Evaluate-STIG User Manual and Configuration Guide

Version 1.2304.0
Last Updated: May 8, 2023

Table of Contents

Glossary and Abbreviations

Checklist (CKL)
Form used to check a system or program’s compliance with a STIG

Not_Applicable (Status)
Neutral status. A STIG check is not applicable to a given system based on STIG recommendations.

Not_Reviewed (Status)
Default status, all STIG checks begin with a *Not_Reviewed* status. The check requires manual intervention from an auditor.

NotAFinding (Status)
Positive status. A STIG check complies with applicable STIG recommendations.

Open (Status)
Negative status. A STIG check does not meet applicable STIG recommendations.

PowerShell
Microsoft command line interface capable of completely managing an operating system

Security Technical Implementation Guide (STIG)
A best practice methodology for securing and hardening IT systems

STIG Check
An individual vulnerability within a given STIG. There are many checks to a single STIG.

STIG Check Status
A categorization for the compliance of an individual STIG check. See *NotAFinding*, *Open*, *Not_Applicable*, and *Not_Reviewed*.

Evaluate-STIG Components

AnswerFiles

Optional XML files to inject common, standardized verbiage into the Comments field for specified Vuln IDs

CKLTemplates

Blank STIG checklist templates used by Evaluate-STIG to build final .ckl files

Modules

The Evaluate-STIG functions and check modules that automate the checklist process

Prerequisites

Scrips to assist in ensuring Windows prerequisites are met and to configure machine to trust the DoD issued code signing certificate.

xml

Contains Evaluate-STIG internal files for ensuring file integrity, STIG detection, XML schemas, and transforms.

Evaluate-STIG.ps1

Main Evaluate-STIG script. Directly executed on Windows and Linux (with PowerShell installed) systems and called by `Evaluate-STIG_Bash.sh` to conduct a scan.

Evaluate-STIG_Bash.sh

Script to allow scans of Linux assets that do not have PowerShell installed. Temporarily extracts PowerShell .tar.gz to the Evaluate-STIG directory, calls `Evaluate-STIG.ps1` to conduct the scan, then removes the PowerShell files from the Evaluate-STIG directory.

Evaluate-STIG.log

Log that asset scan is recorded to.
Located under *`%windir%\temp\Evaluate-STIG`* (Windows) or `/tmp/Evaluate-STIG` (Linux), the log is formatted to be viewed with `cmtrace.exe`. Using `cmtrace.exe`, the log can be viewed in real-time as the script executes.

Evaluate-STIG_Remote.log

Log that remote scan activity is recorded to.
Located under *`%temp%\Evaluate-STIG`* (Windows), the log is formatted to be viewed with `cmtrace.exe`. Using `cmtrace.exe`, the log can be viewed in real-time as the script executes.

Introduction

Evaluate-STIG is a suite of PowerShell scripts and modules intended to automate the creation of Security Technical Implementation Guide (STIG) checklists. It is only used for documenting STIG compliance and not for configuring to STIG requirements.

Evaluate-STIG can greatly reduce or eliminate the manual efforts typically required when documenting compliance into STIG Viewer compatible checklist files (CKL) while providing more complete, accurate, and consistent results. Evaluate-STIG will automatically detect which STIGs are required for the asset being scanned ensuring the applicable CKLs are created. Documentation that used to take hours or days can now be completed in minutes.

Finding Details for each STIG item will be populated with the system's actual configuration and the appropriate Status determined for the Check. To increase automation, user-defined Answer Files may be configured to insert standardized verbiage into the Comments section for STIG items that are policy based, mitigation, or justification documentation for known open items.

Each run of the Evaluate-STIG will automatically create a backup of existing checklists and preserve previous administrator Comments for *Open* and *Not_Reviewed* items. Unless defined in an Answer File, STIG checks that are not supported by Evaluate-STIG will remain as *Not_Reviewed* (regardless of the previous checklist's status) but the administrator comments will be retained. This is to ensure that non-reviewed items are validated each quarter and identify potential configuration drift.

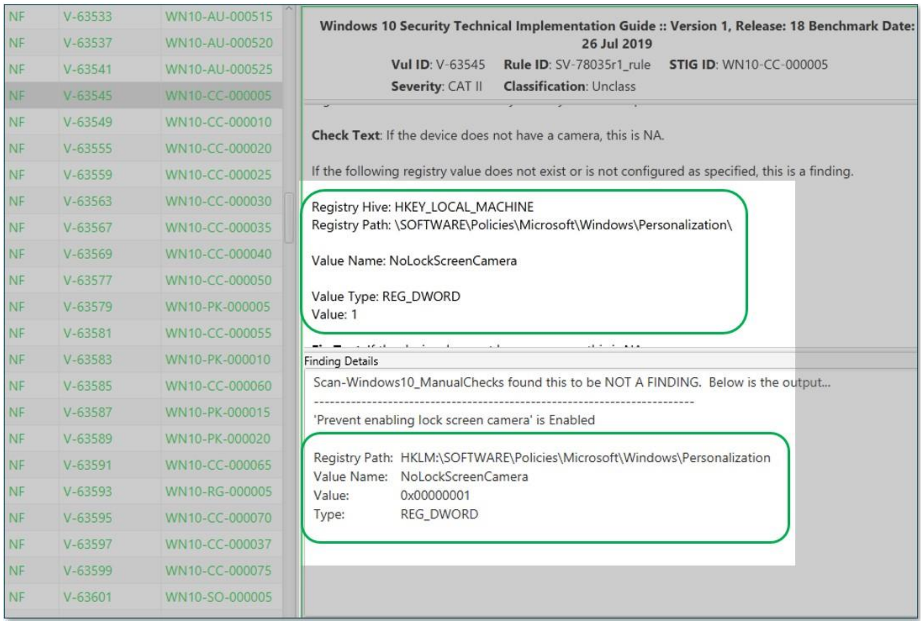
Evaluate-STIG can validate both computer and user settings. For user settings, the script identifies the most recent user profile that group policy was updated or the last user to log on and creates a temporary copy of the user's registry hive to `HKLM:\Evaluate-STIG_UserHive\<SID>`. The script searches this temporary copy to validate if settings are configured per STIG. The temporary copy will be removed at the end of processing.

Evaluate-STIG supports several deployment styles including Standalone, Remote, Scheduled Task, and via configuration management tools (e.g. Microsoft SCCM). Each of these deployments, as well as their operational procedures are discussed in Deployment Scenarios and Procedures. A detailed breakdown of Evaluate-STIG's command line options are provided in PowerShell Usage and Available Options.

In the simplest scenario, Evaluate-STIG is ran by acquiring the latest updates (discussed in Getting the Latest Updates), opening an elevated PowerShell prompt and running the Evaluate-STIG script with no arguments. This operation automatically determines applicable STIGs, creates CKL files, and outputs results to `C:\Users\Public\Documents\STIG-Compliance`.

Benefits to Business Processes

Evaluate-STIG is able to assist an auditor in STIG compliance and allows for a more complete STIG CKL over the traditional method of STIG compliance checking. Following the traditional process, the burden is upon the Analyst to determine which STIGs are applicable to a machine, to create blank CKL files, and to process each STIG individually. In contrast, Evaluate-STIG allows for the automation of determining STIG applicability and the creation of CKL files, as well as a more complete automated evaluation of STIG checks. Coupled with domain specific, custom AnswerFiles, this automation can be further improved and customized for individual domains. Furthermore, through the use of Evaluate-STIG, an analyst can easily verify each check's evaluated results due to the similarities between the Finding Details output and Check Text description; examples of which are shown in Figure 1 and Figure 2.



NF	V-63533	WN10-AU-000515
NF	V-63537	WN10-AU-000520
NF	V-63541	WN10-AU-000525
NF	V-63545	WN10-CC-000005
NF	V-63549	WN10-CC-000010
NF	V-63555	WN10-CC-000020
NF	V-63559	WN10-CC-000025
NF	V-63563	WN10-CC-000030
NF	V-63567	WN10-CC-000035
NF	V-63569	WN10-CC-000040
NF	V-63577	WN10-CC-000050
NF	V-63579	WN10-PK-000005
NF	V-63581	WN10-CC-000055
NF	V-63583	WN10-PK-000010
NF	V-63585	WN10-CC-000060
NF	V-63587	WN10-PK-000015
NF	V-63589	WN10-PK-000020
NF	V-63591	WN10-CC-000065
NF	V-63593	WN10-RG-000005
NF	V-63595	WN10-CC-000070
NF	V-63597	WN10-CC-000037
NF	V-63599	WN10-CC-000075
NF	V-63601	WN10-SO-000005

Windows 10 Security Technical Implementation Guide :: Version 1, Release: 18 Benchmark Date: 26 Jul 2019

Vul ID: V-63545 **Rule ID**: SV-78035r1_rule **STIG ID**: WN10-CC-000005
Severity: CAT II **Classification**: Unclass

Check Text: If the device does not have a camera, this is NA.

If the following registry value does not exist or is not configured as specified, this is a finding.

Registry Hive: HKEY_LOCAL_MACHINE
Registry Path: \SOFTWARE\Policies\Microsoft\Windows\Personalization\

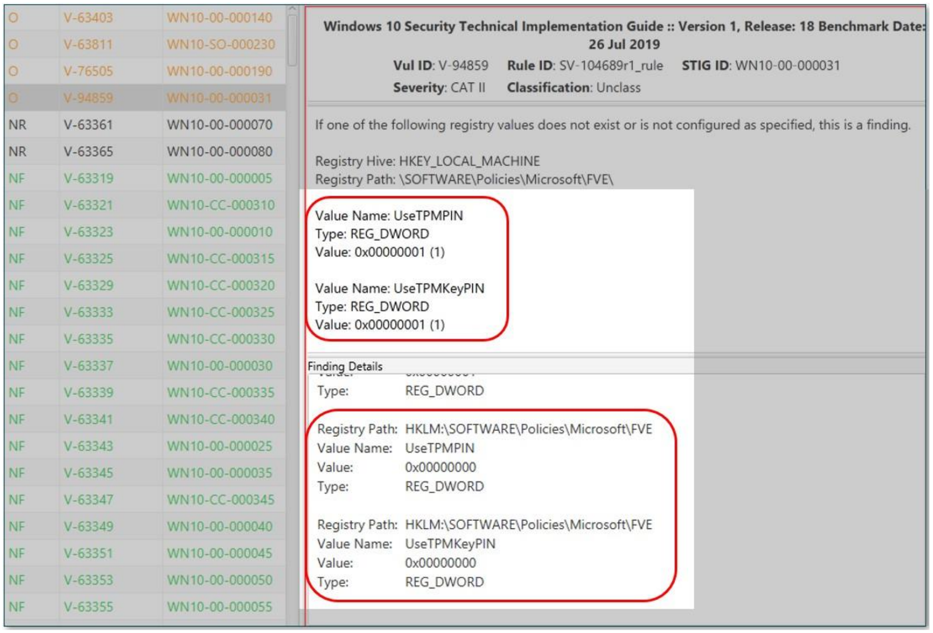
Value Name: NoLockScreenCamera

Value Type: REG_DWORD
Value: 1

Finding Details

```
Scan-Windows10_ManualChecks found this to be NOT A FINDING.  Below is the output...
--------------------------------------------------------------------------
'Prevent enabling lock screen camera' is Enabled

Registry Path:  HKLM:\SOFTWARE\Policies\Microsoft\Windows\Personalization
Value Name:     NoLockScreenCamera
Value:          0x00000001
Type:           REG_DWORD
```

Figure 1: Sample Finding Details for *NotAFinding* status



O	V-63403	WN10-00-000140
O	V-63811	WN10-SO-000230
O	V-76505	WN10-00-000190
O	V-94859	WN10-00-000031
NR	V-63361	WN10-00-000070
NR	V-63365	WN10-00-000080
NF	V-63319	WN10-00-000005
NF	V-63321	WN10-CC-000310
NF	V-63323	WN10-00-000010
NF	V-63325	WN10-CC-000315
NF	V-63329	WN10-CC-000320
NF	V-63333	WN10-CC-000325
NF	V-63335	WN10-CC-000330
NF	V-63337	WN10-00-000030
NF	V-63339	WN10-CC-000335
NF	V-63341	WN10-CC-000340
NF	V-63343	WN10-00-000025
NF	V-63345	WN10-00-000035
NF	V-63347	WN10-CC-000345
NF	V-63349	WN10-00-000040
NF	V-63351	WN10-00-000045
NF	V-63353	WN10-00-000050
NF	V-63355	WN10-00-000055

Windows 10 Security Technical Implementation Guide :: Version 1, Release: 18 Benchmark Date: 26 Jul 2019

Vul ID: V-94859 **Rule ID**: SV-104689r1_rule **STIG ID**: WN10-00-000031
Severity: CAT II **Classification**: Unclass

If one of the following registry values does not exist or is not configured as specified, this is a finding.

Registry Hive: HKEY_LOCAL_MACHINE
Registry Path: \SOFTWARE\Policies\Microsoft\FVE\

Value Name: UseTPMPIN
Type: REG_DWORD
Value: 0x00000001 (1)

Value Name: UseTPMKeyPIN
Type: REG_DWORD
Value: 0x00000001 (1)

Finding Details

```
Type:           REG_DWORD

Registry Path:  HKLM:\SOFTWARE\Policies\Microsoft\FVE
Value Name:     UseTPMPIN
Value:          0x00000000
Type:           REG_DWORD

Registry Path:  HKLM:\SOFTWARE\Policies\Microsoft\FVE
Value Name:     UseTPMKeyPIN
Value:          0x00000000
Type:           REG_DWORD
```

Figure 2: Sample Finding Details for *Open* status

Finally, as PowerShell has now been ported to the Linux environment, the Evaluate-STIG tool is capable of providing a single, common tool for STIG compliance checks in the future. This would simplify the compliance check procedure by reducing the number of tools used by auditors. Given Evaluate-STIG's modular nature, new evaluations can be added with ease, regardless of the operating system platform, and without sacrificing overall system performance.

System Requirements

Supported Operating Systems

Windows

- Windows 10
- Windows 11
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

Linux

Note: Requires **libicu** *and* **lshw** *be installed.*

- Oracle Linux 7
- Oracle Linux 8
- RHEL/CentOS 7
- RHEL 8
- Ubuntu 16.04 LTS
- Ubuntu 18.04 LTS
- Ubuntu 20.04 LTS

Powershell

- PowerShell 5.1 | PowerShell 7.x or greater (PowerShell 6 is not supported)
  Note: Using **Evaluate-STIG_Bash.sh**, *PowerShell is not required to be "installed" on Linux systems.*

STIG Viewer

- STIG Viewer 2.17 or greater (*https://cyber.mil/stigs/srg-stig-tools/*)
- For viewing completed checklists

Prerequisites

PowerShell Configuration (Windows Only)

By default, PowerShell restricts custom made scripts from running on a Windows system for security reasons. In order to provide the most secure and trusted operation possible, Evaluate-STIG has been signed using a DOD provided certificate. These processes are for if PowerShell's Execution Policy is configured as "*AllSigned*" or "*RemoteSigned*" (status can be checked via `Get-ExecutionPolicy`). If the Execution Policy is "*Restricted*", you must first change execution policy to a supported configuration (`Set-ExecutionPolicy`) or follow the steps for *No Touch* assets below. If the Execution Policy is "*Unrestricted*", this section is unnecessary. Please note, however, that an "*Unrestricted*" system allows any PowerShell script to be ran on the computer and should be considered a security concern.

For domain-joined systems, Group Policy should be configured to automatically install the certificate (`.\Evaluate-STIG\Prerequisites\Certificates\CS.NSWCCD.001.cer`) to the computer's `Local Machine\Trusted Publisher` store. For standalone or non-domain systems, the certificate will need to be installed manually. For *No Touch* assets (i.e. PIT), either the certificate needs to be installed or PowerShell's security policy needs to be temporarily bypassed so Evaluate-STIG can audit the system. Two batch files are included with Evaluate-STIG to assist with these prerequisites:

- **`Test-Prerequisites.bat`** – This script will verify the certificates are installed in the proper stores and that execution policy is at a supported configuration.
- **`Import-Certificates.bat`** – This script will import the certificates into the appropriate stores to ensure PowerShell trusts the code.

For No Touch assets or anyone not wanting to install the certificate, open PowerShell as an Administrator and use the following command to allow Evaluate-STIG to bypass a more restricted Execution Policy:

```
PowerShell.exe -ExecutionPolicy Bypass -File Evaluate-STIG.ps1
<Evaluate-STIG Options>
```

⇨ ***NOTE:*** *See Available Options for a description of the `<Evaluate-STIG Options>`.*

Getting the Latest Updates

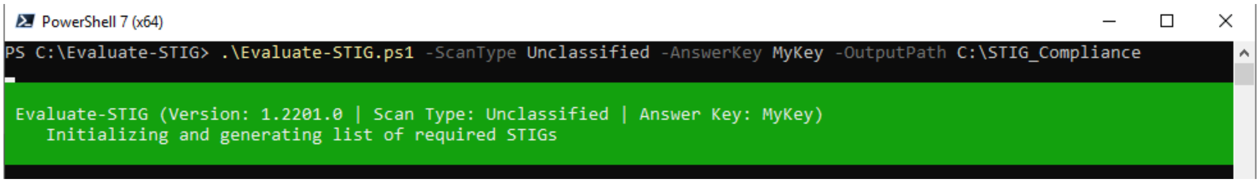
Accurate STIG compliance relies on using the most up to date evaluation tools, which means checking for Evaluate-STIG tool updates at least every quarter when DISA releases new STIGs. While Evaluate-STIG provides automation for supported STIGs (supported STIG list is shipped with the Evaluate-STIG application), the traditional process for STIG compliance checks will need to be performed for any non-supported STIGs. Evaluate-STIG updates can be acquired two different ways:

- Navigating to *https://spork.navsea.navy.mil/nswc-crane-division/evaluate-stig/-/releases* and downloading the .zip archive.
- Running Evaluate-STIG with the `-Update` option.

PowerShell Usage

From an elevated PowerShell prompt:

```
.\Evaluate-STIG.ps1 [-ScanType "Unclassified"/"Classified"]
[-Marking <String>] [-AnswerKey <String>] [-AFPath <String>]
[-OutputPath <String>] [-SelectSTIG <String>] [-ExcludeSTIG <String>]
[-ComputerName <String>] [-CiscoConfig <String>] [-ThrottleLimit <int>]
[-AltCredential] [-GenerateOQE] [-NoPrevious] [-ApplyTattoo]
[-ListSupportedProducts] [-ListApplicableProducts] [-Version] [-Update]
[-Proxy <String>]
```



```
PS C:\Evaluate-STIG> .\Evaluate-STIG.ps1 -ScanType Unclassified -AnswerKey MyKey -OutputPath C:\STIG_Compliance

Evaluate-STIG (Version: 1.2201.0 | Scan Type: Unclassified | Answer Key: MyKey)
   Initializing and generating list of required STIGs
```

Figure 3: Example of basic Evaluate-STIG tool usage with options

The help page can be obtained with: `Get-Help .\Evaluate-STIG.ps1 -Full`

Available Options

`-ScanType <"Unclassified"/"Classified">`

Example: `-ScanType Classified`

Unclassified (default): If -ScanType is not specified, this is used by default. Drives checks that are based on asset classification

Classified: Drives checks that are based on asset classification

`-Marking <String>`

Example: `-Marking "CUI"`

Use to optionally set Marking in CKL and on header/footer of files generated by Evaluate-STIG. Example use is "CUI"/"Confidential"/"Secret"/"Top Secret" but accepts any marking string.

