Evaluate-STIG

User Guide

1.2507.6

NAVSEA
WARFARE CENTERS

Table of Contents

1 Introduction

Evaluate-STIG is a PowerShell tool for automating Security Technical Implementation Guide (STIG) scans and can optionally output results to STIG Viewer compatible checklist files (both CKL and CKLB formats). It is only used for documenting STIG compliance state and not for configuring to STIG requirements. Evaluate-STIG can greatly reduce or eliminate the manual efforts typically required for documenting compliance while providing more complete, accurate, and consistent documentation. Labor efforts that previously could consume hours or days can now be completed in minutes.

Evaluate-STIG, designed with automation as the priority, detects which of the supported STIGs are required for the asset and reduces the risk of missed STIGs. It can scan both the local system as well as remote assets. For larger networks, it can be deployed by configuration management tools (e.g., Microsoft Configuration Manager, IBM BigFix, PDQ Deploy, Ansible, etc.) to maximize automation efficiencies.

Some of our users:

2 System Requirements

2.1 Supported Operating Systems

Evaluate-STIG may be run on the following operating systems:

Amazon Linux 2023	Oracle Linux 7	Windows 7 *	Windows Server 2019
Canonical Ubuntu 16.04	Oracle Linux 8	Windows 10	Windows Server 2022
Canonical Ubuntu 18.04	Oracle Linux 9	Windows 11	Windows Server 2025
Canonical Ubuntu 20.04	Red Hat Enterprise Linux 7	Windows Server 2008 R2 *	
Canonical Ubuntu 22.04	Red Hat Enterprise Linux 8	Windows Server 2012 / R2 *	
Canonical Ubuntu 24.04	Red Hat Enterprise Linux 9	Windows Server 2016	

* Requires Windows Management Framework 5.1

2.2 Prerequisites

Windows	• PowerShell 5.1 | PowerShell 7.3 or greater (PowerShell 6 is not supported)
	• Compatible PowerShell Execution Policy (depends on code signing certificate trust)
	  ○ Recommend adding included DOD-issued code signing certificate to the following store: `Local Machine\Trusted Publishers` store
	• For MS SQL:
	  ○ Administrator level permission to both the operating system and SQL instance/database

Linux	• Libraries **`libicu`**, **`lshw`**, **`dmidecode`**, and **`bc`** must be installed
	• When running from:
	  ○ **`PowerShell`** direct: PowerShell 7.3 or greater must be installed (note RHEL7 only supports PowerShell 7.3)
	  ○ **`Bash`**: Supported PowerShell archive (7.3 or greater) renamed to **`powershell.tar.gz`** located within the Evaluate-STIG folder. Use --DownloadPS or perform this manually.
	• If using **`fapolicyd`** for application whitelisting, **`pwsh`** and **`libhostfxr.so`** must be allowed.

Support Scripts

Included in the `Evaluate-STIG\Prerequisites` folder are scripts to help ensure systems meet the above prerequisites:

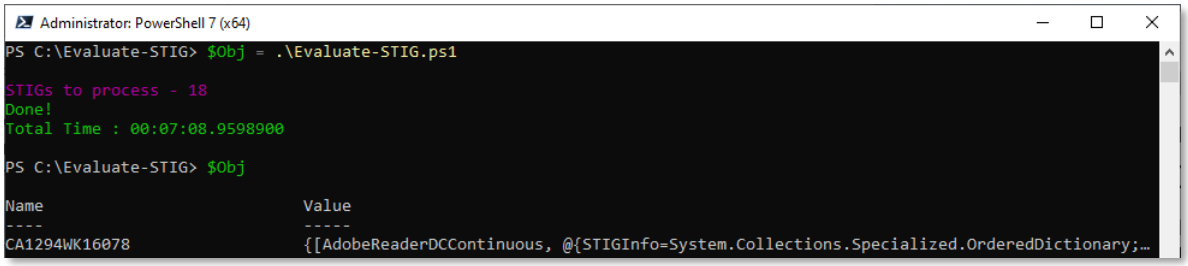
- **Windows**
  - **`Test-Prerequisites.bat`** – To verify execution policy, certificate trust, and that no files have the blocked attribute.
  - **`Import-Certificates.bat`** – To import the code signing certificate chain into the correct stores.
- **Linux**
  - **`Test-Prerequisites.sh`** – To verify **`libicu`** and **`lshow`** are installed and remind about whitelisting if **`fapolicyd`** is installed.

3 Usage

Evaluate-STIG is primarily designed as a command-line tool that should be executed from an elevated PowerShell prompt. In the simplest form, a default, full scan of the local machine is performed by calling the `Evaluate-STIG.ps1` file with no options. Below example will store the output into a variable named `$Obj`:

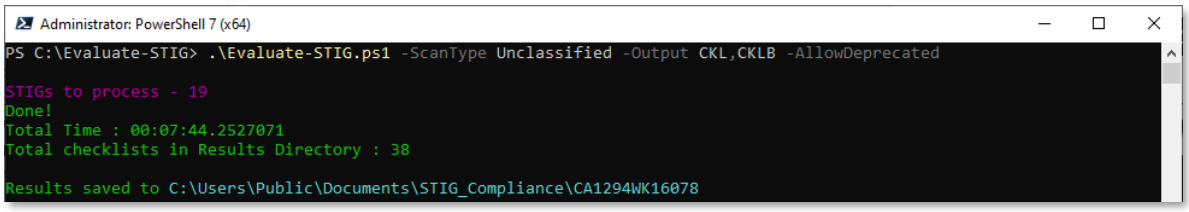
```
PS C:\Evaluate-STIG> $Obj = .\Evaluate-STIG.ps1
```

Which can be examined after:



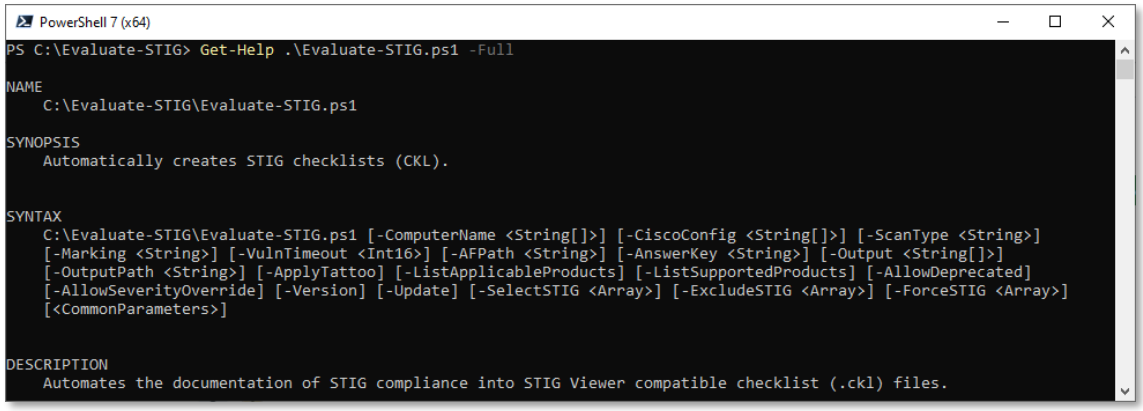
Add parameters as necessary to customize the scan to your needs. Below example will perform an unclassified (default) scan, saving the results as both `.CKL` and `.CKLB` files, and enable deprecated STIG scanning:

```
PS C:\Evaluate-STIG> .\Evaluate-STIG.ps1 -ScanType Unclassified -Output CKL,CKLB -AllowDeprecated
```



