
# -----
```

```
# 2) Set policy via registry
```

```
# -----
```

```
$regPath    = 'HKLM:\SOFTWARE\Policies\Microsoft\Internet Explorer>Main'
```

```
$valueName  = 'NotifyDisableIEOptions'
```

```
$desiredValue = 0 # 'Never' notify, IE11 disabled as standalone
```

```
if (-not (Test-Path $regPath)) {
```

```
    New-Item -Path $regPath -Force | Out-Null
```

```
}
```

```

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

# -----
# 3) Verification
# -----
$verifyReg = Get-ItemProperty -Path $regPath -Name $valueName -ErrorAction
SilentlyContinue
$actualValue = $verifyReg.$valueName

if ($actualValue -eq $desiredValue -and ($feature -eq $null -or $feature.State -eq 'Disabled')) {
    $complianceStatus = 'Compliant'
} else {
    $complianceStatus = 'Non-Compliant'
}

$result = [pscustomobject]@{
    ComputerName      = $env:COMPUTERNAME
    STIG_ID           = $stigId
    SettingName       = 'Disable Internet Explorer 11 as a standalone browser'
    RegistryPath      = 'HKLM\SOFTWARE\Policies\Microsoft\Internet Explorer>Main'
    ValueName          = $valueName
    RequiredValue     = $desiredValue
    ActualValue        = $actualValue
    IE_Feature_Action = $featureAction
    Registry_Action   = $regAction
    ComplianceStatus  = $complianceStatus
    Timestamp          = (Get-Date)
}

$result | Format-List *
# Optional CSV output for your report:

```

```
# $result | Export-Csv ".\WN11-CC-000391_$(($env:COMPUTERNAME)).csv"
-NoTypeInformation -Append
```

Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000391
- Status: **Passed**

Evidence:

The screenshot shows the Tenable Vulnerability Management interface. At the top, it displays the scan name "Win11DSBruceNov11". Below the title, there are tabs for "Vulns by Plugin", "Audits" (which is selected), "Vulns by Asset", and "History". A search bar contains the finding ID "WN11-CC-000391", and a results count of "1 Results" is shown. The main pane displays a single audit result: "0 Failed", "0 Warning", and "1 Passed". The passed result is for the family "Windows Compliance Checks" and has a status of "Passed". On the right side, there is a summary of vulnerabilities: 0 Critical, 0 High, 0 Medium, and 0 Low. Below this, the "Scan Details" section provides information about the scan: STATUS Completed, START TIME 11/11/2025 at 12:16 PM, TEMPLATE Advanced Network Scan, SCANNER LOCAL-SCAN-ENGINE-01, and TARGETS 10.1.0.118.

6. Conclusion

The finding **WN11-CC-000391** 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.