`-VulnTimeout <int>`

Example: `-VulnTimeout 30`

Set the maximum time in minutes allowed for a singular Vuln ID check to run. Default is 15 minutes.

`-AnswerKey <KeyName>`

Example: `-AnswerKey MyKey`

Use to instruct Evaluate-STIG which Answer Key to use for determining if a comment from an answer file should be applied. Answer Keys are per Vuln ID and user-defined within the answer file. If not specified, the default Answer Key will be "DEFAULT". Answer Keys/Files are discussed in Answer Files.

`-AFPath <Path>`

Example: `-AFPath \\FileShare01\AnswerFiles`

Specifies an alternative repository for Answer Files. Can be either a local or UNC path. The default location is `.\Evaluate-STIG\AnswerFiles`.

`-OutputPath <Path>`

Example: `-OutputPath C:\STIG_Complicance`

Sets the path for saving checklist files and artifacts. May be a local path or UNC path (e.g. \\myfileshare.com\share). If using in conjunction with `-ComputerName`, ensure the host computer's account has write access to the path specified. Also, ensure the destination folder is created as Evaluate-STIG will not create it. If `-OutputPath` is not specified, defaults to

Windows: `C:\Users\Public\Documents\`
Linux: `/opt/`

`-SelectSTIG <String>`

Example: `-SelectSTIG Chrome,IE11,MSEdge,Win10`

Specify which STIG(s) to scan. Use [Tab] or [CTRL + Space] to properly select STIG(s) by their short names. For multiple STIGs, separate with a comma. This option cannot be used with `-ExcludeSTIG`. STIG short names are identified in the `ListSupportedProducts` option.

`-SelectVuln <String>`

Example: `-SelectSTIG Firefox -SelectVuln V-251553,V-251557`

Specify which vulnerability IDs to include in scan. For multiple vuln IDs, separate with commas. Requires `-SelectSTIG` parameter. Results will be saved to a "_Partial" folder under <OutputPath>.

`-ExcludeVuln <String>`

Example: `-SelectSTIG Firefox -ExcludeVuln V-251553,V-251557`

Specify which vulnerability IDs to exclude from scan. For multiple vuln IDs, separate with commas. Requires `-SelectSTIG` parameter.

Note: If a vuln ID is both selected (`-SelectVuln`*) and excluded (*`-ExcludeVuln`*), exclusion will take precedent for that vuln ID. Example :* `Evaluate-STIG.ps1 -SelectSTIG Firefox -SelectVuln V-251553,V-251557 -ExcludeVuln V-251557` *would result in a partial CKL with only V-251553 completed because V-251557 is ultimately excluded per* `-ExcludeVuln`*.*

`-ExcludeSTIG <String>`

Example: `-ExcludeSTIG Chrome,IE11,MSEdge,Win10`

Specify which STIG(s) to exclude from scanning. Use [Tab] or [CTRL + Space] to properly select STIG(s) by their short names. For multiple STIGs, separate with a comma. This option cannot be used with `-SelectSTIG`. STIG short names are identified in the `ListSupportedProducts` option.

`-ComputerName <ComputerName|List>`

Example: `-ComputerName Workstation01, "C:\ComputerList.txt"`

Windows Only. Supports multiple computers through a comma separated list. Input can be a computer name, IP address, a text file of computer names (one name per line), a PowerShell array object, or a combination. Copies the Evaluate-STIG content to the remote computer's `%windir%\Temp` and executes the scan. Host will wait for completion of scan on remote computer. Administrator rights on remote computer and properly configured Windows Remote Management (WinRM) is required for this option.

`-CiscoConfig <Path to File/Folder>`

Example: `-CiscoConfig C:\ShowTech.txt, "C:\ShowTechFolder\"`

Scan Cisco `show tech-support` output file(s). Supports multiple computers through a comma separation. Input can be path to a `show tech-support` file or a folder of `show tech-support` files.

`-ThrottleLimit <int>`

Example: `-ThrottleLimit 7`

The number of concurrent scans to run when using `-ComputerName` or `-CiscoConfig`. The default is 10.

`-AltCredential`

Windows Only. Prompts for an alternate credential to use for remote scans. If the connection to the remote machine fails with the alternate credential, Evaluate-STIG will fallback to the launching user and attempt the connection. Requires `-ComputereName` input.

`-GenerateOQE`

Creates Objective Quality Evidence (OQE) files in output path.

`-NoPrevious`

Disable moving current CKLs to Previous folder.

`-ApplyTattoo`

Applies Evaluate-STIG tattooing on the system. Mainly used for providing a detection method to configuration management tools.

Windows: Creates "Version" and "LastRun" values under `HKLM:\SOFTWARE\Evaluate-STIG`
Linux: Creates "Version" and "LastRun" values in `/etc/Evaluate-STIG`

`-ListSupportedProducts`

List all products (STIGS) supported by Evaluate-STIG, then exits. All other switches are ignored. Does not show which STIGs are applicable to a system.

`-ListApplicableProducts`

List all Evaluate-STIG supported STIGs that are applicable to the asset. All other switches are ignored.

`-Version`

Display the version of Evaluate-STIG and the running path.

`-Update`

Downloads updated Evaluate-STIG code from SPORK. Answer Files are ignored to prevent overwriting of user customization.

`-Proxy <String>`

Example: `-Proxy 10.0.0.2:8080`
Example: `-Proxy http://proxy.mil:1234`

Sets the proxy to use for `-Update`.

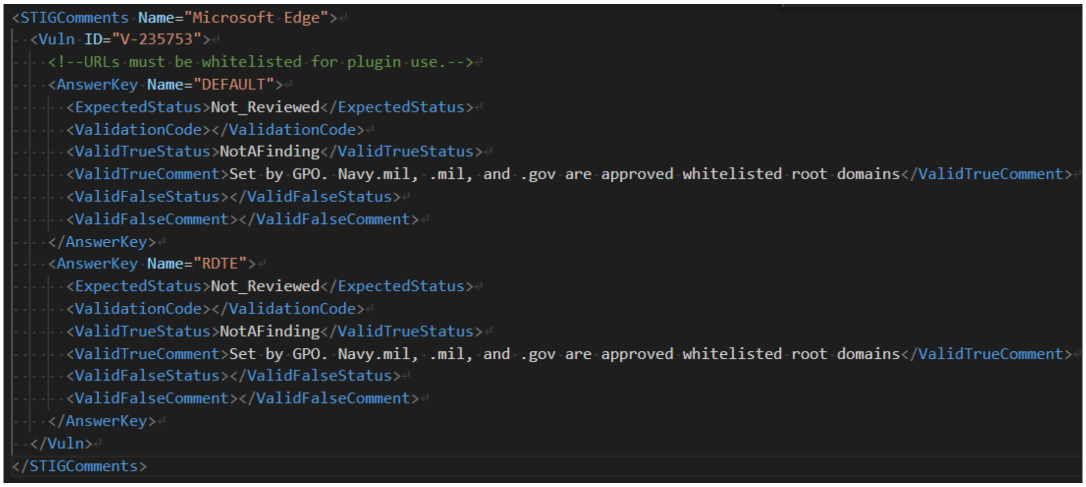
⇨ ***NOTE:*** *Refer to* https://docs.microsoft.com/en-us/windows/win32/winrm *for more information on WinRM and how to enable.*

Configuration and Customization

While Evaluate-STIG can be run out-of-the-box with no additional configuration, there are options to customize it for a given environment to increase its automation capability or to select specific STIGs with which to scan a machine.

Answer Files

Answer Files can be created to further automate *Not_Reviewed* findings which cannot be evaluated technically or may require further validation than the typical true or false, yes or no STIG check. Answer Files can also be used to provide a justification or mitigation to known *Open* findings. Answer Files will place an approved comment in the Comments field of the checklist finding, can change the resultant STIG Check Status of the finding, and may include code to validate that the criteria are met for the answer data to be applied. Answer Files must be properly formatted and may be stored in the `.\`*`Evaluate-STIG\AnswerFiles`* folder or an alternate location when using the `-AFPath` option.



```
<STIGComments Name="Microsoft Edge">
  <Vuln ID="V-235753">
    <!--URLs must be whitelisted for plugin use.-->
    <AnswerKey Name="DEFAULT">
      <ExpectedStatus>Not_Reviewed</ExpectedStatus>
      <ValidationCode></ValidationCode>
      <ValidTrueStatus>NotAFinding</ValidTrueStatus>
      <ValidTrueComment>Set by GPO. Navy.mil, .mil, and .gov are approved whitelisted root domains</ValidTrueComment>
      <ValidFalseStatus></ValidFalseStatus>
      <ValidFalseComment></ValidFalseComment>
    </AnswerKey>
    <AnswerKey Name="RDTE">
      <ExpectedStatus>Not_Reviewed</ExpectedStatus>
      <ValidationCode></ValidationCode>
      <ValidTrueStatus>NotAFinding</ValidTrueStatus>
      <ValidTrueComment>Set by GPO. Navy.mil, .mil, and .gov are approved whitelisted root domains</ValidTrueComment>
      <ValidFalseStatus></ValidFalseStatus>
      <ValidFalseComment></ValidFalseComment>
    </AnswerKey>
  </Vuln>
</STIGComments>
```

Figure 4: Example of an Answer File entry for the .NET Framework 4 STIG Vulnerability V-225236

STIGComments Name: Name or ShortName for the STIG as it appears in the `ListSupportedProducts` output. Evaluate-STIG will automatically use this Answer File for the STIG provided the data here equals the **Name** or **ShortName** value.

Vuln ID: The Vuln ID from the STIG. Multiple VULN ID Sections may be specified in a single Answer File but only one for each Vuln ID is allowed (e.g. V-12345 and V-67890 may both be configured in a single Answer File, but V-12345 configured twice in the same Answer File is not allowed).

AnswerKey Name: User defined answer key name. Typically, an answer key will be created for each of your environments so you can call the appropriate key for the system(s) being scanned. A value of “DEFAULT” can be used to apply a standardized comment to all systems and all environments. If a DEFAULT key and a named key exist (as shown in Figure 4), the named key will take precedence if specified with the `-AnswerKey` option. Multiple AnswerKey sections may be specified within a single Vuln ID section.

ExpectedStatus: The initial status from the checklist after the scans are complete. *Not_Reviewed, Open, NotAFinding,* or *Not_Applicable*.

ValidationCode: PowerShell code that must return a true/false value. If result returns "true", answer data will be applied as specified with **ValidTrueStatus** and **ValidTrueComment**. If anything other than "true", **ValidFalseStatus** and **ValidFalseComment** are used. Leaving **ValidationCode** blank is considered "true".

ValidTrueStatus: The status set if **ValidationCode** returns "true". Either *Not_Reviewed, Open, NotAFinding,* or *Not_Applicable*. If this and **ValidationFalseStatus** are blank, **ExpectedStatus** is used.

ValidTrueComment: The verbiage to be input into the Comments field for the STIG item if **ValidationCode** returns "true".

ValidFalseStatus: The status set if **ValidationCode** returns anything other than "true". Either *Not_Reviewed, Open, NotAFinding,* or *Not_Applicable*. If this and **ValidationTrueStatus** are blank, **ExpectedStatus** is used.

ValidFalseComment: The verbiage to be input into the Comments field for the STIG item if **ValidationCode** returns anything other than "true".

⇨ ***NOTE:*** *Comments from an AnswerFile will always override administrator comments from a previous checklist.*

Answer File Processing Flow

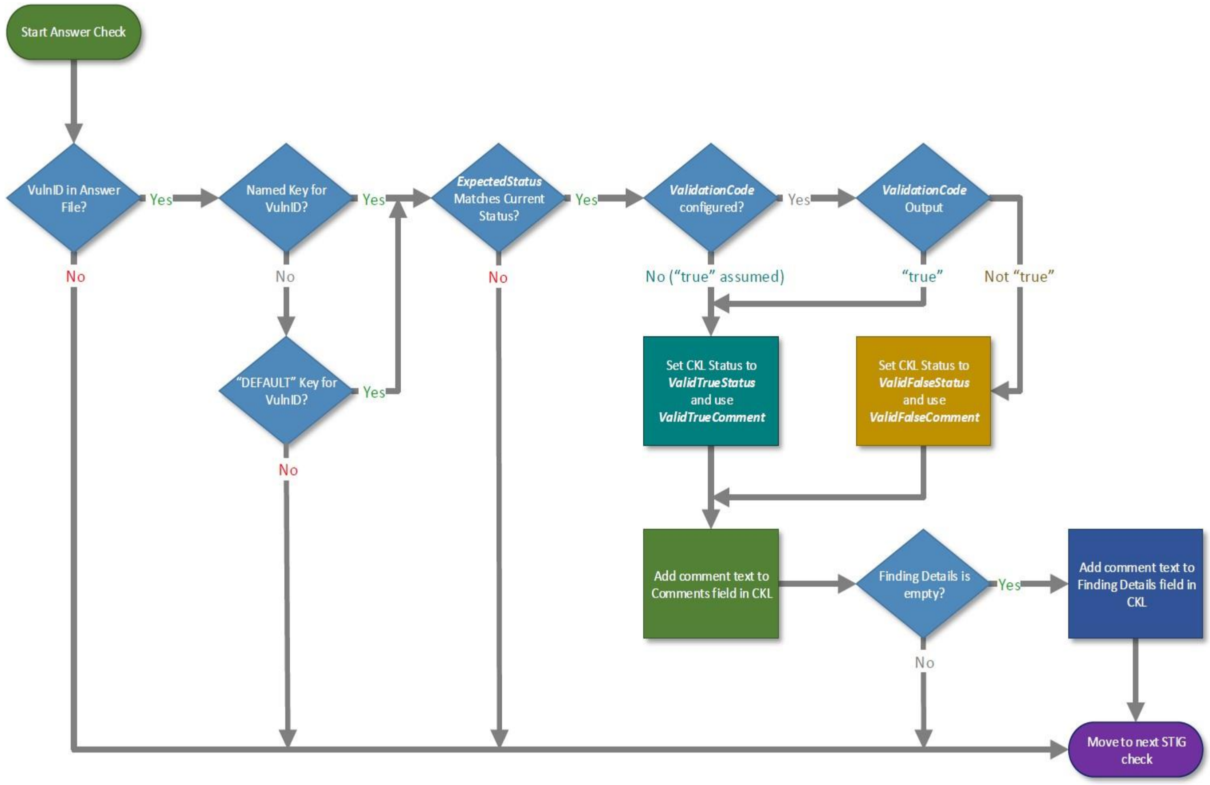

Start Answer Check
VulnID in Answer File?
Yes
No
Named Key for VulnID?
Yes
No
"DEFAULT" Key for VulnID?
Yes
No
ExpectedStatus Matches Current Status?
Yes
No
ValidationCode configured?
Yes
No ("true" assumed)
ValidationCode Output
"true"
Not "true"
Set CKL Status to *ValidTrueStatus* and use *ValidTrueComment*
Set CKL Status to *ValidFalseStatus* and use *ValidFalseComment*
Add comment text to Comments field in CKL
Finding Details is empty?
Yes
No
Add comment text to Finding Details field in CKL
Move to next STIG check

Figure 5: Flowchart for processing Answer Files

Creating Organization-Specific AnswerFiles

Depending on the STIG under evaluation, there could be several organization-specific checks requiring institutional knowledge in order to process. The following procedures guide the creation of an AnswerFile but note that each AnswerFile/AnswerKey is unique to an organization. Another item to remember is, because of an AnswerFile's uniqueness, **each warfare center will need to manage their AnswerFile upkeep and add them to a repository or the proper Evaluate-STIG folder (*`.\Evaluate-STIG\AnswerFiles\`*) for Evaluate-STIG to utilize them.** The easiest way to get a proper template is to copy an existing AnswerFile.

1. Within *`.\Evaluate-STIG\AnswerFiles\`*, name the XML AnswerFile after the STIG to which it belongs with "_AnswerFile.xml" appended
   a. Example: `WindowsServer2016_AnswerFile.xml`
2. Copy the contents of another AnswerFile, or use Figure 4 as a reference
   a. **STIGComments Name** needs to reflect the **Name** or **ShortName** from `ListSupportedProducts`.
3. Under the **STIGComments Name** tag, add all applicable **Vuln ID** tags
4. For each **Vuln ID** tag, provide a unique **AnswerKey Name** for your organization
   a. The AnswerKey string is what is used as an argument when running Evaluate-STIG
5. In the **ApprovedComment** tag, add any comment you would like record when this AnswerKey entry is used
6. Configure the **ExpectedStatus** to match the results of the generated Evaluate-STIG CKL
7. Develop the **ValidationCode** tag with PowerShell code if there are additional conditions that should be met before applying the final status
   a. This code should only return a "true" or "false" response
   b. If **ValidationCode** is left blank, the **ValidTrueStatus** and **ValidTrueComment** will be applied if **ExpectedStatus** is matched
8. Repeat Steps 4 – 8 as for as many Vuln IDs are applicable

Deployment Scenarios and Procedures

This section will cover the procedures and configurations necessary to deploy Evaluate-STIG in several useful scenarios including Standalone, Remote Computer, Scheduled Task, and via SCCM. For Linux deployments, note that only the standalone deployment and SCCM deployment may be applicable.

In terms of Evaluate-STIG's processing performance, while system memory, or RAM, may be a factor in the Evaluate-STIG tool's performance, the program runs in a single thread and therefore no additional performance is gained from machines with a large core count.

Formatting Key

Anything to be typed into a computer will be in a `monospace font`

Folder, File, and Program names will be in *`italic monospace font`*

Menu navigation will be denoted with a greater than (>) symbol: File > New

Key presses are given in square brackets and simultaneous key presses are denoted with a plus symbol: [Ctrl + Alt + Del]

Standalone

The Evaluate-STIG tool can be used as a standalone tool for assessing STIG compliance on a single machine. While other methods of deployment exist for mass compliance auditing (discussed in System Center Configuration Manager), the standalone method is simple and allows for the checking of machines not connected to a network. The following depicts several examples – along with procedures for running the program – one may find useful when deploying the Evaluate-STIG tool in such a manner.