Use `Get-Help` to display Evaluate-STIG's help information:

```
PS C:\Evaluate-STIG> Get-Help .\Evaluate-STIG.ps1 -Full
```



3.1 PowerShell Parameters

Scan Settings

The parameters below may be used for customizing scan settings by adding to your command line:

-ScanType

Parameter Type:	<String>
Description:	Classification of asset being scanned. This is solely for achieving a `Not Applicable` status for checks that are classification dependent.
Valid Entries:	• **Unclassified** • **Classified**
Default:	"Unclassified"
Example:	

```
.\Evaluate-STIG.ps1 -ScanType Unclassified
```

-Marking

Parameter Type:	<String>
Description:	Use to optionally set the **Marking** field in *.CKL* and **Target Data Classification** field in *.CKLB* output files.
Example:	

```
.\Evaluate-STIG.ps1 -Marking MyMarking
```

-TargetComments

Parameter Type:	<String>
Description:	Use to optionally set the `TargetComments` field in *.CKL* and *.CKLB* output files.
Example:	

```
.\Evaluate-STIG.ps1 -TargetComments "My comments"
```

-VulnTimeout

Parameter Type:	<Int16> [Valid Range: 1-1440]
Description:	Set the maximum time in minutes allowed for a singular Vuln ID check to run.
Default:	"15"
Example:	

```
.\Evaluate-STIG.ps1 -VulnTimeout 30
```

-FileSearchTimeout

Parameter Type:	<Int16> [Valid Range: 1-1440]
Description:	Set the maximum time in minutes allowed for the file type pre-scan search to run. For STIGs requiring a scan of all hard disks for specific file types (e.g. `.pfx`).
Default:	"240"
Example:	

```
.\Evaluate-STIG.ps1 -FileSearchTimeout 120
```

-AnswerKey

Parameter Type:	<String>
Description:	Use to direct 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 this parameter is not specified, Evaluate-STIG will still attempt to use the "DEFAULT" key if configured in the answer file for that Vuln ID.**
Default:	"DEFAULT"
Example:	

```
.\Evaluate-STIG.ps1 -AnswerKey MyKey
```

-AFPath

Parameter Type:	<String>
Description:	Specify location of folder with answer files. May be a local or UNC path. Default location is ".\Evaluate-STIG\AnswerFiles". **Must point to a folder of answer files and not an answer file itself. If this parameter is not specified, Evaluate-STIG will still attempt to use the "DEFAULT" key if configured in the answer file for that Vuln ID.**
Default:	".\Evaluate-STIG\AnswerFiles\"
Example:	

```
.\Evaluate-STIG.ps1 -AFPath \\Server01\AnswerFiles\
```

-Output

Parameter Type:	<String[]>
Description:	Specify outputs to generate. Multiple outputs may be specified though comma separation.
Valid Entries:	• **Console:** Results will be returned to the console as a PowerShell object. When outputting to the console, it is recommended results be stored in a variable so they can be walked after (e.g. `$Obj = .\Evaluate-STIG.ps1`). • **CKL:** Results will be saved to singular *.CKL* files per STIG. Compatible with STIG Viewer 2.17 • **CKLB:** Results will be saved to singular *.CKLB* files per STIG. Compatible with STIG Viewer 3.x. • **CSV:** Results will be saved to singular *.CSV* files per STIG. • **XCCDF:** Results will be saved to singular *.XCCDF.XML* files per STIG. • **CombinedCKL:** Results will be saved to a multi-STIG *.CKL* file. *Note that some STIGs are not combinable and will have their results saved to singular .CKL files alongside the combined CKL.* • **CombinedCKLB:** Results will be saved to a multi-STIG *.CKLB* file. *Note that some STIGs are not combinable and will have their results saved to singular .CKLB files alongside the combined CKLB.* • **CombinedCSV:** Results will be saved to a multi-STIG *.CSV* file. *Note that some STIGs are not combinable and will have their results saved to singular .CSV files alongside the combined CSV.* • **OQE:** Generate Objective Quality Evidence (OQE) artifacts on Windows systems. • **Summary:** Generate summary reports of the scans. • **STIGManager:** Send results to a STIG Manager instance. See STIG Manager for more. • **Splunk:** Send results to a Splunk HTTP Event Collector. See Splunk for more.
Default:	"Console"
Example:	

```
$Obj = .\Evaluate-STIG.ps1 -Output Console,CombinedCKL,CKLB
```

-JSON

Parameter Type:	<Switch>
Description:	Output Console Object in JSON Format using OutputPayload Options. JSON Objects are based on Group ID.
Requires:	-Output Console
Example:	

```
$Obj =.\Evaluate-STIG.ps1 -Output Console -JSON
```

-OutputPayload

Parameter Type:	<Array>
Description:	Specify which fields to output when outputting to CSV, JSON, or Splunk. Order of fields will be retained from command line. For multiple, separate with commas. Default is all fields. Requires -Output CSV|CombinedCSV|Splunk or -JSON.
Valid Entries:	• **Title:** Name of the STIG. • **Version:** Version of the STIG. • **ReleaseDate:** Release date of the STIG. • **Classification:** Classification of the STIG. • **HostName:** Asset name. • **Site:** Site name for web server STIGs. • **Instance:** Instance or database name for database STIGs. • **IP:** IP address of asset. • **MAC:** MAC address of asset. • **FQDN:** Fully Qualified Domain Name of asset. • **Role:** Role of asset (Workstation|Member Server|Domain Controller|None). • **GroupID:** Group ID (vulnerability ID) from STIG. • **GroupTitle:** Group Title (Rule Name) from STIG. • **RuleID:** Rule ID of Group ID from the STIG. • **STIGID:** STIG ID of Group ID from the STIG. • **Severity:** Severity of Group ID from the STIG. • **SeverityOverride:** SeverityOverride from result if used. • **Justification:** Justification for the SeverityOverride from result if used. • **LegacyIDs:** Legacy IDs for the Group ID from the STIG. • **RuleTitle:** Rule Title of the Group ID from the STIG. • **Discussion:** Discussion for the Group ID from the STIG. • **CheckText:** Check Text for the Group ID from the STIG. • **FixText:** Fix Text for the Group ID from the STIG. • **CCI:** CCI reference(s) for the Group ID from the STIG. • **Status:** Status of check. • **FindingDetails:** Finding Details for check. • **Comments:** Comments for check. • **ESVersion:** Version of Evaluate-STIG used. • **StartTime:** Time that the scan for the STIG began.
Requires:	-Output CSV|CombinedCSV|Splunk or -JSON
Example:	

```
.\Evaluate-STIG.ps1 -Output CSV -OutputPayload HostName,Title,GroupID,Status,FindingDetails
```