As mentioned before, the Evaluate-STIG program is a PowerShell script. Due to this there are several ways the script may be accessed including having the Evaluate-STIG tool directory listed in the Windows' environment Path variable or navigating the tool's folder hierarchy manually. These examples will assume a starting position at the root Evaluate-STIG folder. The results of the scan, if not specified, are stored in *C:\Users\Public\Documents\STIG-Compliance*.

⇨ ***NOTE:*** *The Evaluate-STIG tool must be run with an account having administrator permissions over the machine.*

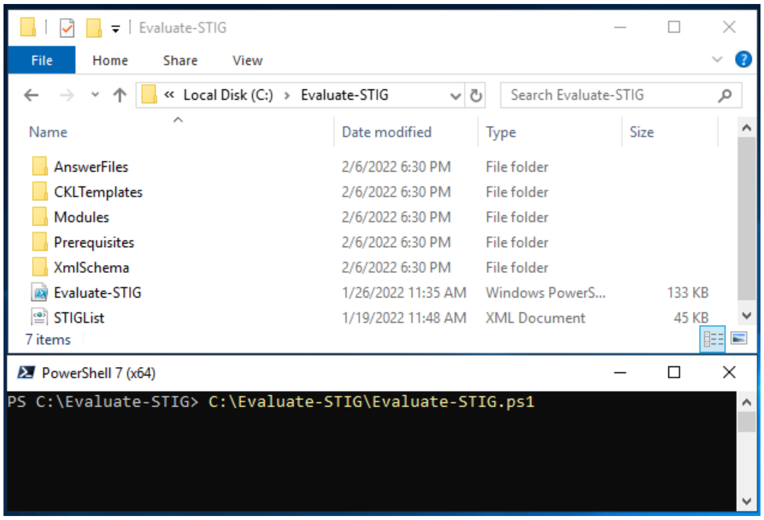
Scenario 1: The most basic use case

By running the Evaluate-STIG tool without any modifications to other files or arguments entered into the tool, Evaluate-STIG will run an unclassified scan and output any generated files to *C:\Users\Public\Documents\STIG_Complicance*. The Evaluate-STIG tool will run all applicable PowerShell checks.

1. Navigate to within the Evaluate-STIG folder in Windows Explorer
2. Open PowerShell (PS)
   a. PS v.5: Start > Windows PowerShell > *Windows PowerShell*
   b. PS v.7: Start > PowerShell > *PowerShell 7 (x64)*
   c. Linux: Type `pwsh` into a terminal, or pipe the Evaluate-STIG.ps1 script into `pwsh`

⇨ ***NOTE:*** *The Evaluate-STIG tool is written for PS v.5 or v.7, however due to current bugs within PS v.7 some checks are only possible in v.5. These are be denoted in Evaluate-STIG's Finding Details output as such.*

3. Enter the full file path to the Evaluate-STIG tool into the PS prompt. As a shortcut, you can click-and-drag the *Evalute-STIG.ps1* file onto the PS prompt, shown in Figure 6.



Name	Date modified	Type	Size
AnswerFiles	2/6/2022 6:30 PM	File folder	
CKLTemplates	2/6/2022 6:30 PM	File folder	
Modules	2/6/2022 6:30 PM	File folder	
Prerequisites	2/6/2022 6:30 PM	File folder	
XmlSchema	2/6/2022 6:30 PM	File folder	
Evaluate-STIG	1/26/2022 11:35 AM	Windows PowerS...	133 KB
STIGList	1/19/2022 11:48 AM	XML Document	45 KB

```
PS C:\Evaluate-STIG> C:\Evaluate-STIG\Evaluate-STIG.ps1
```

Figure 6: Entering Evaluate-STIG file location into PowerShell

4. Press [Enter] to run the Evaluate-STIG tool

Scenario 2: Run a custom scan with redirected output

The Evaluate-STIG tool is designed to be as automatic as possible. Because of this, using command line arguments is simple, yet allows the end-user to dictate various parameters of scans, as well as to direct the scan's output to an appropriate datastore. Remember, proper access must be given on the remote datastore for the user account running the command, typically an administrator-level account.

1. Determine the Answer File that should be used in the scan
   a. In Windows Explorer, navigate to *`.\Evaluate-STIG\AnswerFiles`*
   b. If the desired Answer File is not specified in this folder, create one based on the example shown in Answer Files
2. If not already, open a PS prompt
   a. PS v.5: Start > Windows PowerShell > *`Windows PowerShell`*
   b. PS v.7: Start > PowerShell > *`PowerShell 7 (x64)`*
3. Enter the path for the Evaluate-STIG tool or drag the file from Windows Explorer into the PS prompt space as shown in Figure 6
4. Add the additional desired arguments to the command before pressing [Enter]
   a. Run a Classified scan:
      `-ScanType Classified`
   b. Use a custom Answer File:
      `-AnswerKey MyKey`
   c. Output to a remote datastore:
      `-OutputPath \\remote-server\datastore\STIG_Compliance`

Remember an *AnswerKey* is defined under a given VulnID in the appropriate Answer File per STIG. The final PS command should appear similar to below, provided the Evaluate-STIG folder is stored on *`C:\`* and the remote datastore is on *remote-server*. Note that the example below is typed as a single line and [Enter] is only pressed once at the end of the line.

```
PS C:\> C:\Evaluate-STIG\Evaluate-STIG.ps1 -ScanType Classified -
AnswerKey MyKey -OutputPath \\remote-
server\datastore\STIG_Compliance
```

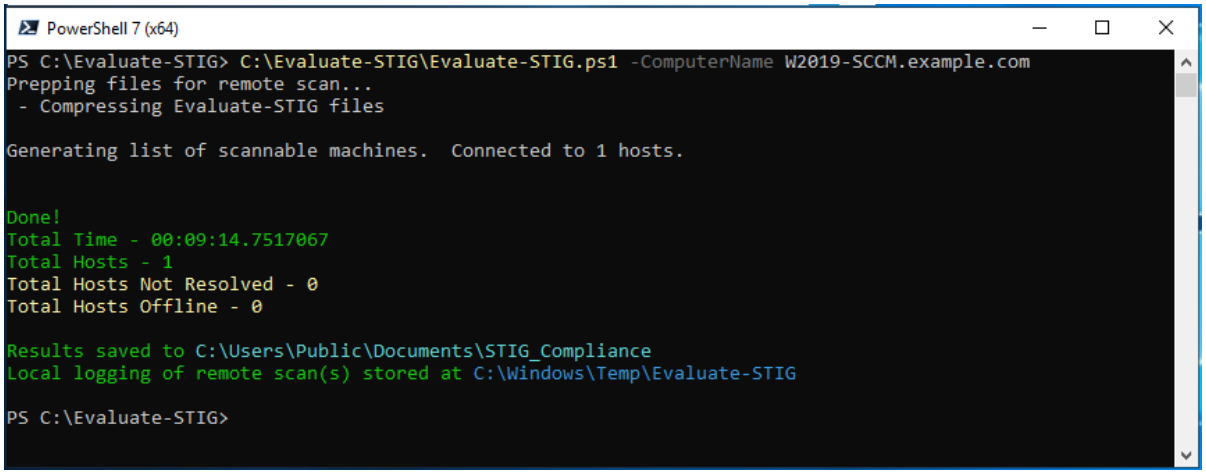
Scanning a Remote Computer

Building from the procedures in Standalone, using Evaluate-STIG on a remote computer is an easy procedure, provided the computer is joined to the same domain as the initiating computer. As of this writing, using the `-ComputerName` flag for non-domain joined systems is unsupported due to the system configuration necessary for WinRM to accept connections properly. That said, provided the initiating and remote system are joined to the same domain, the process involves a single step of difference from the Standalone procedures above.

⇨ ***NOTE:*** Following the 20H2 update for Windows, Evaluate-STIG will fail to scan a Linux remote computer via SSH if the SSH Logon banner is enabled. See Evaluate-STIG fails to remote scan Linux machine using SSH for more details.

The -`ComputerName` flag copies the contents of `.\Evaluate-STIG` to the remote computer, and executes the Evaluate-STIG script as the initiating user. `–AltCredential` can be used to specify an alternative user instead. Evaluate-STIG will fall back to the initiating credential if the alternative credential provided fails. This allows for a single command make use of two credentials, which may be helpful for scanning workstations and servers together. An example of the command and output are given in Figure 7.

1. Within an elevated PS prompt, execute Evaluate-STIG with the `-ComputerName` option and supply the Fully Qualified Domain Name (FQDN) for the remote computer: `.\Evaluate-STIG.ps1 -ComputerName Win2019DC`



```
PS C:\Evaluate-STIG> C:\Evaluate-STIG\Evaluate-STIG.ps1 -ComputerName W2019-SCCM.example.com
Prepping files for remote scan...
 - Compressing Evaluate-STIG files

Generating list of scannable machines.  Connected to 1 hosts.


Done!
Total Time - 00:09:14.7517067
Total Hosts - 1
Total Hosts Not Resolved - 0
Total Hosts Offline - 0

Results saved to C:\Users\Public\Documents\STIG_Compliance
Local logging of remote scan(s) stored at C:\Windows\Temp\Evaluate-STIG

PS C:\Evaluate-STIG>
```

Figure 7: Executing Evaluate-STIG on the remote computer *domainClient-2016-1.crane.lcl*

2. In order to retrieve the results, open Windows Explorer
3. In the address bar, navigate to `\\`*`[FQN pf RemoteComputer]\c$\Users\Public\Documents\STIG_Compliance`*

The task needs to complete for files will appear. Log files are located at `C:\Windows\Temp\Evaluate-STIG` on the remote computer. Procedures for viewing the task status and log from the initiator computer are below. To view the files created remotely, follow the procedures below.

1. On the initiator computer, open Windows Explorer
2. In the address bar, navigate to `\\`*`[FQDN of RemoteComputer]\c$\Windows\Temp\Evaluate-STIG`*
3. Open the `Evaluate-STIG.log` file in a text editor or *`cmtrace.exe`*

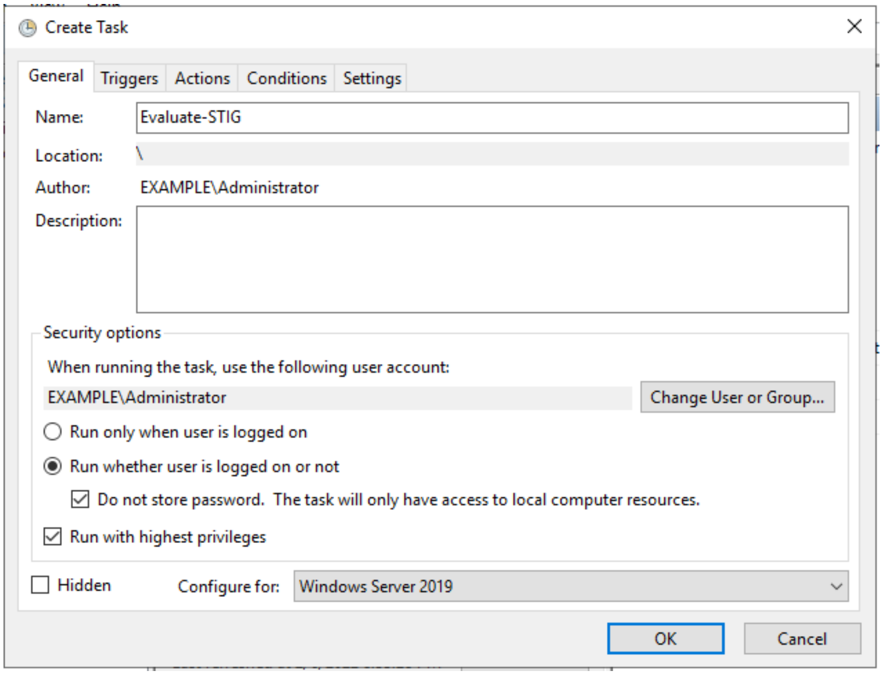
⇨ ***NOTE:*** `cmtrace.exe` *can be obtained from the System Center 2012 R2 Configuration Manager Toolkit (https://www.microsoft.com/en-us/download/details.aspx?id=50012). Only the* `cmtrace.exe` *file is needed from the toolkit.*

Scheduled Task

A scheduled task allows Evaluate-STIG to execute at a set time in the future, such as after business hours. To have the script execute at a later time or date, use the following procedures.

1. On the computer to be scanned, select Start > Windows Administrative Tools > *`Task Scheduler`*

2. Right-Click *Task Scheduler (Local)* in the left-most pane and select “Create Task”
3. Within the pop-up window, under the *General* tab, give the task a name such as `Evaluate-STIG`
4. Select which user account to use by clicking the *Change User or Group…* button
5. Choose the radio button “Run whether the user is logged on or not”
6. Check the checkbox marked “Run with highest privileges”
7. Under the “Configure for:” combobox, select the most appropriate operating system (see Figure 8)



Name: Evaluate-STIG

Location: \

Author: EXAMPLE\Administrator

Description:

Security options

When running the task, use the following user account:

EXAMPLE\Administrator

○ Run only when user is logged on

● Run whether user is logged on or not

☑ Do not store password. The task will only have access to local computer resources.

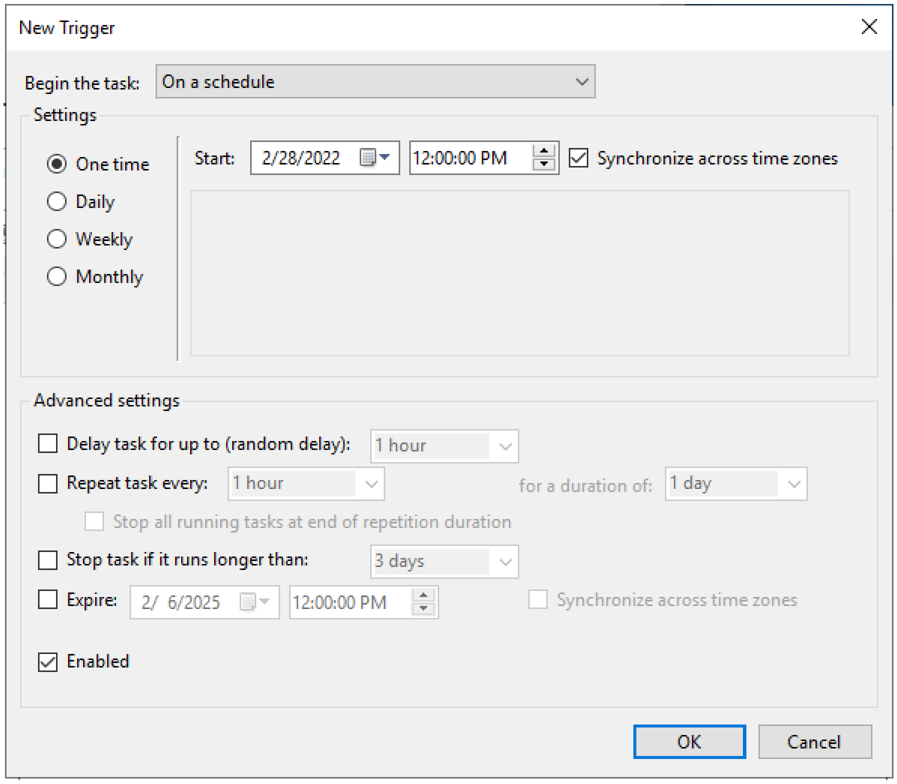
☑ Run with highest privileges

□ Hidden Configure for: Windows Server 2019

Figure 8: Create Task dialog with options configured to run as Administrator for Windows Server 2016

8. Under the *Triggers* tab, click the *New…* button

9. Within the *New Trigger* dialog, configure the specific time the task should run and ensure the "Enabled" checkbox is selected (see Figure 9)



Begin the task: On a schedule

● One time
○ Daily
○ Weekly
○ Monthly

Start: 2/28/2022 12:00:00 PM ☑ Synchronize across time zones

☐ Delay task for up to (random delay): 1 hour

☐ Repeat task every: 1 hour for a duration of: 1 day

☐ Stop all running tasks at end of repetition duration

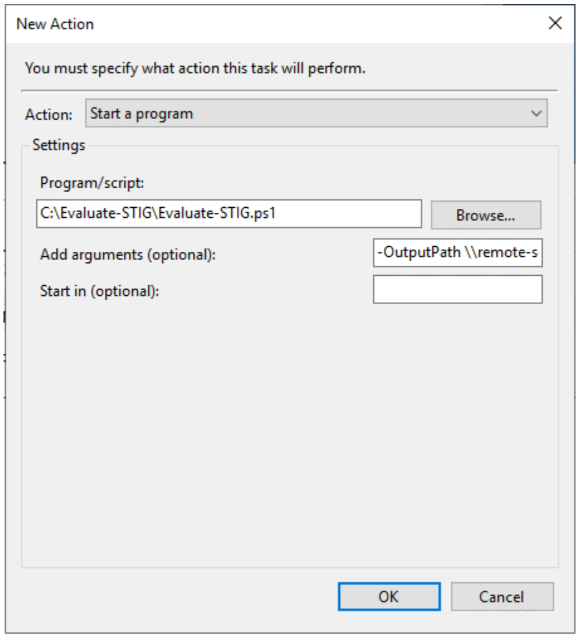
☐ Stop task if it runs longer than: 3 days

☐ Expire: 2/ 6/2025 12:00:00 PM ☐ Synchronize across time zones

☑ Enabled

Figure 9: Edit Trigger dialog with options to run the task once at midnight

10. Under the *Actions* tab, select the *New…* button
11. In the *New Action* dialog, ensure the "Action" is set to "Start a program"
12. Browse for the Evaluate-STIG script using the *Browse…* button under "Program/script"
13. Specify any arguments in the "Add arguments (optional)" textbox such as `-OutputPath \\Remote-Share\STIG-Compliance`
14. Leave the "Start in (optional)" textbox empty (see Figure 10)



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

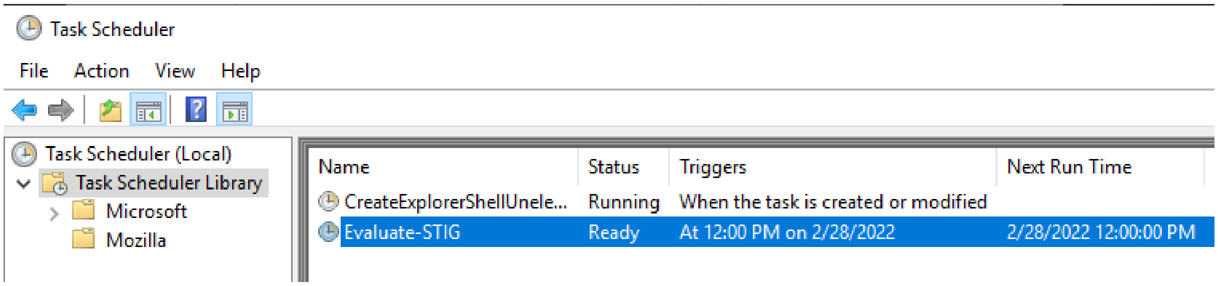
Program/script: C:\Evaluate-STIG\Evaluate-STIG.ps1

Add arguments (optional): -OutputPath \\remote-s

Start in (optional):

Figure 10: Edit Action dialog specifying what program to run and with which arguments

15. The *Conditions* and *Settings* tabs can be configured if desired, but are not necessary
16. After selecting the *OK* button on all open dialog boxes to complete the configuration, the configured task can be viewed by navigating to Task Scheduler (Local) > Task Scheduler Library in the left-most pane of Task Scheduler, shown in Figure 11



Name	Status	Triggers	Next Run Time
CreateExplorerShellUnele...	Running	When the task is created or modified	
Evaluate-STIG	Ready	At 12:00 PM on 2/28/2022	2/28/2022 12:00:00 PM

Figure 11: User-defined Scheduled Task

System Center Configuration Manager

Evaluate-STIG was designed to be deployed by system management tools like Microsoft's System Center Configuration Manager (SCCM) to maximize automation. For anyone unaware, SCCM, now packaged within Microsoft Endpoint Configuration Manager, is a software package designed to manage software installation and configuration for all computer systems within an organization regardless of operating system or form factor. SCCM provides remote control, patch management, software distribution, operating system deployment, network access protection, and hardware and software inventory.