-OutputPath

Parameter Type:	<String>
Description:	Specify location to save files produced by -Output. May be a local or UNC path. A folder for the machine name will be created automatically in [OutputPath].
Default:	• **Windows:** "C:\Users\Public\Documents\STIG_Compliance" • **Linux:** "/opt/STIG_Compliance"
Requires:	-Output <CKL|CKLB|CSV|XCCDF|CombinedCKL|CombinedCKLB|CombinedCSV|Summary|OQE>
Example:	

```
.\Evaluate-STIG.ps1 -Output CombinedCKL,CKLB -OutputPath \\Server01\MyShare\
```

-PreviousToKeep

Parameter Type:	<Int16>
Description:	Number of previous scan session outputs to retain in -OutputPath. Scan outputs are the items requested with -Output. Retained results will be moved to a **[OutputPath]\Previous\[Results Date-Time]** folder. • Using **-PreviousToKeep 0** will remove all previous results and only retain current results. • Using a negative value (e.g. **-PreviousToKeep -1**) will keep all previous results.
Default:	"1"
Requires:	-Output
Example:	

```
.\Evaluate-STIG.ps1 -Output CombinedCKL,CKLB -PreviousToKeep 5
```

-SelectSTIG

Parameter Type:	<Array>
Description:	By default, Evaluate-STIG will scan for all supported STIGs applicable to the asset. This parameter is to limit the scan to only certain STIG(s). Use [Tab] or [CTRL + Space] to properly select STIG(s) by their short names. Multiple STIGs may be selected using comma separation. This option cannot be used with -ExcludeSTIG. STIG short names are identified in the -ListSupportedProducts.
Example:	

```
.\Evaluate-STIG.ps1 -SelectSTIG Chrome,MSEdge
```

-SelectVuln

Parameter Type:	<Array>
Description:	Specify which vulnerability IDs to include in scan. Entries must be in the V-#### format as listed in the STIG. For multiple vulnerability IDs, separate with commas. If outputting to *.CKL* or *.CKLB*, results will be saved to a "_Partial" folder under [OutputPath].
Requires:	-SelectSTIG
Example:	

```
.\Evaluate-STIG.ps1 -SelectSTIG Chrome,MSEdge -SelectVuln V-221558,V-235719
```

-ExcludeVuln

Parameter Type:	<Array>
Description:	Specify which vulnerability IDs to exclude from scan. Entries must be in the V-#### format as listed in the STIG. For multiple vulnerability IDs, separate with commas. ***If a vulnerability ID is both selected and excluded, -ExcludeVuln wins and the vulnerability ID will not be scanned.***
Requires:	-SelectSTIG
Example:	

```
.\Evaluate-STIG.ps1 -SelectSTIG Chrome,MSEdge -ExcludeVuln V-221558,V-235719
```

-ExcludeSTIG

Parameter Type:	<Array>
Description:	This parameter is to exclude certain STIG(s) from a scan and scan for all other applicable STIGs. Use [Tab] or [CTRL + Space] to properly exclude STIG(s) by their short names. Multiple STIGs may be excluded through comma separation. This option cannot be used with -SelectSTIG. STIG short names are identified in the -ListSupportedProducts.
Example:	

```
.\Evaluate-STIG.ps1 -ExcludeSTIG Chrome,MSEdge
```

-ForceSTIG

WARNING Evaluate-STIG results are not guaranteed with this option. Use at own risk.

Parameter Type:	<Array>
Description:	By default, Evaluate-STIG will determine which STIGs are applicable to the asset based on defined criteria for that STIG. This parameter is to force certain STIG(s) to be scanned regardless of Evaluate-STIG's applicability check. Use [Tab] or [CTRL + Space] to list STIG(s) by their short names. Multiple STIGs may be forced using comma separation. STIG short names are identified in the -ListSupportedProducts option. ***If a STIG is both excluded and forced, -ForceSTIG will win and the STIG will be scanned.*** This parameter is required if an unsupported STIG is to be included in the scan.
Example:	

```
.\Evaluate-STIG.ps1 -ForceSTIG ADDomain,JavaJRE8Windows
```

-AllowDeprecated

Parameter Type:	<Switch>
Description:	STIGs that Evaluate-STIG supports but have been removed from https://www.cyber.mil/ are considered deprecated and will not be scanned by default. This parameter will enable the detection and scan of deprecated STIGs. ***If a STIG is forced with -ForceSTIG, deprecation will be ignored and the STIG will always be scanned.***
Example:	

```
.\Evaluate-STIG.ps1 -AllowDeprecated
```

-AllowSeverityOverride

Parameter Type:	<Switch>
Description:	Enables setting the Security Override and Justification fields in the checklist when the STIG contains verbiage that a specific criterion changes the severity of the check. ***Refer to your organization's policy if usage of the Security Override feature in STIG Viewer is allowed.***
Example:	

```
.\Evaluate-STIG.ps1 -AllowSeverityOverride
```

-AllowIntegrityViolations

Parameter Type:	<Switch>
Description:	Ignores Evaluate-STIG file integrity validation failures and allows the scan to continue.
Example:	

```
.\Evaluate-STIG.ps1 -AllowIntegrityViolations
```

-ApplyTattoo

Parameter Type:	<Switch>
Description:	Marks the asset with Evaluate-STIG tattooing. ***Note: If -SelectSTIG is used, this parameter is ignored.***
Example:	

```
.\Evaluate-STIG.ps1 -ApplyTattoo
```

-SMCollection

Parameter Type:	<String>
Description:	Used to direct Evaluate-STIG which STIG Manager collection settings to use within the Preferences.xml file. See STIG Manager for more.
Requires:	-Output STIGManager
Example:	

```
.\Evaluate-STIG.ps1 -Output STIGManager -SMCollection MyCollection
```

-SMPassphrase

Parameter Type:	<String>
Description:	If SMImport_CLIENT_CERT is encrypted, this provides the passphrase to decrypt the certificate's key defined in SMImport_CLIENT_CERT_KEY. ***The passphrase will be converted to a SecureString before used.***
Requires:	-Output and -SMCollection
Example:	

```
.\Evaluate-STIG.ps1 -Output STIGManager -SMCollection MyCollection -SMPassphrase MyPassphrase
```

-SplunkHECName

Parameter Type:	<String>
Description:	Use to direct Evaluate-STIG which configured Splunk HTTP Event Collection (HEC) to upload results. See Splunk for more.
Requires:	-Output Splunk
Example:	

```
.\Evaluate-STIG.ps1 -Output Splunk -SplunkHECName MyHECName
```

Remote and Cisco Options

The parameters below are used to initiate remote and Cisco config file scans. Most of the Scan Setting parameters discussed previously may be used in addition to these to customize remote scans:

-ComputerName

Parameter Type:	<String[]>
Description:	Use this parameter to perform a remote scan of Windows/Linux systems. Entries may be a computer name, IP address, text file of computers (one per line), a PowerShell array object, or a combination. Multiple may be specified using comma separation. ***Entries must be resolvable in DNS or the hosts file. This parameter is only valid for Windows hosts. See Remote Scanning for more.***
Example:	

```
.\Evaluate-STIG.ps1 -ComputerName Workstation01,"C:\ComputerList.txt"
```

-AltCredential

Parameter Type:	<Switch>
Description:	By default, for remote Windows scans, Evaluate-STIG will use the credential that launched PowerShell on the host computer to connect to the remote computer. Use this parameter to be prompted for an alternate credential to use for remote connections. ***This parameter is only for remote Windows connections, not Linux. See Remote Scanning for more.***
Requires:	-ComputerName
Example:	

```
.\Evaluate-STIG.ps1 -ComputerName Workstation01, "C:\ComputerList.txt" -AltCredential
```

-CiscoConfig

Parameter Type:	<String[]>
Description:	Path to Cisco `show tech-support` output file(s)/folder(s) to be scanned. Multiple entries may be specified using comma separation. ***If a folder is specified, the folder will be recursively searched for qualifying `show tech-support` files. See Cisco Scanning for more.***
Example:	

```
.\Evaluate-STIG.ps1 -CiscoConfig C:\ShowTech.txt,"C:\ShowTechFolder\"
```

-SelectDeviceType

Parameter Type:	<String[]>
Description:	For *Cisco IOS*, *IOS-XE Catalyst 9k* series, and *IOS-XE ISR* devices, specify if the device is acting as a router, switch or both using comma separation. All other IOS-XE devices will be automatically detected and this parameter ignored if set.
Valid Entries:	• **Router** • **Switch**
Requires:	-CiscoConfig
Example:	

```
.\Evaluate-STIG.ps1 -CiscoConfig C:\IOS_ShowTech.txt -SelectDeviceType Router,Switch
```

-ThrottleLimit

Parameter Type:	<Int16>
Description:	Maximum number of computers or configuration files to scan concurrently for remote and Cisco scans.
Requires:	-ComputerName or -CiscoConfig
Default:	"10"
Example:	

```
.\Evaluate-STIG.ps1 -ComputerName Workstation01,"C:\ComputerList.txt" -ThrottleLimit 15
```

Utility Options

Below are non-scan related parameters available in Evaluate-STIG:

-ListSupportedProducts

Parameter Type:	<Switch>
Description:	Displays all STIGs supported by Evaluate-STIG.
Example:	

```
.\Evaluate-STIG.ps1 -ListSupportedProducts
```

-ListApplicableProducts

Parameter Type:	<Switch>
Description:	Displays all STIGs supported by Evaluate-STIG that would be applicable to the asset. May be used in conjunction with -AllowDeprecated to include deprecated STIGs in the applicability check.
Example:	

```
.\Evaluate-STIG.ps1 -ListApplicableProducts [-AllowDeprecated]
```

-Version

Parameter Type:	<Switch>
Description:	Displays the version of Evaluate-STIG.
Example:	

```
.\Evaluate-STIG.ps1 -Version
```

-Update

Parameter Type:	<Switch>
Description:	Updates to the current version of Evaluate-STIG available on SPORK.
Requires:	Connection to the DODIN.
Example:	

```
.\Evaluate-STIG.ps1 -Update
```

-LocalSource

Parameter Type:	<String>
Description:	Updates from a path that contains the extracted Evaluate-STIG content.
Requires:	-Update
Example:	

```
.\Evaluate-STIG.ps1 -Update –LocalSource \\Server01\Evaluate-STIG\
```

-Proxy

Parameter Type:	<String>
Description:	Proxy to use when updating Evaluate-STIG. System proxy used by default.
Requires:	-Update
Example:	

```
.\Evaluate-STIG.ps1 -Update -Proxy 192.168.2.1:8080
```

3.2 Bash Wrapper Script

Evaluate-STIG provides a Bash wrapper script (Evaluate-STIG_Bash.sh) for Linux systems that do not have PowerShell installed. PowerShell is still used to perform the scan but it is a temporarily extracted instance of PowerShell that does not leave the Evaluate-STIG folder. **The PowerShell archive is not included with Evaluate-STIG and must be added by either using the --DownloadPS option or manually downloading.**

Parameters

--DownloadPS

Description:	Downloads the current version of PowerShell and saves it as powershell.tar.gz in the Evaluate-STIG folder.
Requires:	Internet access. Alternatively, the appropriate powershell-7.[x].[x]-linux.tar.gz manually downloaded, renamed to powershell.tar.gz, and placed in the Evaluate-STIG folder. • **RHEL 7 -** https://github.com/PowerShell/PowerShell/releases/tag/v7.3.0 • **All Others -** https://github.com/PowerShell/PowerShell/releases/latest
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --DownloadPS
```

--PSPath

Description:	Path to directory containing PowerShell executable (**pwsh**) if PowerShell is installed.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --PSPath /opt/microsoft/powershell/7/
```

--ScanType

Description:	Sets the -ScanType parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --ScanType Unclassified
```

--Marking

Description:	Sets the -Marking parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Marking MyMarking
```

--TargetComments

Description:	Sets the -TargetComments parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --TargetComments "My comments"
```

--VulnTimeout

Description:	Sets the -VulnTimeout parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --VulnTimeout 30
```

--FileSearchTimeout

Description:	Sets the -FileSearchTimeout parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --FileSearchTimeout 120
```

--AnswerKey

Description:	Sets the -AnswerKey parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --AnswerKey MyKey
```

--AFPath

Description:	Sets the -AFPath parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --AFPath //Server01/AnswerFiles/
```

--Output

Description:	Sets the -Output parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output CombinedCKL,CKLB
```

--JSON

Description:	Sets the -JSON parameter for Evaluate-STIG.ps1.
Requires	--Output Console
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output Console --JSON
```

--OutputPayload

Description:	Sets the -OutputPayload parameter for Evaluate-STIG.ps1.
Requires	--Output CSV|CombinedCSV|Splunk or --JSON
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output CSV --OutputPayload HostName,Title,GroupID,Status
```