This section will provide step-by-step guidance to creating a SCCM Application for deploying Evaluate-STIG. It is assumed there is a fully functioning SCCM Current Branch deployment in your environment.

Before deploying Evaluate-STIG with SCCM, we need to first prepare the environment and Evaluate- STIG's configuration. For this example, we'll reference Figure 12.

- **fileserver1.contoso.com**: Server to host the file share for clients to save results.
- **sccm1.contoso.com**: SCCM server that also has a file share called **PackageSource** where existing SCCM Application source content is stored.

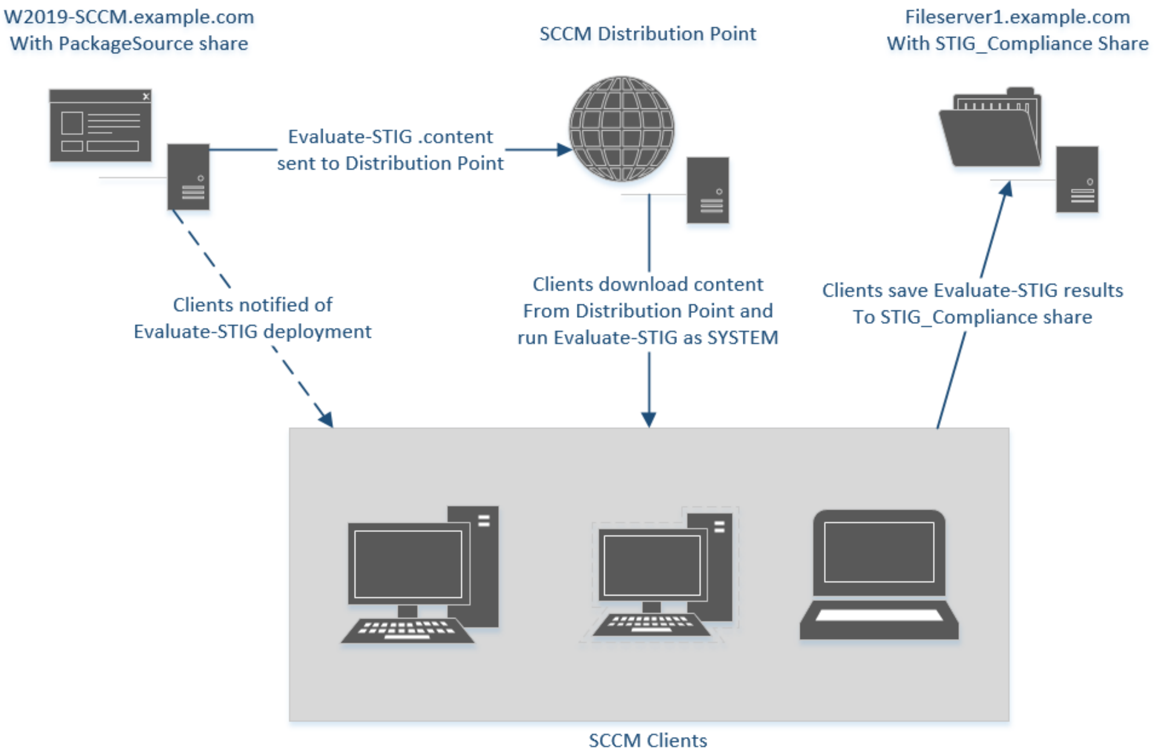
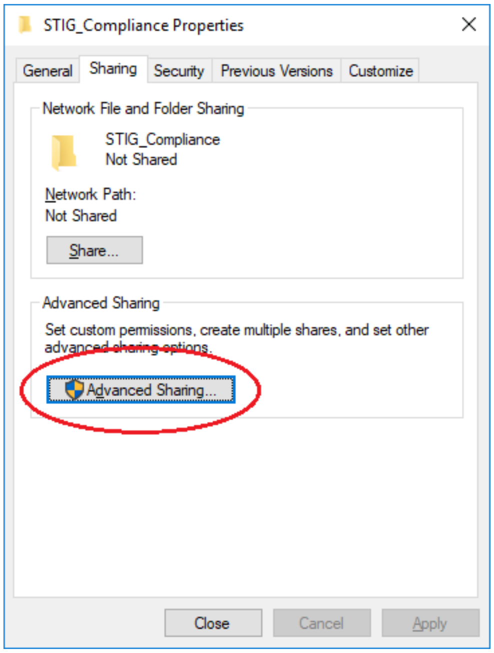


Figure 12: Example network for deployment of Evaluate-STIG via SCCM

Create the File Share for Clients to Save Results

⇨ ***NOTE:*** *Since SCCM executes its client processing under the SYSTEM account, we need to grant computer accounts write access on both share and NTFS permissions for the file share.*

1. On **fileserver1.contoso.com,** create a folder such as `C:\STIG_Compliance`
2. Right-click on the new `STIG_Compliance` folder and select *Properties*
3. In the *Properties* dialog, switch to the *Sharing* tab and click the *Advanced Sharing* button (see Figure 13).



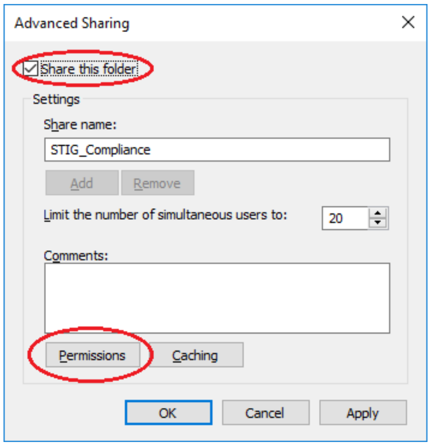
STIG_Compliance
Not Shared

Network Path:
Not Shared

Set custom permissions, create multiple shares, and set other advanced sharing options.

Figure 13: How to access Advanced Sharing in folder properties

4. Select the checkbox for “Share this folder”
5. Click to the *Permissions* button, to open the permissions dialog, shown in Figure 14



☑ Share this folder

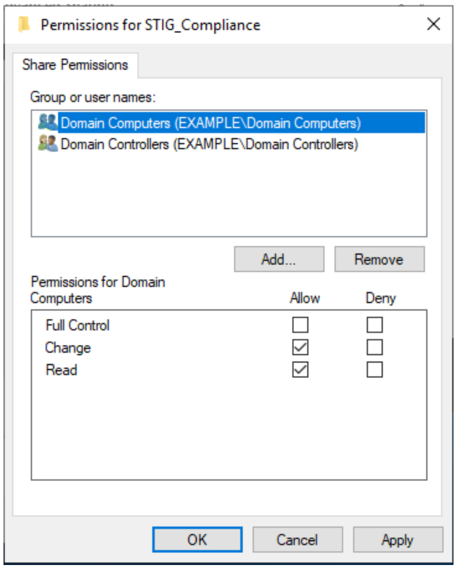
Share name: STIG_Compliance

Limit the number of simultaneous users to: 20

Comments:

Figure 14: Advanced Sharing dialog

6. Modify the *Share Permissions* to include the `Domain Computers` group with Change and Read permissions. If domain controllers are to be scanned, add the `Domain Controllers` group with these same permissions
    a. Click the *Add* button and use the *Add Users and Groups* dialog to add the appropriate groups

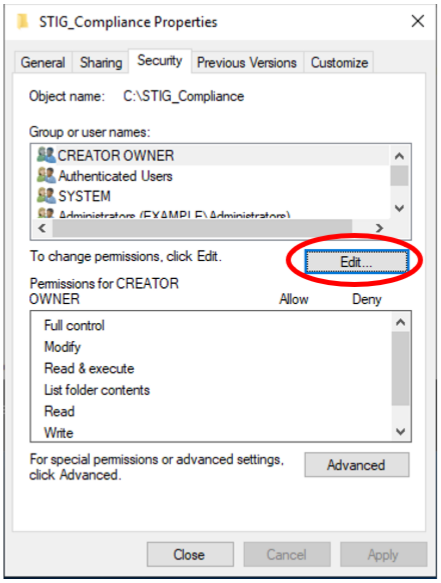


Group or user names:

- Domain Computers (EXAMPLE\Domain Computers)
- Domain Controllers (EXAMPLE\Domain Controllers)

Permissions for Domain Computers	Allow	Deny
Full Control	□	□
Change	☑	□
Read	☑	□

7. Close the *Permissions* and *Advanced Sharing* dialogs screens by clicking *OK*
8. Switch to the *Security* tab and click the *Edit* button (see Figure 15)



Object name: C:\STIG_Compliance

Group or user names:

- CREATOR OWNER
- Authenticated Users
- SYSTEM
- Administrators (EXAMPLE\Administrators)

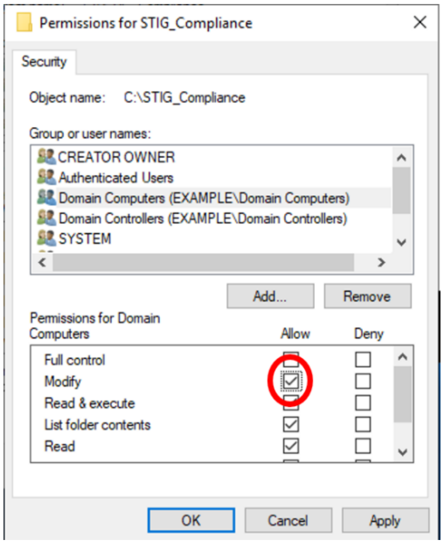
To change permissions, click Edit.

Permissions for CREATOR OWNER	Allow	Deny
Full control		
Modify		
Read & execute		
List folder contents		
Read		
Write		

For special permissions or advanced settings, click Advanced.

Figure 15: Security tab in folder properties

9. Click the *Add…* button to add the `Domain Computers` group and grant the Modify permissions as shown in Figure 16
   a. If domain controllers need to be scanned, add the `Domain Controllers` group with these same permissions



Permissions for STIG_Compliance

Security

Object name: C:\STIG_Compliance

Group or user names:

- CREATOR OWNER
- Authenticated Users
- Domain Computers (EXAMPLE\Domain Computers)
- Domain Controllers (EXAMPLE\Domain Controllers)
- SYSTEM

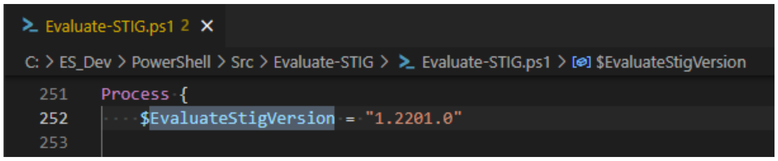
Permissions for Domain Computers	Allow	Deny
Full control	□	□
Modify	☑	□
Read & execute	□	□
List folder contents	☑	□
Read	☑	□

Figure 16: Security dialog for folder sharing permissions

10. Click *OK* to close the *Security Permissions* dialog, then click *OK* to close the *STIG_Compliance Properties* dialog

Prepare Evaluate-STIG for Deployment

1. After obtaining the latest version of the Evaluate-STIG tool (see Getting the Latest Updates), extract the Evaluate-STIG folder to the SCCM environment's content source repository (`\\W2019-SCCM1.example.com\PackageSource`)
2. Open `.\Evaluate-STIG\Evaluate-STIG.ps1` using a text editor
    a. If file extensions are not visible, within `Windows Explorer` switch to the *View* tab and select the checkbox for "File name extensions" in the "Show/hide" section
3. Locate the `$EvaluateStigVersion` variable and make a note of the version number found there, shown in Figure 17
4. Close the text editor application



Evaluate-STIG.ps1 2 ✕
C: > ES_Dev > PowerShell > Src > Evaluate-STIG > Evaluate-STIG.ps1 > $EvaluateStigVersion

```
Process {
    $EvaluateStigVersion = "1.2201.0"

```

Figure 17: Location of Evaluate-STIG version number

Create the Evaluate-STIG Application in SCCM

1. On `W2019-SCCM.example.com`, open the SCCM console
    a. Select Start > Microsoft System Center > `Configuration Manager Console`
2. Within the SCCM Console, in the left pane, select *Software Library* and navigate to Overview > Application Management
3. Right-Click "Applications" and select "Create Application" as shown in Figure 18

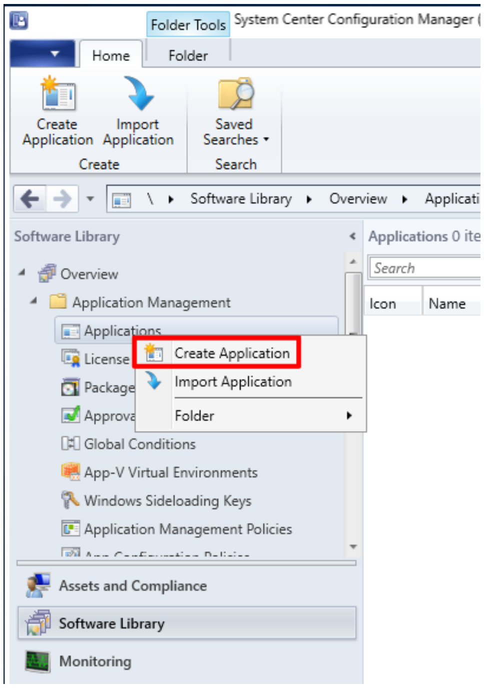
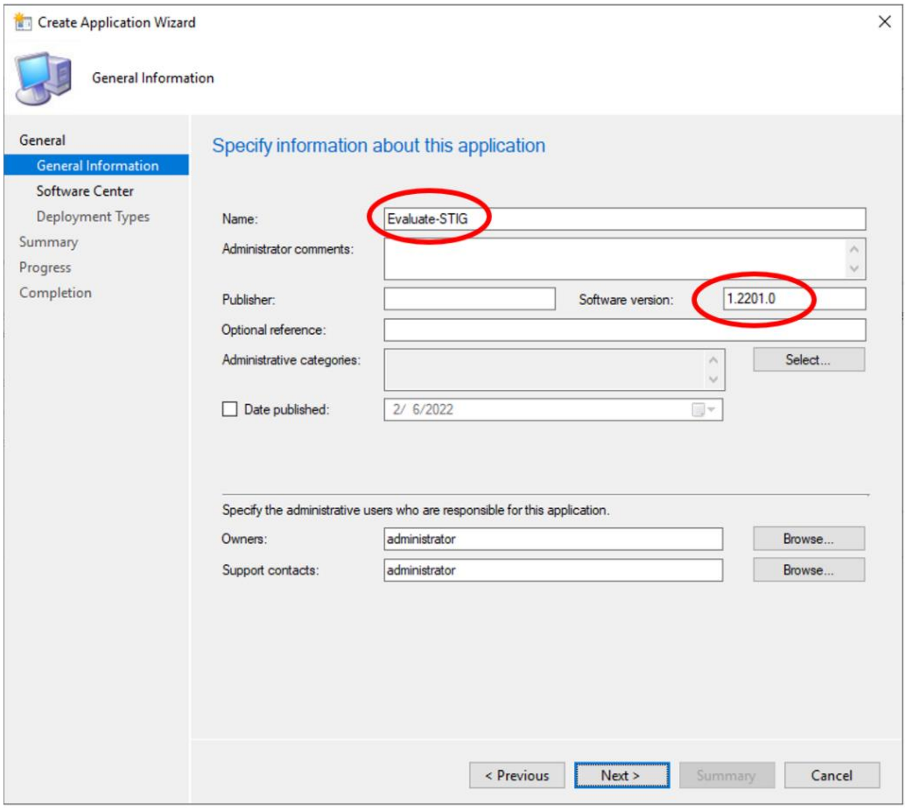


Figure 18: Selecting to create an application in SCCM

4. In the *Create Application Wizard*, under *Specify settings for this application*, select the "Manually specify the application information" radio button and click *Next*
5. On the *Specify Information about this application* page, enter `Evaluate-STIG` for the application name and the version number noted in Step 3 of Prepare Evaluate-STIG for Deployment under "Software version" (see Figure 19)
6. On the following page, *Specify the Configuration Manager Application Catalog entry*, you may optionally add an icon for the application by selecting the *Browse…* button near the "Icon" section, otherwise keep everything on this page as default
7. On the next page, *Configure deployment types and the priority in which they will be applied for this application*, click the *Add…* button to open the corresponding wizard
8. Within the *Create Deployment Type Wizard*, change the deployment type to "Script Installer" as shown in Figure 20, click *Next*
9. Enter `Evaluate-STIG` into the name field on the *Specify general information for this deployment type*, click next
10. On *Specify information about the content to be delivered to target devices*, enter in the following information into the respective fields as shown in Figure 21
    a. "Content Location": `\\W2019-SCCM.example.com\PackageSource\Evaluate- STIG`
    b. "Installation Program": `Powershell.exe -ExecutionPolicy Bypass -File Evalaute-STIG.ps1 -OutputPath Fileserver1.example.com\STIG_Compliance -ApplyTattoo`
    c. Deselect the "Run installation and uninstall program as 32-bit process on 64-bit clients" checkbox



Specify information about this application

Name: Evaluate-STIG

Administrator comments:

Publisher:

Software version: 1.2201.0

Optional reference:

Administrative categories:

□ Date published: 2/ 6/2022

Specify the administrative users who are responsible for this application.