--OutputPath

Description:	Sets the -OutputPath parameter for Evaluate-STIG.ps1.
Requires	--Output
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output CombinedCKL,CKLB --OutputPath //Server01/MyShare
```

--PreviousToKeep

Description:	Sets the -PreviousToKeep parameter for Evaluate-STIG.ps1.
Requires	--Output
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output CombinedCKL,CKLB –PreviousToKeep 5
```

--SelectSTIG

Description:	Sets the -SelectSTIG parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --SelectSTIG Firefox,Ubuntu20
```

--SelectVuln

Description:	Sets the -SelectVuln parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Requires:	--SelectSTIG
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --SelectSTIG Firefox,Ubuntu20 --SelectVuln V-251545,V-238196
```

--ExcludeVuln

Description:	Sets the -ExcludeVuln parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Requires:	--SelectSTIG
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --SelectSTIG Firefox,Ubuntu20 --ExcludeVuln V-251545,V-238196
```

--ExcludeSTIG

Description:	Sets the -ExcludeSTIG parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --ExcludeSTIG Firefox,Ubuntu20
```

--ForceSTIG

WARNING Evaluate-STIG results are not guaranteed with this option. Use at own risk.

Description:	Sets the -ForceSTIG parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --ForceSTIG Firefox,Ubuntu20
```

--AllowDeprecated

Description:	Sets the -AllowDeprecated parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --AllowDeprecated
```

--AllowSeverityOverride

Description:	Sets the -AllowSeverityOverride parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --AllowSeverityOverride
```

--AllowIntegrityViolations

Description:	Sets the -AllowIntegrityViolations parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --AllowIntegrityViolations
```

--ApplyTattoo

Description:	Sets the -ApplyTattoo parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --ApplyTattoo
```

--SMCollection

Description:	Sets the -SMCollection parameter for Evaluate-STIG.ps1.
Requires:	--Output STIGManager
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output STIGManager--SMCollection MyCollection
```

--SMPassphrase

Description:	Sets the -SMPassphrase parameter for Evaluate-STIG.ps1.
Requires:	--Output STIGManager and --SMCollection
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output STIGManager--SMCollection MyCollection --SMPassphrase MyPassphrase
```

--SplunkHECName

Description:	Sets the -SplunkHECName parameter for Evaluate-STIG.ps1.
Requires	--Output Splunk
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Output Splunk --SplunkHECName MyHECName
```

--CiscoConfig

Description:	Sets the -CiscoConfig parameter for Evaluate-STIG.ps1. Use comma separation for multiple.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --CiscoConfig /opt/ShowTech.txt,/opt/ShowTechFolder/
```

--SelectDeviceType

Description:	Sets the -SelectDeviceType parameter for Evaluate-STIG.ps1.
Requires:	--CiscoConfig
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --CiscoConfig /opt/IOS_ShowTech.txt --SelectDeviceType Router,Switch
```

--ThrottleLimit

Description:	Sets the -ThrottleLimit parameter for Evaluate-STIG.ps1.
Requires:	--CiscoConfig
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --CiscoConfig /opt/ShowTech.txt,/opt/ShowTechFolder/ --ThrottleLimit 15
```

--ListSupportedProducts

Description:	Sets the -ListSupportedProducts parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --ListSupportedProducts
```

--ListApplicableProducts

Description:	Sets the -ListApplicableProducts parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --ListApplicableProducts
```

--Version

Description:	Sets the -Version parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Version
```

--Update

Description:	Sets the -Update parameter for Evaluate-STIG.ps1.
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Update
```

--LocalSource

Description:	Sets the -LocalSource parameter for Evaluate-STIG.ps1.
Requires:	--Update
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Update --LocalSource //Server01/Evaluate-STIG
```

--Proxy

Description:	Sets the -Proxy parameter for Evaluate-STIG.ps1 or --DownloadPS parameter.
Requires:	--Update or --DownloadPS
Example:	

```
sudo bash Evaluate-STIG_Bash.sh --Update --Proxy 192.168.2.1:8080
```

3.3 Answer Files

Answer files are user-defined XML files to further automate *Not Reviewed* checks that cannot be evaluated technically or may contain verbiage in the STIG's `Check Text` preventing Evaluate-STIG from reaching a definitive status. Answer Files are also useful for providing justification or mitigation to known *Open* checks. Answer Files will place user-defined text into the `Comments` field of the checklist, can change the resultant `Status` of the check, and may include PowerShell code to add logic for ensuring certain criteria is met before the answer is applied. Answer files may be stored in the `.\Evaluate-STIG\AnswerFiles` folder or an alternate location when using the `-AFPath` option.

⇨ ***Note:*** *The `Evaluate-STIG_GUI.ps1` provides a graphical interface for creating, editing, and maintaining your answer files. Unless you are very comfortable working with raw XML, it is recommended to use this to ensure a properly formatted answer file.*

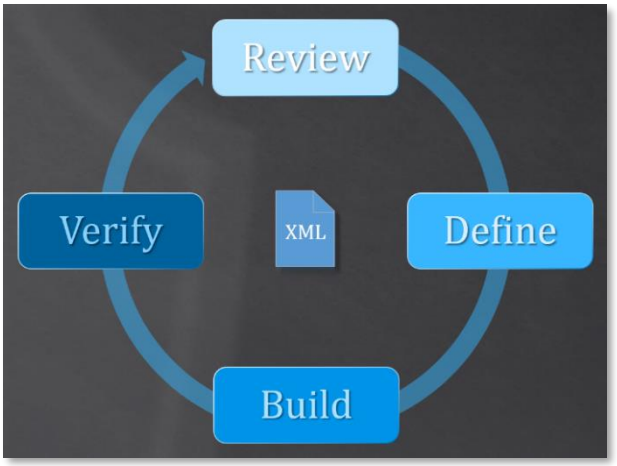
Answer files are per-STIG. Ideally, you will have a single answer file for each STIG that contains all checks needing answers for that STIG. An answer file may contain multiple vulnerability IDs, multiple answer keys per vulnerability ID, and multiple Indexes per answer key.

Evaluate-STIG will automatically select the answer file to use from `[AFPath]` by matching the Evaluate-STIG's `STIG ShortName` with the `<STIGComments Name>` element within the answer file. **If multiple answer files for the STIG exist, Evaluate-STIG will utilize the most recently modified file.**

⇨ ***Note:*** *Answer files are not intended to hide or explain away non-compliant checks that do not have an approved reason to be in a non-compliant state. Answer file development should start AFTER a good baseline of asset configuration has been established and an Evaluate-STIG scan of that configuration has been completed.*

Development Cycle

Answer files are an advanced feature of Evaluate-STIG. They can be written from scratch using any text editor or use the Evaluate-STIG GUI included with Evaluate-STIG to draft answer files in a GUI interface. Collaboration between SMEs and cybersecurity authorities should be fostered to certify the quality and trust of the answers being provided. The following development cycle is recommended when developing answer file content:



1. **Review:** SME and cyber authority should review scan results for checks that could benefit from an answer file.

2. **Define:** SME and cyber authority collaborate on verbiage (`Comment`) and any criteria that may need developed (`ValidationCode`).

3. **Build:** SME writes / updates answer file based on defined requirements.

4. **Verify:** SME executes a re-scan using the answer file to confirm expectation is met.

Structure

Answer files are XML documents and must follow proper structure. Evaluate-STIG will validate every answer file against a schema to ensure proper formatting. Any that fail validation will be recorded in the Evaluate-STIG.log and console. Below is the basic structure of an answer file:

```
<STIGComments Name="_replace_with_stig_shortname_">
  <Vuln ID="V-00000">
    <AnswerKey Name="_name_that_makes_sense_for_your_needs_or_'DEFAULT'">
      <Answer Index="1" ExpectedStatus="NR" Hostname="" Instance="" Database="" Site="" ResultHash="">
        <ValidationCode></ValidationCode>
        <ValidTrueStatus></ValidTrueStatus>
        <ValidTrueComment></ValidTrueComment>
        <ValidFalseStatus></ValidFalseStatus>
        <ValidFalseComment></ValidFalseComment>
      </Answer>
    </AnswerKey>
  </Vuln>
</STIGComments>
```

<STIGComments>

Description:	STIG the answer file targets. Evaluate-STIG automatically associates this answer file with the STIG identified here.
Parent Element:	None – top level
Occurrence:	This element may only be specified once within the answer file.
Attributes:	• **Name:** STIG ShortName as displayed in -ListSupportedProducts or -SelectSTIG.
Example:	

```
<STIGComments Name="MSEdge">
```

<Vuln>

Description:	Group ID or Rule ID the answer is for.
Parent Element:	<STIGComments>
Occurrence:	This element may be specified multiple times within the <STIGComments> section but each occurrence must be unique. If both the Group ID and Rule ID for the same check are configured, first listed in the answer file wins.
Attributes:	• **ID:** Group ID or Rule ID from STIG. ***Note:*** *Using Rule ID may require more frequent updates to the answer file. If the Rule ID in the STIG changes due to a STIG update, the ID will need to be updated in the answer file before it will be applicable.*
Example:	

```
<Vuln ID="V-235753"> or <Vuln ID=" SV-235753r1015297_rule">
```

<AnswerKey>

Description:	Key name that is called with -AnswerKey parameter.
Parent Element:	<Vuln>
Occurrence:	This element may be specified multiple times within each <Vuln> section but each occurrence within <Vuln> must be unique.
Attributes:	• **Name:** May be "DEFAULT" or any other text. See Selection of Winning Answer for how Evaluate-STIG selects the key when multiple AnswerKey sections are configured within a <Vuln>.
Example:	

```
<AnswerKey Name="MyNetworkName"> or <AnswerKey Name="DEFAULT">
```

<Answer>

Description:	Criteria for determining answer applicability. Weights are calculated for each answer to determine the winning answer. Attributes that are configured but not valid in the context of the STIG are not included in weighting. See Identifying Optional Attributes for more on the attributes noted as "optional".
Parent Element:	<AnswerKey>
Occurrence:	This element may only be specified multiple times within an <AnswerKey> section provided the Index attribute is unique.
Attributes:	• **Index:** Unique identifier. May be any value but must be unique to other indexes within the same <AnswerKey>. • **ExpectedStatus:** The Status that Evaluate-STIG determined the check to be without modification. If this does not match the Status that Evaluate-STIG determined, the answer is ignored. Must be a valid status entry. • **Hostname:** [***Optional***] Hostname(s) of systems this answer applies to. Use comma separation to specify multiple. • **Instance:** [***Optional***] Instance(s) this answer applies to. Use comma separation to specify multiple. • **Database:** [***Optional***] Database(s) this answer applies to. Use comma separation to specify multiple. • **Site:** [***Optional***] Site(s) this answer applies to. Use comma separation to specify multiple. • **ResultHash:** [***Optional***] ResultHashes(s) this answer applies to. Use comma separation to specify multiple.
Example:	

```
<Answer Index="1" ExpectedStatus="NR" Hostname="PC1" Instance="" Database="" Site="" ResultHash="">
```

<ValidationCode>

Description:	[***Optional***] PowerShell code to be executed before applying answer. This is for providing logic to checks that must meet additional criteria. Status and Comment determined by the value returned. Returned value may be **$true**/**$false** or a Hashtable object. If object, it MUST contain both **Valid** (as Boolean) and **Results** keys. Both Results and the answer file comment will be written to the Comments field of the STIG check. If this element is left empty, then **$true** is assumed. See Validation Code section for more.
Parent Element:	<AnswerKey>
Occurrence:	This element may only be specified once within each <AnswerKey> section.
Expected Value:	PowerShell that returns **$true**\|**$false**, a hashtable, or left empty.
Example:	

```
<ValidationCode>Test-Path $env:windir</ValidationCode>
```

<ValidTrueStatus>

Description:	Status of check if <ValidationCode> code is empty, returns **$true**, or Valid in hashtable is **$true**. If empty scanned Status will remain unchanged.
Parent Element:	<AnswerKey>
Occurrence:	This element may only be specified once within each <AnswerKey> section.
Expected Value:	May be left empty or a valid status value used.
Example:	

```
<ValidTrueStatus>NotAFinding</ValidTrueStatus>
```

<ValidTrueComment>

Description:	Text to be put into Comments if <ValidationCode> is empty or returns **$true**, or Valid in hashtable is **$true**.
Parent Element:	<AnswerKey>
Occurrence:	This element may only be specified once within each <AnswerKey> section.
Expected Value:	Any text is acceptable.
Example:	

```
<ValidTrueComment>My comment for when ValidationCode is empty or returns $true.</ValidTrueComment>
```

<ValidFalseStatus>

Description:	Status of check if <ValidationCode> is not **$true**, or Valid in hashtable is **$false**. If empty scanned Status will remain unchanged.
Parent Element:	<AnswerKey>
Occurrence:	This element may only be specified once within each <AnswerKey> section.
Expected Value:	May be left empty or a valid status value used.
Example:	

```
<ValidFalseStatus>Open</ValidFalseStatus>
```

<ValidFalseComment>

Description:	Text to be put into Comments if <ValidationCode> is not **$true**, or Valid in hashtable is **$false**.
Parent Element:	<AnswerKey>
Occurrence:	This element may only be specified once within each <AnswerKey> section.
Expected Value:	Any text is acceptable.
Example:	

```
<ValidFalseComment>My comment for when ValidationCode returns a value not $true.</ValidFalseComment>
```