Owners: administrator

Support contacts: administrator

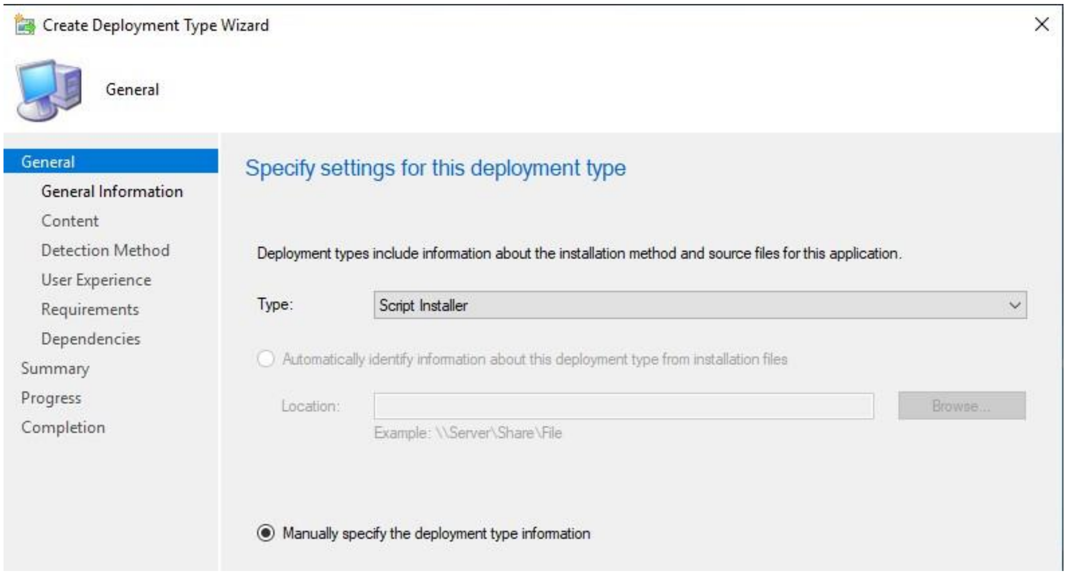
Figure 19: Providing an application name and software version within the SCCM Create Application Wizard

11. On *Specify how this deployment type is detected*, select the "Configure rules to detect the presence of this deployment type" radio button and click *Add Clause…*

Configure the settings within the *Detection Rule* dialog exactly as shown in Table 1 and Figure 22.

Table 1: Detection Rule dialog settings

Setting Type:	Registry
Key:	SOFTWARE\Evaluate-STIG
Hive:	HKEY_LOCAL_MACHINE
Value:	Version
Data Type:	Version
This registry setting must satisfy the following rule to indicate the presence of this application:	<Select>
Operator:	Equals
Value:	1.2201.0



Create Deployment Type Wizard

General

Specify settings for this deployment type

Deployment types include information about the installation method and source files for this application.

Type: Script Installer

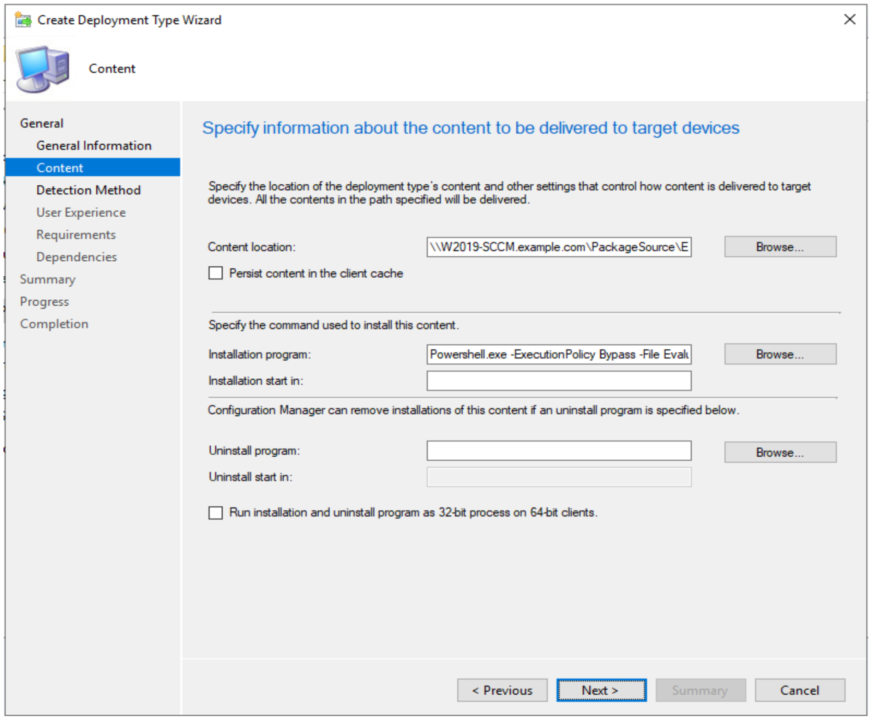
○ Automatically identify information about this deployment type from installation files

Location:

Example: \\Server\Share\File

● Manually specify the deployment type information

Figure 20: Configuring the deployment type to be "Script Installer"



Create Deployment Type Wizard

Content

Specify information about the content to be delivered to target devices

Specify the location of the deployment type's content and other settings that control how content is delivered to target devices. All the contents in the path specified will be delivered.

Content location: \\W2019-SCCM.example.com\PackageSource\E

□ Persist content in the client cache

Specify the command used to install this content.

Installation program: Powershell.exe -ExecutionPolicy Bypass -File Evalu

Installation start in:

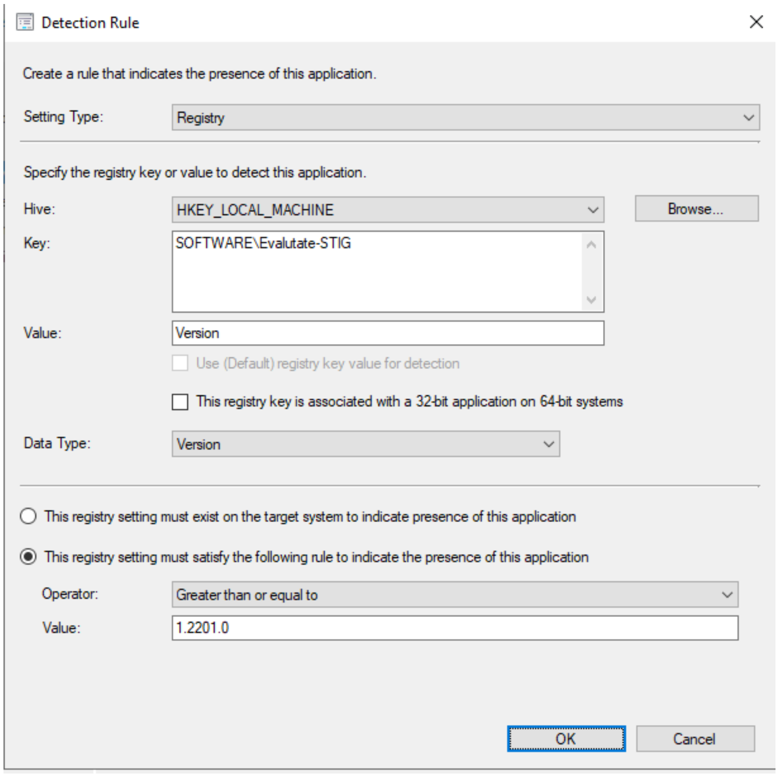
Configuration Manager can remove installations of this content if an uninstall program is specified below.

Uninstall program:

Uninstall start in:

□ Run installation and uninstall program as 32-bit process on 64-bit clients.

Figure 21: Specifying delivery information in the Create Deployment Type Wizard



Detection Rule

Create a rule that indicates the presence of this application.

Setting Type: Registry

Specify the registry key or value to detect this application.

Hive: HKEY_LOCAL_MACHINE

Key: SOFTWARE\Evalutate-STIG

Value: Version

□ Use (Default) registry key value for detection

□ This registry key is associated with a 32-bit application on 64-bit systems

Data Type: Version

○ This registry setting must exist on the target system to indicate presence of this application

● This registry setting must satisfy the following rule to indicate the presence of this application

Operator: Greater than or equal to

Value: 1.2201.0

Figure 22: Settings for the Detection Rule dialog

12. After configuring the *Detection Rule* as specified, click *Next*
13. Configure *Specify user experience settings for the application* as shown in Table 2 and Figure 23
    a. If the application times out due to clients having a large file system, increasing the "Maximum allowed run time" is recommended

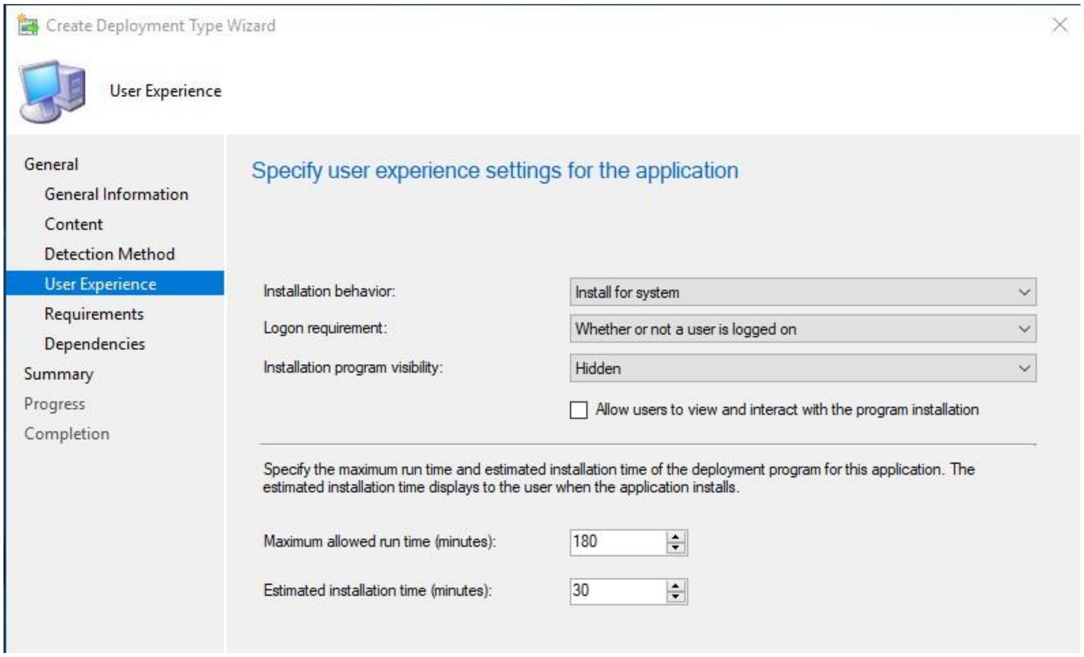
Table 2: User experience settings

Installation behavior:	Install for system
Logon requirement:	Whether or not a user is logged on
Installation program visibility:	Hidden
Allow users to view and interact with the program installation:	<Deselect>
Maximum allowed run time (minutes):	180
Estimation installation time (minutes):	30

14. On the *Specify installation requirements for this deployment type*, click *Add…*
15. Configure the *Create Requirement* dialog as described in Table 3.

Table 3: Create Requirement dialog settings

Category:	Device
Condition:	Operating System
Rule type:	Value
Operator:	One of
Operating Systems to Select:	Windows 10, Windows Server2012R2, 2016, 2019



Specify user experience settings for the application

Installation behavior: Install for system

Logon requirement: Whether or not a user is logged on

Installation program visibility: Hidden

☐ Allow users to view and interact with the program installation

Specify the maximum run time and estimated installation time of the deployment program for this application. The estimated installation time displays to the user when the application installs.

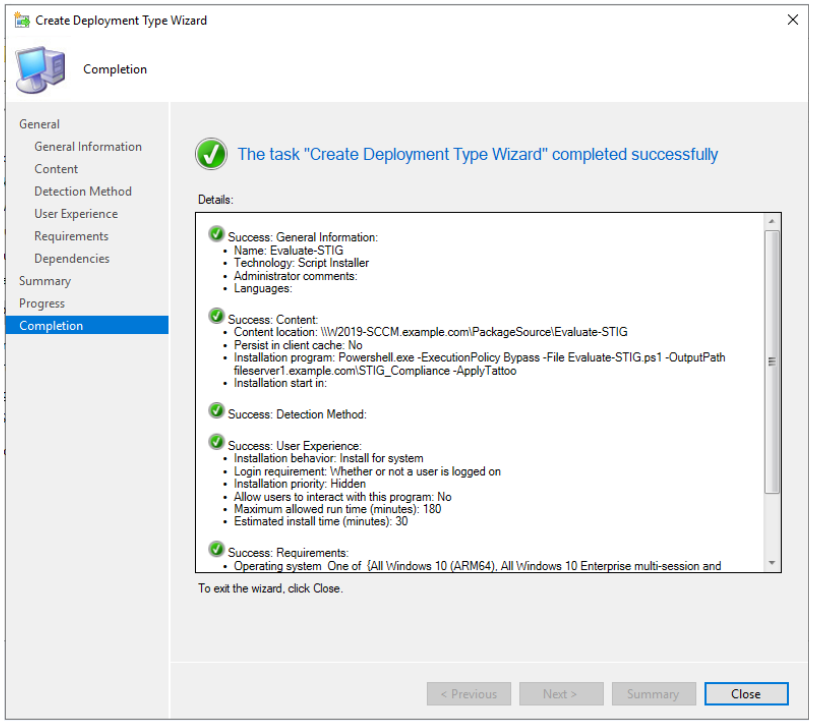
Maximum allowed run time (minutes): 180

Estimated installation time (minutes): 30

Figure 23: User experience settings example

16. Skip the *Specify software dependencies for this deployment type* as there are none, click *Next*
17. On the *Summary* page, confirm the configured settings are as expected, then click *Next*

Provided everything was configured properly, there should be a completion screen with every task completed successfully as in Figure 24.



The task "Create Deployment Type Wizard" completed successfully

Details:

Success: General Information:
- Name: Evaluate-STIG
- Technology: Script Installer
- Administrator comments:
- Languages:

Success: Content:
- Content location: \\W2019-SCCM.example.com\PackageSource\Evaluate-STIG
- Persist in client cache: No
- Installation program: Powershell.exe -ExecutionPolicy Bypass -File Evaluate-STIG.ps1 -OutputPath fileserver1.example.com\STIG_Compliance -ApplyTattoo
- Installation start in:

Success: Detection Method:

Success: User Experience:
- Installation behavior: Install for system
- Login requirement: Whether or not a user is logged on
- Installation priority: Hidden
- Allow users to interact with this program: No
- Maximum allowed run time (minutes): 180
- Estimated install time (minutes): 30

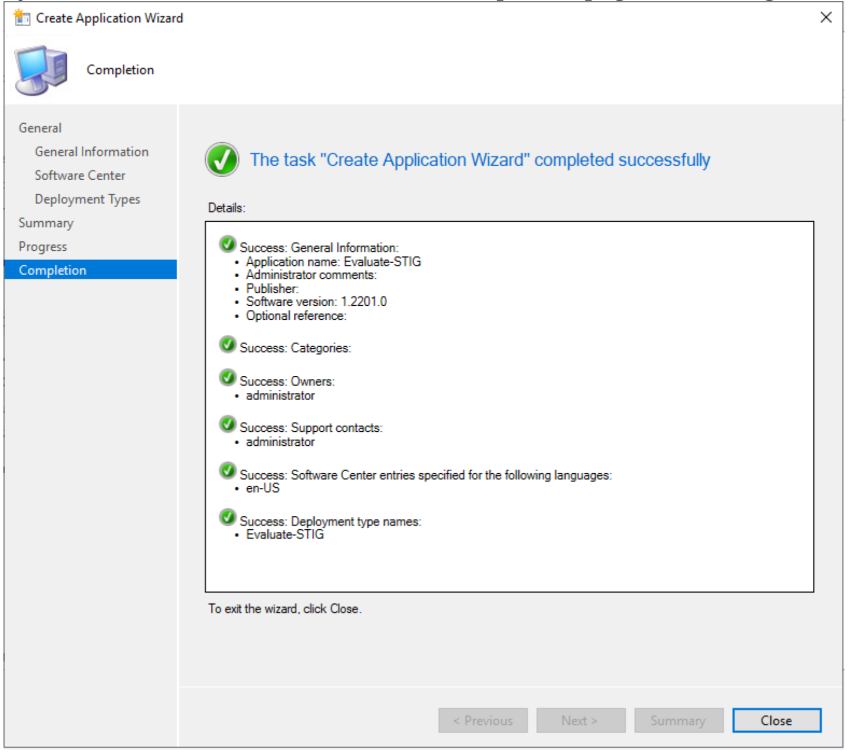
Success: Requirements:
- Operating system One of {All Windows 10 (ARM64), All Windows 10 Enterprise multi-session and

To exit the wizard, click Close.

Figure 24: Successful set up of the Deployment Type Wizard

18. Click *Close* on the *Create Deployment Type Wizard* and click *Next* on the *Create Application Wizard*
19. Confirm the settings of the *Create Application Wizard* and if correct, click *Next*

If no errors occur, you should be met with a successful completion page such as Figure 25.



The task "Create Application Wizard" completed successfully

Details:

Success: General Information:
- Application name: Evaluate-STIG
- Administrator comments:
- Publisher:
- Software version: 1.2201.0
- Optional reference:

Success: Categories:

Success: Owners:
- administrator

Success: Support contacts:
- administrator

Success: Software Center entries specified for the following languages:
- en-US

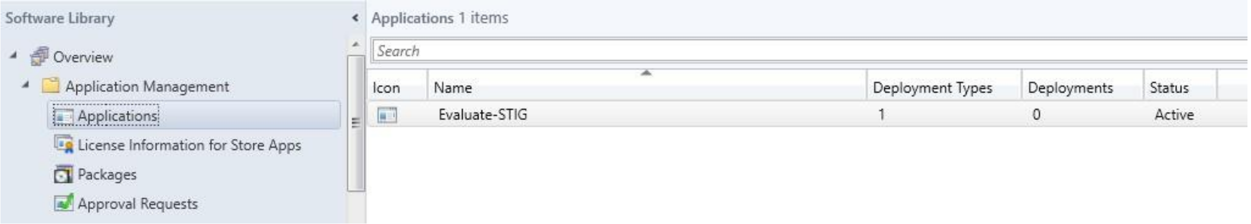
Success: Deployment type names:
- Evaluate-STIG

To exit the wizard, click Close.

Figure 25: Successful configuration of Create Application Wizard

20. Click *Close* to complete the *Create Application Wizard*

Back within SCCM, there should now be an entry in *Applications* for *Evaluate-STIG*, such as in Figure 26.