Status Formats

The following formats are valid entries for ExpectedStatus, ValidTrueStatus, and ValidFalseStatus. They may be mixed formats if desired. Evaluate-STIG will automatically cross-reference formats when processing status entries:

EvalSTIG	CKL	CKLB	XCCDF
NR	Not_Reviewed	Not_Reviewed	notchecked
NF	NotAFinding	not_a_finding	pass
O	Open	Open	fail
NA	Not_Applicable	Not_Applicable	notapplicable

Validation Code

The <ValidationCode> element is an optional configuration for providing logic within your answer files to determine which status and comment should be applied to the result. The code must return either a Boolean value or a hashtable containing Boolean **Valid** and string **Results** values. If the output is $true, then the ValidTrueStatus and ValidTrueComment are applied. Otherwise, the ValidFalseStatus and ValidFalseComment are applied. If <ValidationCode> is not configured, then $true is assumed and ValidTrueStatus and ValidTrueComment are always applied. Examples:

Returning a simple Boolean value:

```
<ValidationCode>
$Fruits = @("Apples", "Oranges", "Peaches")
if ("Bananas" -in $Fruits) {
    return $true
}
else {
    return $false
}
</VAlidationCode>
```

Returning a hashtable value:

```
<ValidationCode>
$ValidationObject = @{
    Valid   = $false
    Results = ""
}

$Fruits = @("Apples", "Oranges", "Peaches")
if ("Oranges" -in $Fruits) {
    $ValidationObject.Valid = $true
    $ValidationObject.Results = "Bananas are in the list of fruits."
}
else {
    $ValidationObject.Valid = $false
    $ValidationObject.Results = "Bananas are NOT in the list of fruits."
}

return $ValidationObject
</VAlidationCode>
```

Exposed Variables

The following scan data will be exposed to answer files that may be called from validation code:

Attribute	Description
$ESPath	Path that Evaluate-STIG.ps1 was executed from.
$ExpectedStatus	Status that Evaluate-STIG determined. Will be in CKL format (refer to Status Formats).
$ResultHash	SHA1 hash of ResultData. Reference FindingDetails for calculated ResultHash.
$ResultData	FindingDetails content after "~~~~~" bar.
$Username	Username processed for HKCU check. Reference to FindingDetails for appropriate Username value.
$UserSID	User SID processed for HKCU check. Reference to FindingDetails for appropriate UserSID value.
$Instance	Instance name processed. Reference FindingDetails for appropriate Instance value.
$Database	Database name processed. Reference FindingDetails for appropriate Database value.
$Site	Site name processed. Reference FindingDetails for appropriate Site value.

Example leveraging these variables in your validation code:

```
if ($Site -eq "Default Web Site") { # If the $Site being scanned is "Default Web Site", process.
    # Initialize hashtable object
    $ValidationObject = @{
        Valid   = $false
        Results = ""
    }

    if ($ResultHash -eq "B6AA552093C875CFBF6E0F8C4BAA73C928D14474") {
        # The $ResultHash matches what is expected for a ValidTrueStatus|ValidTrucComment
        $ValidationObject.Valid = $true
        $ValidationObject.Results = "ResultHash for $($Site) equals the expected hash value."
    }
    else {
        # The $ResultHash differs from expected so ValidFalseStatus|ValidFalseComment is applied.
        $ValidationObject.Valid = $false
        $ValidationObject.Results = "ResultHash for $($Site) is not the expected hash value "
    }

    return $ValidationObject
}
```

Selection of Winning Answer

Evaluate-STIG uses the following process and weighting for determining which answer should be applied when the answer file has a configuration for the GroupID or RuleID currently being scanned. Details on the selected answer and the calculated weight for the vulnerability ID can be found in the `Evaluate-STIG.log`.

1. AnswerKey Selection

a. Answer Key that equals value of `-AnswerKey` parameter on command line OR
b. Answer Key with name of "DEFAULT".

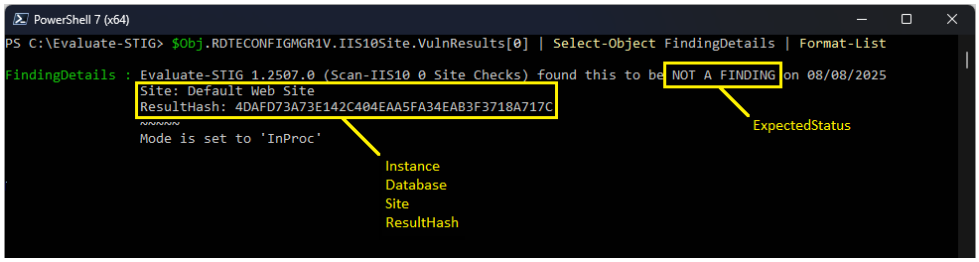
2. Index Selection

All Indexes are weighted based on the configured attributes that are applicable (see Identifying Optional Attributes). The highest is the selected answer. If multiple Indexes are tied with the highest weight, the first winning Index listed in the answer file will be selected. The following attributes, if configured in the Index AND within context of the STIG, are summed to calculate the weight:

Attribute	Weight	Description
`ExpectedStatus`	0	ExpectedStatus is a hard requirement. If not a match, the Index is ignored. Any format may be used provided it equates to the status returned by Evaluate-STIG.
`Hostname`	5	Name of the machine. Comma separate for multiple. If configured, the machine name must match or Index is ignored.
`Instance`	4	Instance (e.g. SQL). Comma separate for multiple. If configured, the Instance must match or Index is ignored.
`Database`	3	Database (e.g. SQL). Comma separate for multiple. If configured, the Database must match or Index is ignored.
`Site`	2	Site (e.g. IIS). Comma separate for multiple. If configured, the Site must match or Index is ignored.
`ResultHash`	1	Hash of Finding Details text below the "~~~~~" line. If configured, the ResultHash must match or Index is ignored.
`<AnswerKey Name>`	16	An `<AnswerKey Name>` match on any value not "DEFAULT" will have a starting weight of 16.