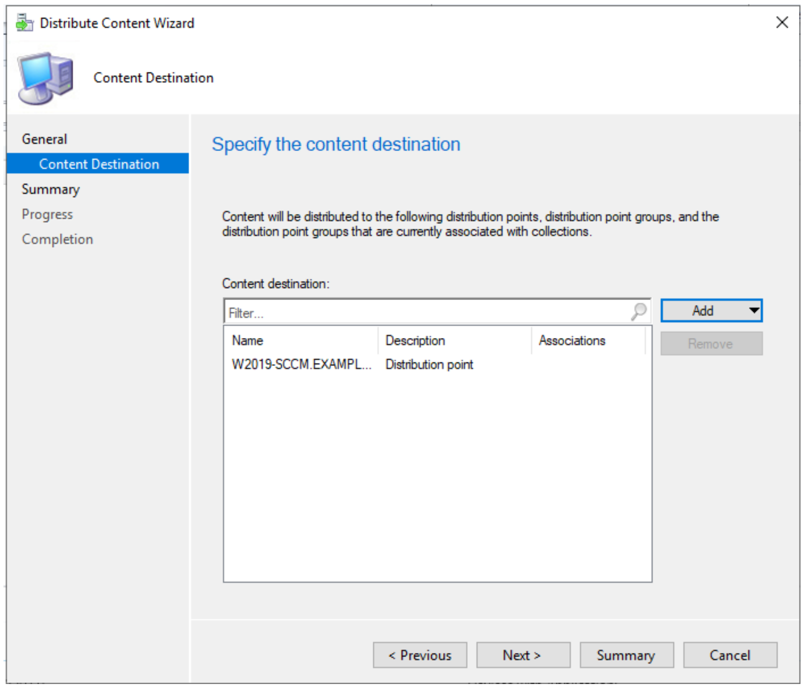
Icon	Name	Deployment Types	Deployments	Status
	Evaluate-STIG	1	0	Active

Figure 26: Confirmation within SCCM that the application was created

Deploying the SCCM Application

After creation the application needs to be distributed to select distribution points for client download and the application needs to be deployed to clients. In the past, the terms "distribution" and "deploying" had a similar meaning in SCCM, however, starting around version 1803, Microsoft has now distinguished the words' meaning to help with clarity. The act of "distributing content" means to download the application content to distribution points, those being other SCCM servers, in order to provide a better client experience and alleviate network traffic. "Deploying content" on the other hand means to alert clients that a new application exists and that the selected clients should follow the *Deployment Type* defined for the application.

1. Right-click on the *Evaluate-STIG* application that was just created and select "Distribute Content" from the context menu
2. In the *Distribute Content Wizard*, deselect the "Detect associated content dependencies and add them to this distribution" checkbox, then click *Next*
3. On *Specify the content destination* page, click the *Add* button to select the appropriate distribution points as shown in Figure 27



Specify the content destination

Content will be distributed to the following distribution points, distribution point groups, and the distribution point groups that are currently associated with collections.

Content destination:

Name	Description	Associations
W2019-SCCM.EXAMPL...	Distribution point	

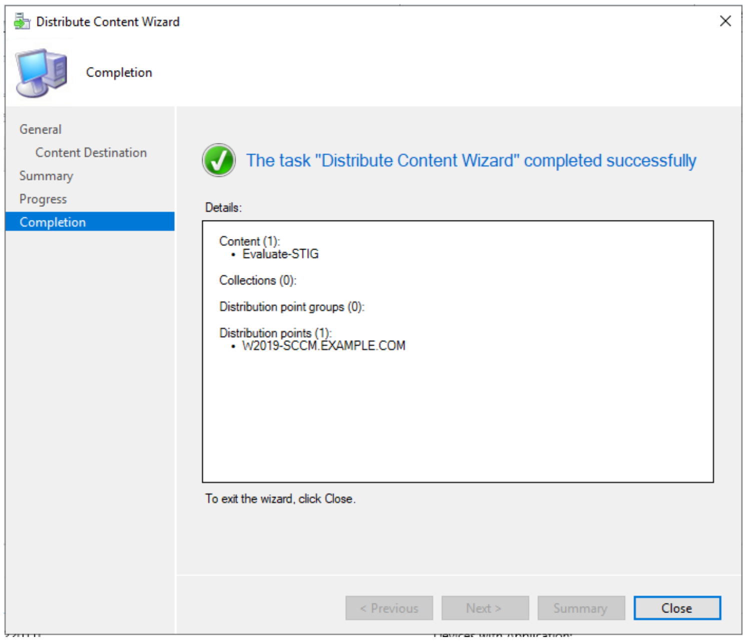
Figure 27: Distribution points for content

4. Click *Next* to progress to the confirmation page, if everything looks acceptable, click *Next*

Successful completion of the task should be met with a *The Distribute Content Wizard completed successfully* message, similar to Figure 28. *Evaluate-STIG* will now be distributed to the clients. From here, the application needs to be deployed to selected clients.

1. Right click the *Evaluate-STIG* application and select *Deploy* or click the *Deploy* button under the Home tab in the menu ribbon, shown in Figure 29
2. Within the *Deploy Software Wizard* click the *Browse* button associated with "Collection" (Figure 30)
    a. Change the *Select Collection* dropdown from "User Collections" to "Device Collections" and select the desired device collection to deploy the content to (Figure 31)
3. Click *Next* and choose the appropriate Distribution Points or Distribution Collections, then click *Next* again
4. On *Specify settings to control how this software is deployed*, ensure "Action" is set to *Install* and "Purpose" is set to *Required* as shown in Figure 32
    a. "Send wake-up packets" can be selected if the network is configured to support that function
5. Set the time to deploy as necessary, or leave as default to deploy the application as soon as possible

6. Leave the rest of the options within the wizard as default unless a specific business reason requires otherwise
7. On the deployment summary page, review the selected options then click *Next*
8. Provided no errors occurred, all deployment options should be met with green checkmarks, similar to Figure 33, and the deployment should appear under the deploy tab in SCCM (Figure 34)



The task "Distribute Content Wizard" completed successfully

Details:

Content (1):
- Evaluate-STIG

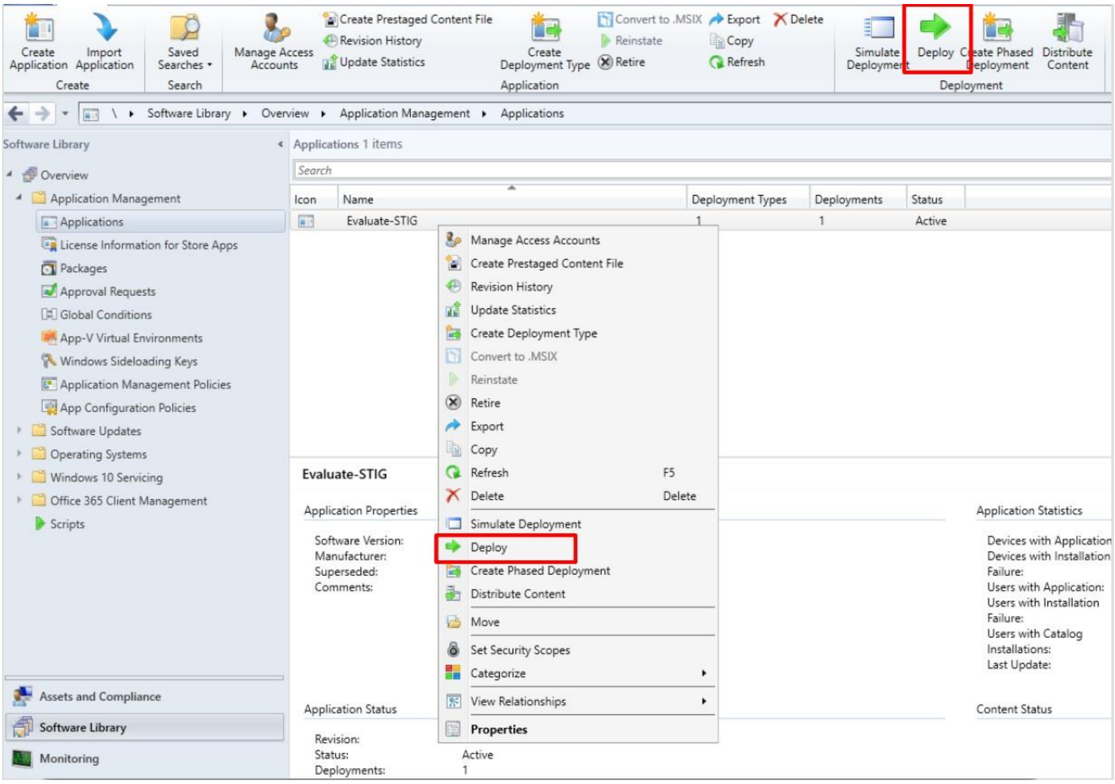
Collections (0):

Distribution point groups (0):

Distribution points (1):
- W2019-SCCM.EXAMPLE.COM

To exit the wizard, click Close.

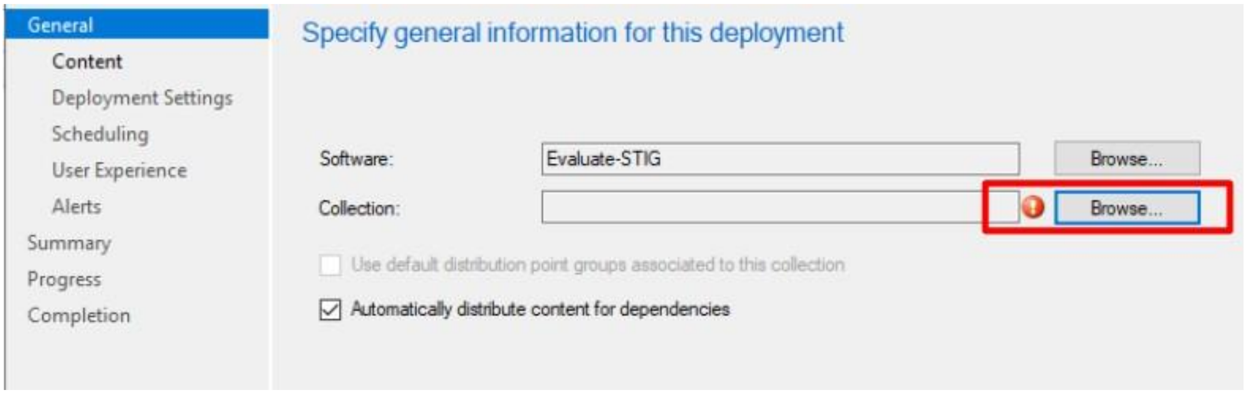
Figure 28: Successful completion of the Distribute Content Wizard



Icon	Name	Deployment Types	Deployments	Status
	Evaluate-STIG	1	1	Active

Revision:
Status: Active
Deployments: 1

Figure 29: Deploy button locations



Specify general information for this deployment

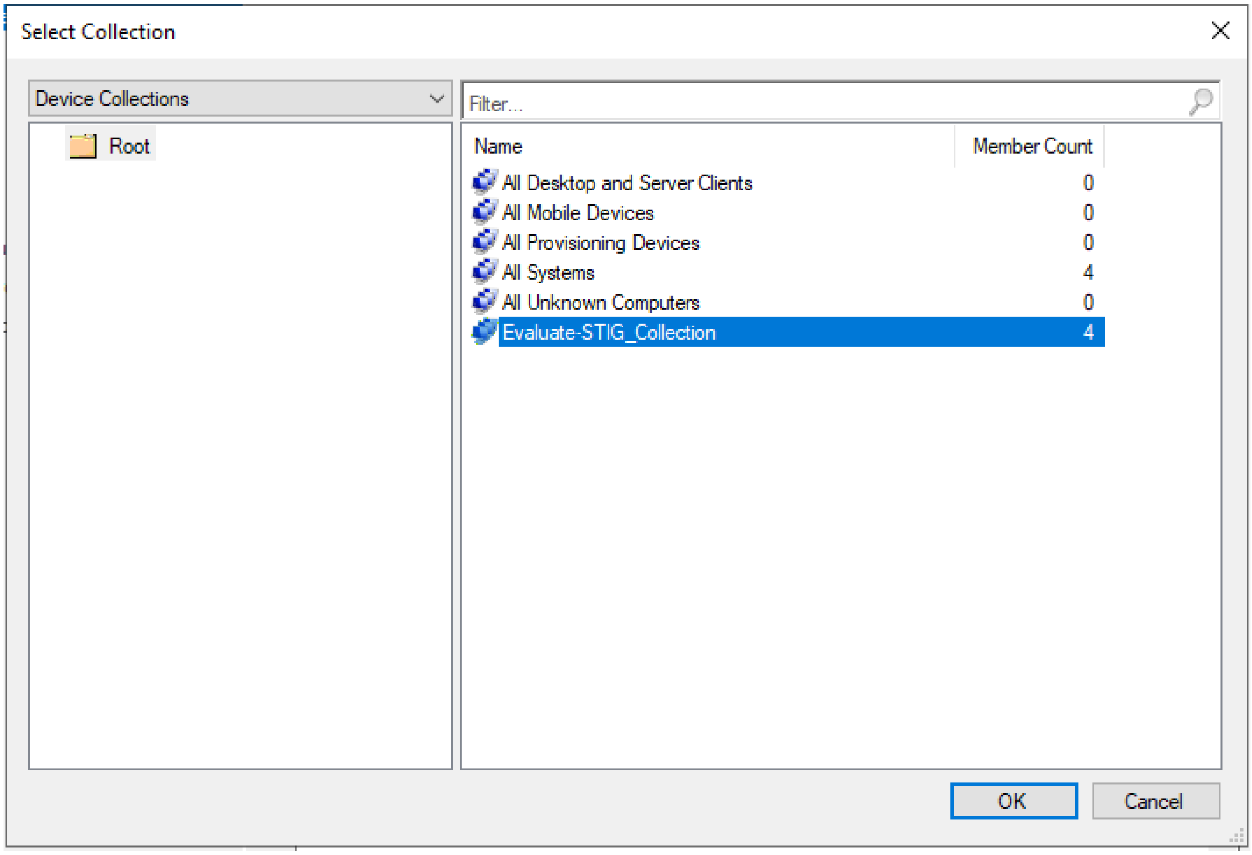
Software: Evaluate-STIG

Collection:

□ Use default distribution point groups associated to this collection

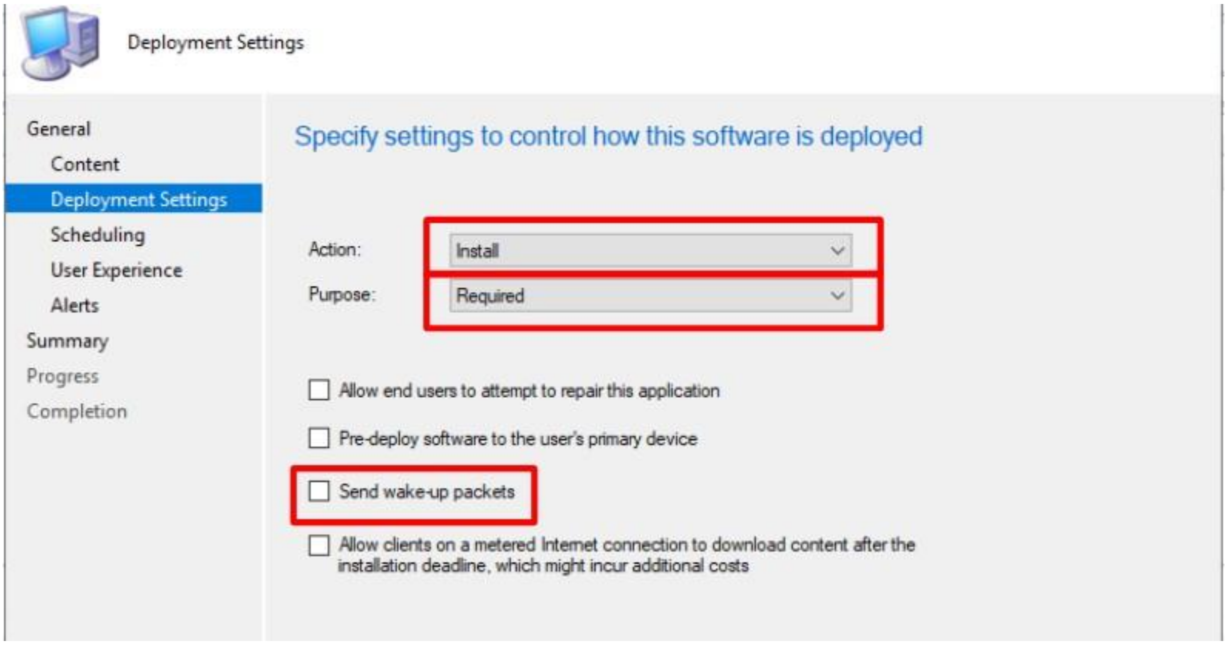
☑ Automatically distribute content for dependencies

Figure 30: Collection browse button



Name	Member Count
All Desktop and Server Clients	0
All Mobile Devices	0
All Provisioning Devices	0
All Systems	4
All Unknown Computers	0
Evaluate-STIG_Collection	4

Figure 31: Device collection settings



Specify settings to control how this software is deployed

Action: Install

Purpose: Required

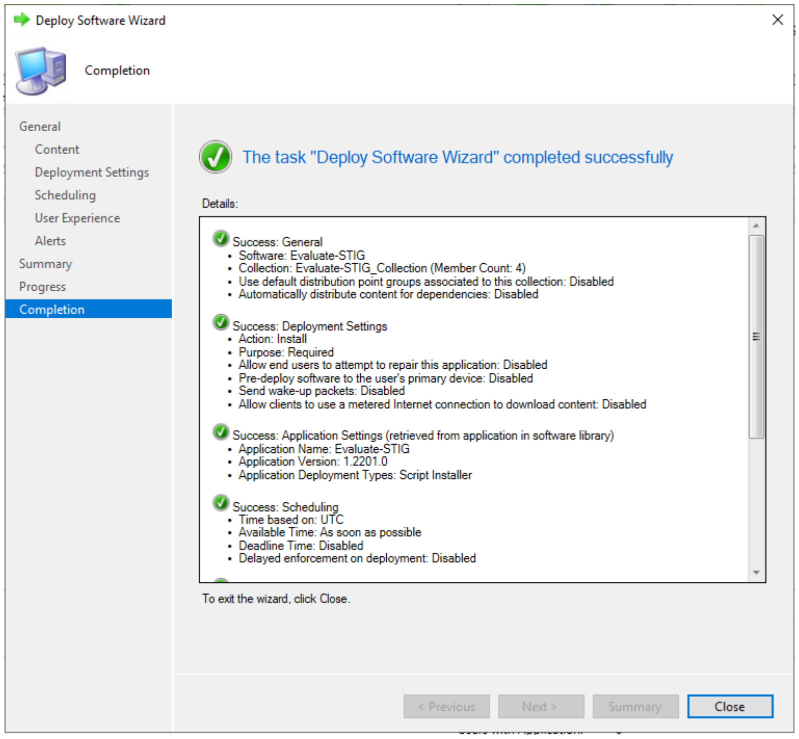
□ Allow end users to attempt to repair this application

□ Pre-deploy software to the user's primary device

□ Send wake-up packets

□ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs

Figure 32: Installation options for application deployment



The task "Deploy Software Wizard" completed successfully

Details:

Success: General
- Software: Evaluate-STIG
- Collection: Evaluate-STIG_Collection (Member Count: 4)
- Use default distribution point groups associated to this collection: Disabled
- Automatically distribute content for dependencies: Disabled

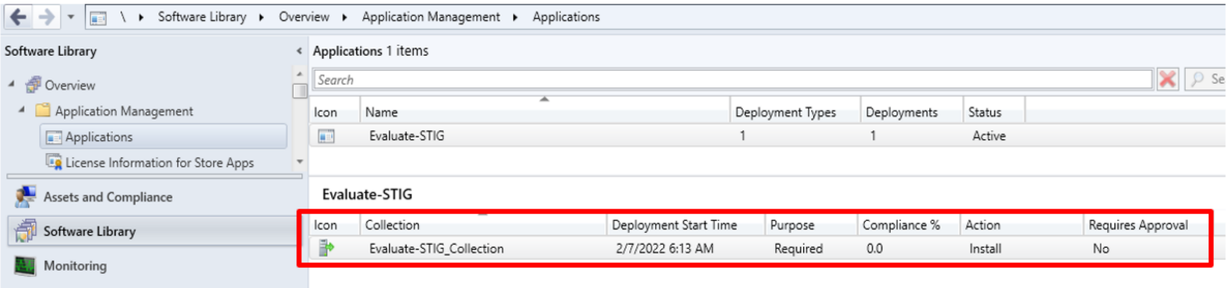
Success: Deployment Settings
- Action: Install
- Purpose: Required
- Allow end users to attempt to repair this application: Disabled
- Pre-deploy software to the user's primary device: Disabled
- Send wake-up packets: Disabled
- Allow clients to use a metered Internet connection to download content: Disabled

Success: Application Settings (retrieved from application in software library)
- Application Name: Evaluate-STIG
- Application Version: 1.2201.0
- Application Deployment Types: Script Installer

Success: Scheduling
- Time based on: UTC
- Available Time: As soon as possible
- Deadline Time: Disabled
- Delayed enforcement on deployment: Disabled

To exit the wizard, click Close.

Figure 33: Successful application deployment



Applications 1 items

Icon	Name	Deployment Types	Deployments	Status
	Evaluate-STIG	1	1	Active

Evaluate-STIG

Icon	Collection	Deployment Start Time	Purpose	Compliance %	Action	Requires Approval
	Evaluate-STIG_Collection	2/7/2022 6:13 AM	Required	0.0	Install	No

Figure 34: Deployment status

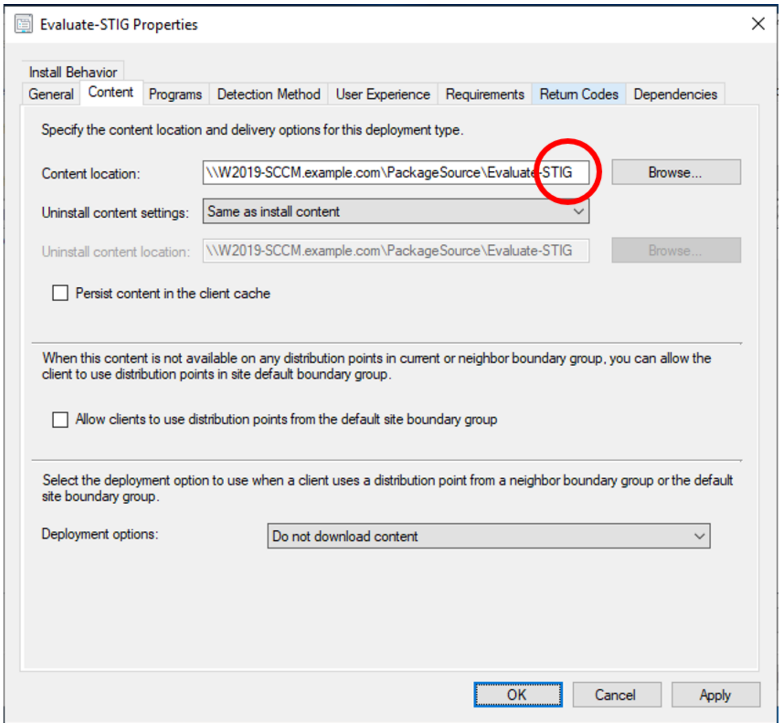
Updating the SCCM Application

STIGs are updated every quarter, and so too is the Evaluate-STIG tool. An updated version number is needed after this process or the SCCM application detection configuration will fail. The following are procedures for updating the SCCM application for an updated version of the Evaluate-STIG tool.

1. Obtain the latest Evaluate-STIG tool, see Getting the Latest Updates
2. Remove the `Evaluate-STIG` folder on `\\W2019-SCCM.example.com\PackageSource\`
3. Extract the updated Evaluate-STIG tool to `\\W2019-SCCM.example.com\PackageSource\Evaluate-STIG`
4. Open `.\Evaluate-STIG\Evaluate-STIG.ps1` within a text editor and note the value for the `$EvaluateStigVersion` variable as described in Prepare Evaluate-STIG for Deployment
5. In SCCM, right-click the *Evaluate-STIG* application and select Properties
6. Within the *Evaluate-STIG Properties*, under the *General Information* tab, update the “Software

Version”

7. Switch to the *Deployment Types* tab and, after selecting the Evaluate-STIG deployment type, click the *Edit…* button
8. Within the new *Evaluate-STIG Properties* deployment type dialog, switch to the *Content* tab
9. Remove the trailing backslash (“\”) from the “Content Location” field as shown in Figure 35
    a. This will force the content to be refreshed at the Distribution Points with the new Evaluate-STIG files



Specify the content location and delivery options for this deployment type.

Content location: \\W2019-SCCM.example.com\PackageSource\Evaluate-STIG

Uninstall content settings: Same as install content

Uninstall content location: \\W2019-SCCM.example.com\PackageSource\Evaluate-STIG

☐ Persist content in the client cache

When this content is not available on any distribution points in current or neighbor boundary group, you can allow the client to use distribution points in site default boundary group.

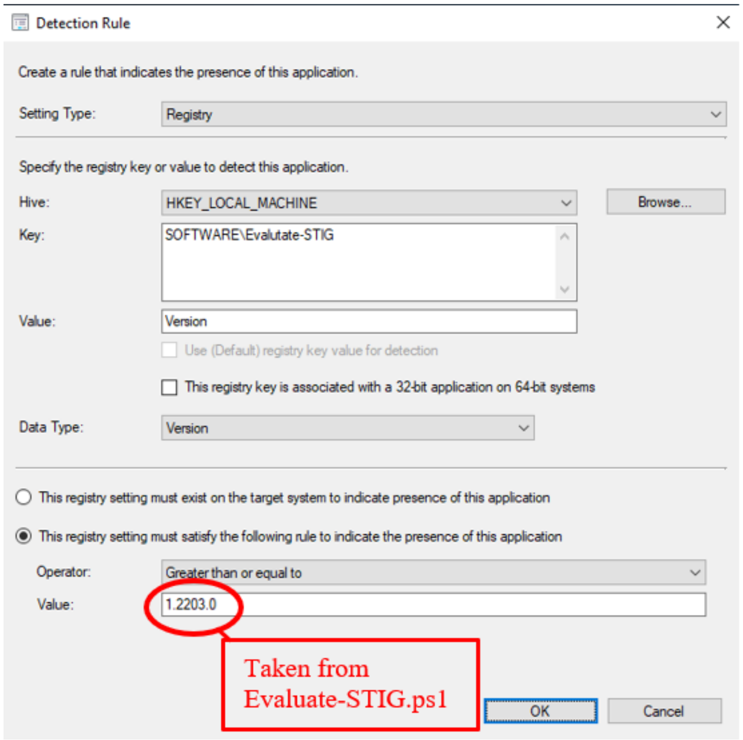
☐ Allow clients to use distribution points from the default site boundary group

Select the deployment option to use when a client uses a distribution point from a neighbor boundary group or the default site boundary group.

Deployment options: Do not download content

Figure 35: Updating the Content Location field in the Deployment Type dialog

10. Switch to the *Detection Method* tab and edit the clause
11. Update the value field under “This registry setting must satisfy the following rule to indicate the presence of this application” to match the latest Evaluate-STIG version, shown in Figure 36



Detection Rule

Create a rule that indicates the presence of this application.

Setting Type: Registry

Specify the registry key or value to detect this application.

Hive: HKEY_LOCAL_MACHINE

Key: SOFTWARE\Evalutate-STIG

Value: Version

□ Use (Default) registry key value for detection

□ This registry key is associated with a 32-bit application on 64-bit systems

Data Type: Version

○ This registry setting must exist on the target system to indicate presence of this application

● This registry setting must satisfy the following rule to indicate the presence of this application

Operator: Greater than or equal to

Value: 1.2203.0

Taken from Evaluate-STIG.ps1

Figure 36: Updating the version number within the Detection Rules dialog

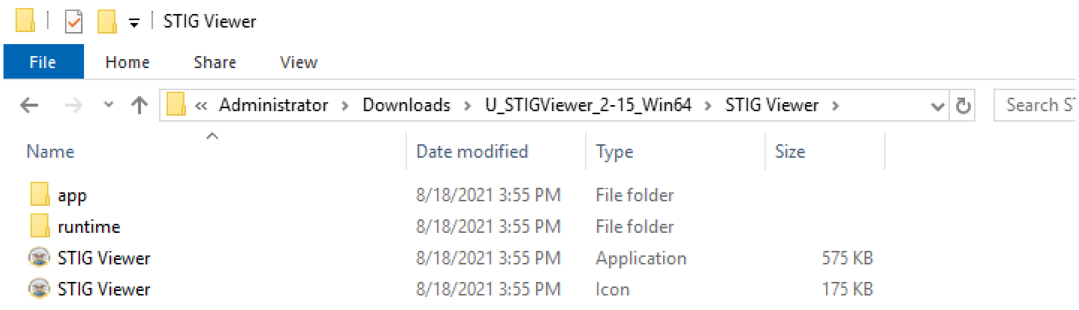
12. Click *OK* to accept the changes to the *Detection Method* rule
13. Click *OK* to accept the changes to the *Deployment Type*
14. Click *OK* to accept the changes to the Evaluate-STIG application

After some time, the Distribution Points should be updated with the updated content. Also, given the updated version number in the *Detection Method*, clients should, after the next policy refresh, automatically re-run Evaluate-STIG based on the deployment parameters.

Processing Results

After executing Evaluate-STIG in its various forms, the output has to be analyzed to determine overall system compliance with the STIGs. This processing is similar to the traditional approach of evaluating STIGs with SCAP, where the resultant CKL must be loaded into STIG Viewer and manually verified. After a scan, results are stored wherever the `-OutputPath` argument redirected to or *`C:\Users\Public\Documents\STIG_Compliance`*. To obtain the latest version of STIG Viewer, visit *https://cyber.mil/stigs/srg-stig-tools/*.

1. Open STIG Viewer
   a. Assuming the file was just retrieved from the cyber.mil website, the folder may resign in your Downloads folder
   b. Extract the files if needed
   c. Navigate into the *`.\U_STIGViewer_X-Y_Win64`* folder, where `X` and `Y` signify version numbers
   d. Double-click on .\STIG Viewer.exe to open STIG Viewer (reference Figure 37)


STIG Viewer
File
Home
Share
View
« Administrator › Downloads › U_STIGViewer_2-15_Win64 › STIG Viewer ›
Search S

Name	Date modified	Type	Size
app	8/18/2021 3:55 PM	File folder	
runtime	8/18/2021 3:55 PM	File folder	
STIG Viewer	8/18/2021 3:55 PM	Application	575 KB
STIG Viewer	8/18/2021 3:55 PM	Icon	175 KB

Figure 37: File to open STIG Viewer

2. Within STIG Viewer, select Checklist > "Open Checklist from File" (see Figure 38)

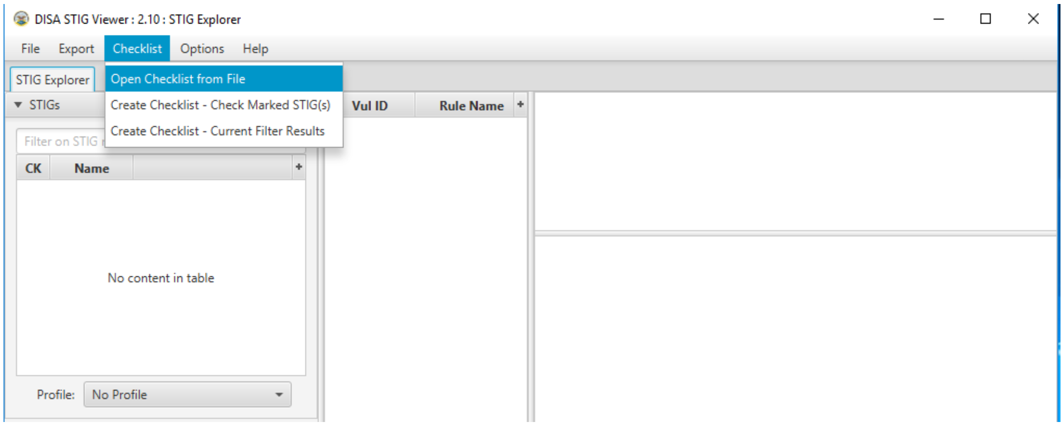
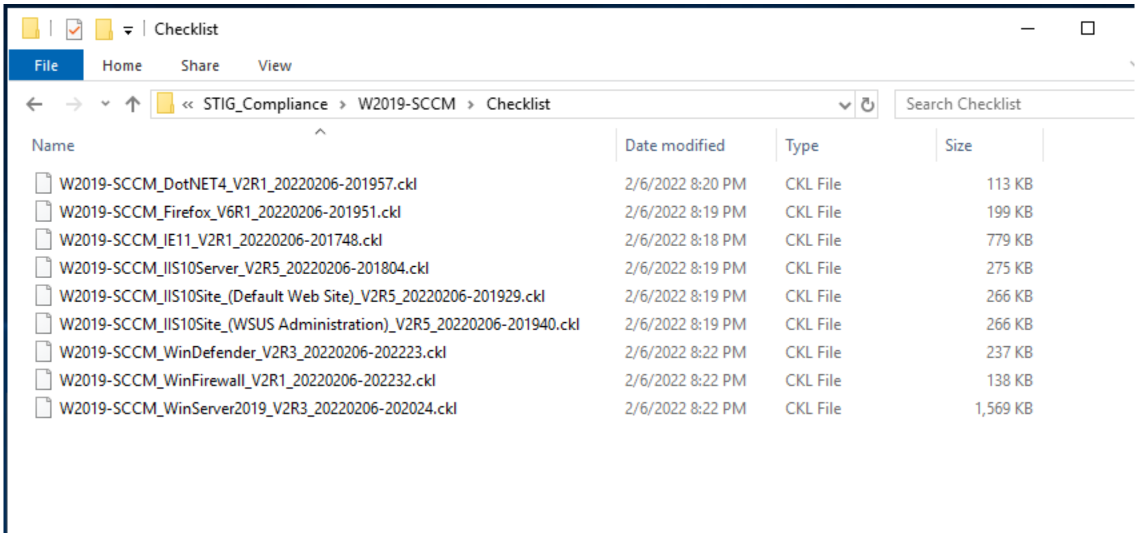


Figure 38: STIG Viewer, how to open a checklist from a file

3. Browse to the STIG_Compliance folder such as `C:\Windows\Temp\Evaluate-STIG\STIG_Compliance`
4. Select the computer to be checked, such as Win2019-SCCM
5. Navigate into the Checklist folder (the entire folder path is shown in Figure 39)
6. Select which STIG CKL to view and click *Open*
   a. Multiple CKL files can be selected at the same time by holding [Ctrl] and selecting them, a continuous range of files can be selected by selecting the start of the range, holding [Shift] and selecting the end of the range



Name	Date modified	Type	Size
W2019-SCCM_DotNET4_V2R1_20220206-201957.ckl	2/6/2022 8:20 PM	CKL File	113 KB
W2019-SCCM_Firefox_V6R1_20220206-201951.ckl	2/6/2022 8:19 PM	CKL File	199 KB
W2019-SCCM_IE11_V2R1_20220206-201748.ckl	2/6/2022 8:18 PM	CKL File	779 KB
W2019-SCCM_IIS10Server_V2R5_20220206-201804.ckl	2/6/2022 8:19 PM	CKL File	275 KB
W2019-SCCM_IIS10Site_(Default Web Site)_V2R5_20220206-201929.ckl	2/6/2022 8:19 PM	CKL File	266 KB
W2019-SCCM_IIS10Site_(WSUS Administration)_V2R5_20220206-201940.ckl	2/6/2022 8:19 PM	CKL File	266 KB
W2019-SCCM_WinDefender_V2R3_20220206-202223.ckl	2/6/2022 8:22 PM	CKL File	237 KB
W2019-SCCM_WinFirewall_V2R1_20220206-202232.ckl	2/6/2022 8:22 PM	CKL File	138 KB
W2019-SCCM_WinServer2019_V2R3_20220206-202024.ckl	2/6/2022 8:22 PM	CKL File	1,569 KB

Figure 39: Default location of STIG CKL files

Figure 40 provides an overview of the STIG Viewer layout and components. In reference to the figure, Table 4 explains each labeled section in more detail. To save any changes made in a given CKL, select File > "Save Checklist" or "Save Checklist As…".

Table 4: Descriptions of STIG Viewer sections

A Lists the open files, which can be navigated using the small dropdown button on the right side of the window. The menu toolbar at the top of the page is context specific which can be viewed by switching between open files.

B Provides an overall summary of the STIG compliance listing *NotAFinding*, *Open*, *Not_Applicable*, and *Not_Reviewed* statistics. Selecting one of the available category tabs acts as a filter.

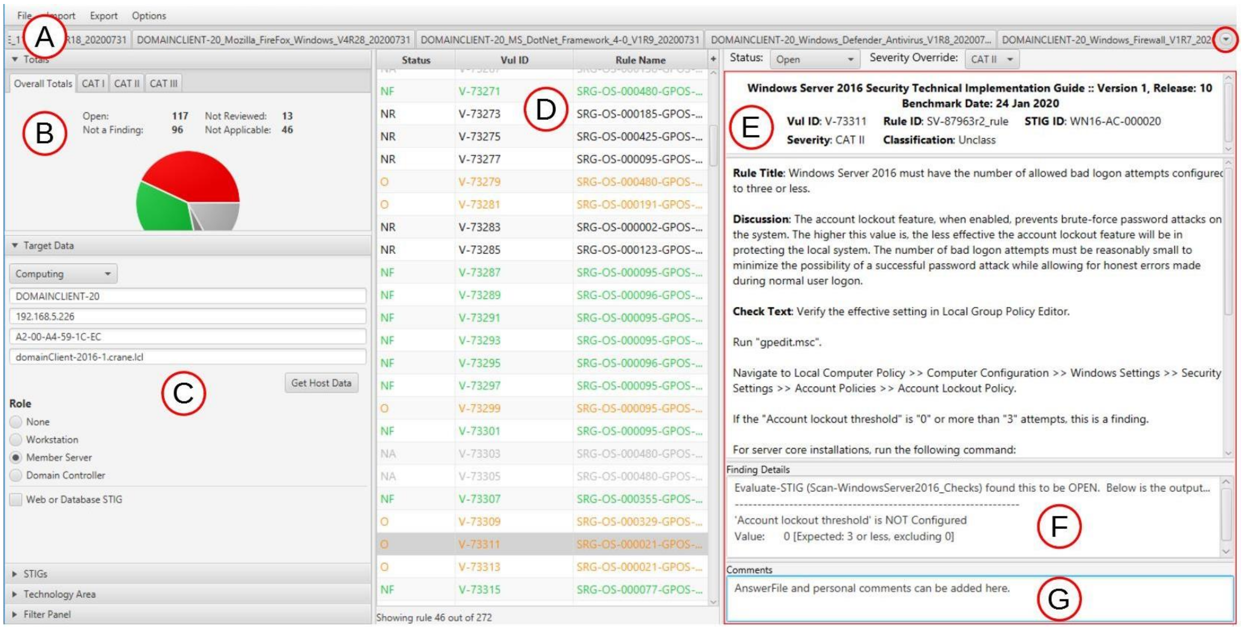
C Provides system details as determined by Evaluate-STIG including the system name, IP address, and role, among other information.

D The Vulnerability Table used to select which vulnerability details to view. The table is color coded and sortable by column to enable faster recognition of issues. Black is *Not_Reviewed*, gray is *Not_Applicable*, green is *NotAFinding*, bronze is *Open:CAT3*, orange is *Open:CAT2*, and red is *Open:CAT1*.

E The Vulnerability Details pane lists such information as VulnID, Severity, Rule Title, how to check for compliance, and how to fix a non-compliant system.

F The Finding Details pane provides any output generated by Evaluate-STIG to prove compliance or non-compliance. Even on *Not_Reviewed* items, it is possible to have relevant Finding Details information provided to quicken the auditor's assessment.

G The Comments pane is where Answer File comments and separate auditor comments can be placed and saved


File
Import
Export
Options
A
R18_20200731
DOMAINCLIENT-20_Mozilla_FireFox_Windows_V4R28_20200731
DOMAINCLIENT-20_MS_DotNet_Framework_4-0_V1R9_20200731
DOMAINCLIENT-20_Windows_Defender_Antivirus_V1R8_202007...
DOMAINCLIENT-20_Windows_Firewall_V1R7_202
Totals
Overall Totals
CAT I
CAT II
CAT III
B
Target Data
C
Get Host Data
D
E
F
G
STIGs
Technology Area
Filter Panel

Open: 117 Not Reviewed: 13
Not a Finding: 96 Not Applicable: 46

Computing
DOMAINCLIENT-20
192.168.5.226
A2-00-A4-59-1C-EC
domainClient-2016-1.crane.lcl

Role
○ None
○ Workstation
● Member Server
○ Domain Controller

□ Web or Database STIG

Status	Vul ID	Rule Name
NF	V-73271	SRG-OS-000480-GPOS-...
NR	V-73273	SRG-OS-000185-GPOS-...
NR	V-73275	SRG-OS-000425-GPOS-...
NR	V-73277	SRG-OS-000095-GPOS-...
O	V-73279	SRG-OS-000480-GPOS-...
O	V-73281	SRG-OS-000191-GPOS-...
NR	V-73283	SRG-OS-000002-GPOS-...
NR	V-73285	SRG-OS-000123-GPOS-...
NF	V-73287	SRG-OS-000095-GPOS-...
NF	V-73289	SRG-OS-000096-GPOS-...
NF	V-73291	SRG-OS-000095-GPOS-...
NF	V-73293	SRG-OS-000095-GPOS-...
NF	V-73295	SRG-OS-000096-GPOS-...
NF	V-73297	SRG-OS-000095-GPOS-...
O	V-73299	SRG-OS-000095-GPOS-...
NF	V-73301	SRG-OS-000095-GPOS-...
NA	V-73303	SRG-OS-000480-GPOS-...
NA	V-73305	SRG-OS-000480-GPOS-...
NF	V-73307	SRG-OS-000355-GPOS-...
O	V-73309	SRG-OS-000329-GPOS-...
O	V-73311	SRG-OS-000021-GPOS-...
O	V-73313	SRG-OS-000021-GPOS-...
NF	V-73315	SRG-OS-000077-GPOS-...

Showing rule 46 out of 272

Status: Open Severity Override: CAT II

Windows Server 2016 Security Technical Implementation Guide :: Version 1, Release: 10
Benchmark Date: 24 Jan 2020

Vul ID: V-73311 **Rule ID**: SV-87963r2_rule **STIG ID**: WN16-AC-000020
Severity: CAT II **Classification**: Unclass

Rule Title: Windows Server 2016 must have the number of allowed bad logon attempts configured to three or less.

Discussion: The account lockout feature, when enabled, prevents brute-force password attacks on the system. The higher this value is, the less effective the account lockout feature will be in protecting the local system. The number of bad logon attempts must be reasonably small to minimize the possibility of a successful password attack while allowing for honest errors made during normal user logon.

Check Text: Verify the effective setting in Local Group Policy Editor.

Run "gpedit.msc".

Navigate to Local Computer Policy >> Computer Configuration >> Windows Settings >> Security Settings >> Account Policies >> Account Lockout Policy.

If the "Account lockout threshold" is "0" or more than "3" attempts, this is a finding.

For server core installations, run the following command:

Finding Details

```
Evaluate-STIG (Scan-WindowsServer2016_Checks) found this to be OPEN.  Below is the output...
--------------------------------------------------------------
'Account lockout threshold' is NOT Configured
Value:     0 [Expected: 3 or less, excluding 0]
```

Comments

AnswerFile and personal comments can be added here.

Figure 40: STIG Viewer layout and components

Troubleshooting

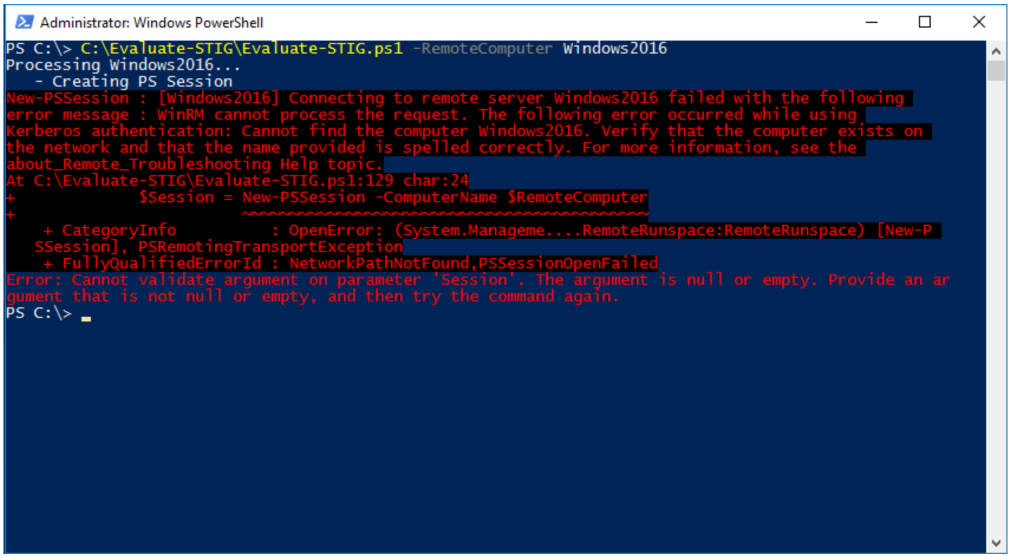
This section is to hold any troubleshooting issues that may occur regarding the Evaluate-STIG tool or the processes discussed in this document.

WinRM cannot process the request

Date First Noticed: 06/2020

Problem Description:

When attempting to run Evaluate-STIG from a PowerShell prompt with the −RemoteComputer (now −ComputerName) argument, an error occurs as shown in Figure 41.



```
PS C:\> C:\Evaluate-STIG\Evaluate-STIG.ps1 -RemoteComputer Windows2016
Processing Windows2016...
   - Creating PS Session
New-PSSession : [Windows2016] Connecting to remote server Windows2016 failed with the following
error message : WinRM cannot process the request. The following error occurred while using
Kerberos authentication: Cannot find the computer Windows2016. Verify that the computer exists on
the network and that the name provided is spelled correctly. For more information, see the
about_Remote_Troubleshooting Help topic.
At C:\Evaluate-STIG\Evaluate-STIG.ps1:129 char:24
+             $Session = New-PSSession -ComputerName $RemoteComputer
+                        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OpenError: (System.Manageme....RemoteRunspace:RemoteRunspace) [New-P
   SSession], PSRemotingTransportException
    + FullyQualifiedErrorId : NetworkPathNotFound,PSSessionOpenFailed
Error: Cannot validate argument on parameter 'Session'. The argument is null or empty. Provide an ar
gument that is not null or empty, and then try the command again.
PS C:\>
```

Figure 41: Evaluate-STIG -ComputerName WinRM error

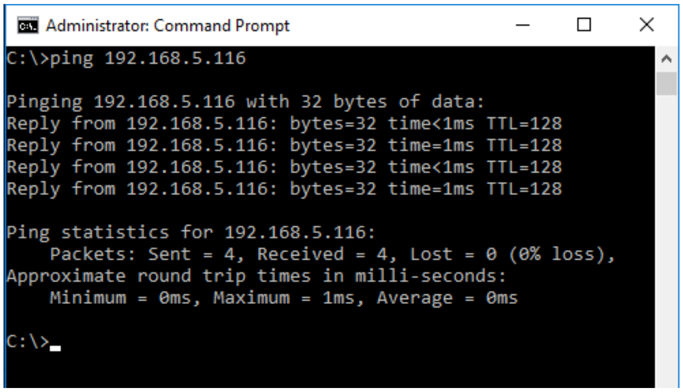
Possible Causes:

1. The remote computer is not part of the domain. This causes a problem with authentication between the two systems as specified in the error message.
2. The initiating computer cannot resolve the hostname of the remote computer. This is caused when the remote computer is missing a DNS entry either in the DNS server or Windows Host file
3. Invalid authentication tokens. The Kerberos authentication token for the remote or initiating computer may be invalid.

Possible Solutions:

1. Computers not part of the same domain cannot use the -ComputerName argument at this time. Instead the remote computer should be added to the SCCM inventory and configured via that method.

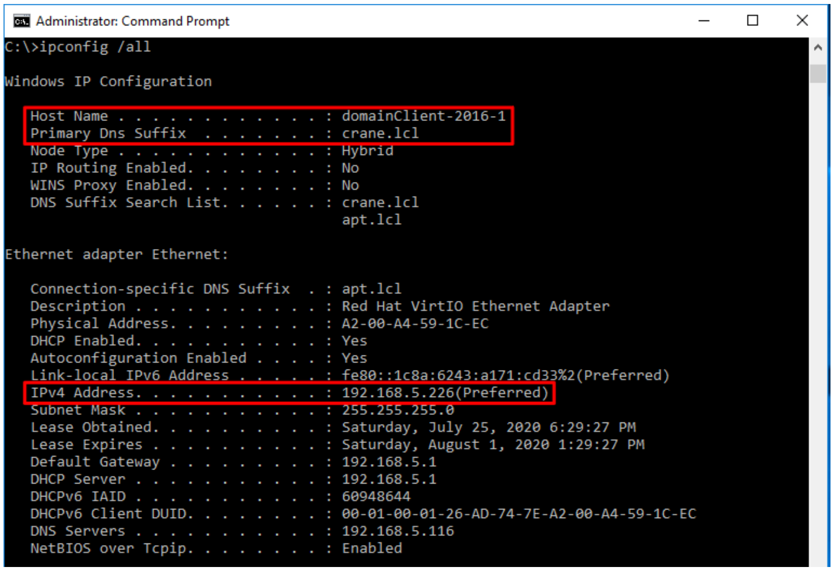
2. A simple test to check if a hostname or IP address are correct is to "ping" the remote computer from the initiating computer. This will not always work, however, depending on firewall rules and access controls.
    a. Open command prompt (CMD) by selecting Start > Windows System > `Command Prompt`
    b. Enter the "ping" command for the remote computer such as: `ping Windows2016`
    c. If no response is received (request timed out), attempt to ping the remote computer with its IP address: `ping 192.168.5.116`
    d. If this succeeds (Figure 42), it is likely the remote computer does not have a valid DNS record. If it fails, a firewall may be in the way or no route exist to the host.



```
C:\>ping 192.168.5.116

Pinging 192.168.5.116 with 32 bytes of data:
Reply from 192.168.5.116: bytes=32 time<1ms TTL=128
Reply from 192.168.5.116: bytes=32 time=1ms TTL=128
Reply from 192.168.5.116: bytes=32 time<1ms TTL=128
Reply from 192.168.5.116: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.5.116:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>_
```

Figure 42: Example of a successful "ping" using an IP address

3. To check if an invalid authentication token is the issue, consult with a server administrator only after attempting the previous two troubleshooting suggestions. Provide any useful relevant information including the initiating computer's hostname and IP address, the remote computer's hostname and IP address (if known), and the results from Possible Solutions 1 and 2.
    a. A computer's hostname and IP address can be obtained within `command prompt` using the command `ipconfig /all` (see Figure 43 for information extraction)



```
C:\>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : domainClient-2016-1
   Primary Dns Suffix  . . . . . . . : crane.lcl
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : crane.lcl
                                       apt.lcl

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : apt.lcl
   Description . . . . . . . . . . . : Red Hat VirtIO Ethernet Adapter
   Physical Address. . . . . . . . . : A2-00-A4-59-1C-EC
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::1c8a:6243:a171:cd33%2(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.5.226(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, July 25, 2020 6:29:27 PM
   Lease Expires . . . . . . . . . . : Saturday, August 1, 2020 1:29:27 PM
   Default Gateway . . . . . . . . . : 192.168.5.1
   DHCP Server . . . . . . . . . . . : 192.168.5.1
   DHCPv6 IAID . . . . . . . . . . . : 60948644
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-26-AD-74-7E-A2-00-A4-59-1C-EC
   DNS Servers . . . . . . . . . . . : 192.168.5.116
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

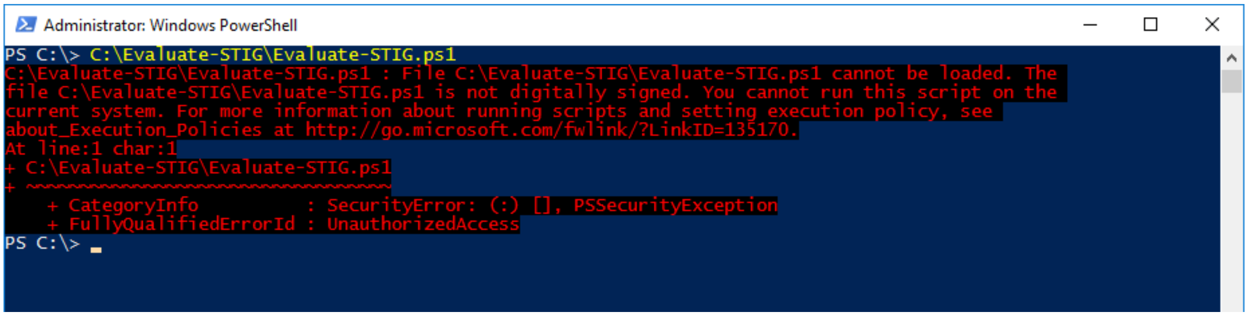
Figure 43: Gathering the hostname and IP address for a local computer system

Evaluate-STIG.ps1 cannot be loaded or is not digitally signed

Date First Noticed: 06/2020

Problem Description:

When attempting to run Evaluate-STIG from a PowerShell prompt, an error occurs stating Evaluate-STIG.ps1 is not digitally signed, such as in Figure 44.



```
PS C:\> C:\Evaluate-STIG\Evaluate-STIG.ps1
C:\Evaluate-STIG\Evaluate-STIG.ps1 : File C:\Evaluate-STIG\Evaluate-STIG.ps1 cannot be loaded. The
file C:\Evaluate-STIG\Evaluate-STIG.ps1 is not digitally signed. You cannot run this script on the
current system. For more information about running scripts and setting execution policy, see
about_Execution_Policies at http://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ C:\Evaluate-STIG\Evaluate-STIG.ps1
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\> _
```

Figure 44: PowerShell error due to Evaluate-STIG not being digitally signed

Possible Solution:

Please refer to Prerequisites

PowerShell Configuration for procedures on how to remedy this issue.

1. If the previous troubleshooting suggestion does not remedy the issue, please contact your system administrator.

Evaluate-STIG fails to remote scan Linux machine using SSH

Date First Noticed: 07/2021

Problem Description:

Following an update to Windows 10 20H2, remote scanning of Linux machines via PowerShell and SSH has failed if an SSH logon banner is enabled. Issue #5142 and Issue #15759 of the PowerShell project states using a Banner option within sshd_config causes the connection to fail. Prior to KB5003173 for 20H2, the error could be negated with error handling, however the issue is now a terminating error. The error was first reported to the PowerShell developers in 2017.

Possible Solution:

⇨ ***NOTE:*** This procedure is against STIG policy, and should therefore be done only with prior approval and as a temporary measure for scanning purposes only.

1. On the Linux system, navigate to the sshd_config file
2. Using a text editor, comment out the line specifying the use of a banner
3. Restart the SSH daemon
4. Scan the system using Evaluate-STIG as documented
5. After the scan, uncomment the banner option

6. Restart the SSH daemon to have the changes take effect

Appendix A: Exit Codes

For troubleshooting purposes, Evaluate-STIG may produce the following exit codes:

- **10009992** – PowerShell version is not supported.
- **10009995** – A timeout occurred. Check `%WINDIR%\Temp\Evaluate-STIG\Evaluate-STIG.log`
- **10010001** – An error occurred. Check `%WINDIR%\Temp\Evaluate-STIG\Evaluate-STIG.log`