Identifying Optional Attributes

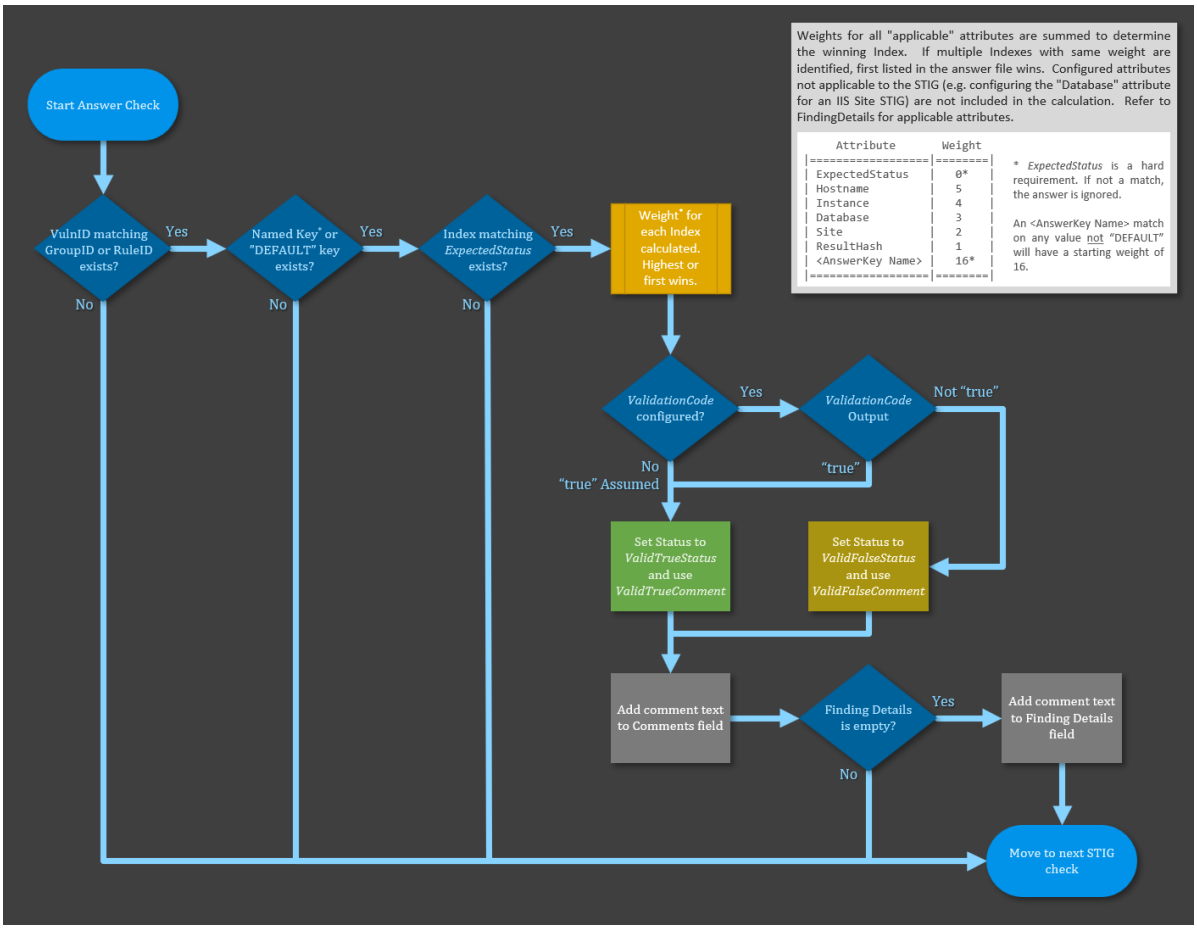
Apart from `Hostname`, which is simply the name of the computer being scanned, values for attributes that are in the context of the STIG can be found in the `Finding Details` section of the output:



Configuring an attribute in the Index will require that the attribute's value from the scan match the Index configuration and that attribute will be added to the weight. If multiple values are configured in the Index with comma separation, each will be checked for a match with the scan's value.

⇨ ***Note:*** *Only attributes that are within the context of the STIG will be displayed in Finding Details. Attributes configured in the Index but not listed in Finding Details are ignored (matching is not required) and not added to the weight.*

Answer File Flowchart



Sample Answer File

Below is an example answer file for the Microsoft Edge STIG. It will address up to two vulnerability IDs (V-235751 and V-235753).

```
<STIGComments Name="MSEdge">
  <Vuln ID="V-235751">
    <!--Edge development tools must be disabled.-->
    <AnswerKey Name="MyCustomKey">
      <Answer Index="1" ExpectedStatus="O" Hostname="" Instance="" Database="" Site="" ResultHash="">
      <ValidationCode></ValidationCode>
      <ValidTrueStatus>O</ValidTrueStatus>
      <ValidTrueComment>RDTE is a development environment so developer tools are required.</ValidTrueComment>
      <ValidFalseStatus></ValidFalseStatus>
      <ValidFalseComment></ValidFalseComment>
    </AnswerKey>
    <AnswerKey Name="DEFAULT">
      <Answer Index="1" ExpectedStatus="O" Hostname="" Instance="" Database="" Site="" ResultHash="">
      <ValidationCode></ValidationCode>
      <ValidTrueStatus>O</ValidTrueStatus>
      <ValidTrueComment>Non RDTE systems must not have developer tools enabled.  This needs fixed.</ValidTrueComment>
      <ValidFalseStatus></ValidFalseStatus>
      <ValidFalseComment></ValidFalseComment>
    </AnswerKey>
  </Vuln>
  <Vuln ID="V-235753">
    <!--URLs must be whitelisted for plugin use if used.-->
    <AnswerKey Name="MyCustomKey">
      <Answer Index="1" ExpectedStatus="NR" Hostname="MyPC1,MyPC2" Instance="" Database="" Site="" ResultHash="">
      <ValidationCode>
$ValidationResults = @{
    Valid = $true
    Results = ""
}

$Key = Get-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Edge\PopupsAllowedForUrls"
$Key.GetValueNames() | ForEach-Object {$($Key.GetValue($_))} | ForEach-Object {
    If ($_ -notin @('[*.]navy.mil','[*.]apps.mil')) {
        $ValidationResults.Valid = $false
        $ValidationResults.Results += "Unapproved URL: $_`n"
    }
}
If ($ValidationResults.Valid -eq $true) {
    $ValidationResults.Results = "Identified URLs are approved."
}

Return $ValidationResults
      </ValidationCode>
      <ValidTrueStatus>NF</ValidTrueStatus>
      <ValidTrueComment>Identified URLs are approved for whitelisting.</ValidTrueComment>
      <ValidFalseStatus>O</ValidFalseStatus>
      <ValidFalseComment>An identified URL is not approved for whitelisting.</ValidFalseComment>
    </AnswerKey>
  </Vuln>
</STIGComments>
```

In this example, the <Answer> will be applied for each <Vuln> depending on the command executed:

```
.\Evaluate-STIG.ps1
```

- **V-235751**:
  If the Status returned by Evaluate-STIG is Open (O), the DEFAULT Index will be applied since -AnswerKey was not specified in the command.
- **V-235753**:
  Will not be processed since -AnswerKey was not specified in the command and there is no "DEFAULT" key configured for this check.

```
.\Evaluate-STIG.ps1 -AnswerKey MyCustomKey
```

- **V-235751**:
  If the Status returned by Evaluate-STIG is Open (O), the RDTE Index will be applied since -AnswerKey MyCustomKey specified in the command.
- **V-235753**:
  If the Status returned by Evaluate-STIG is Not Reviewed (NR) AND the computer name is either “MyPC1” or “MyPC2”, then MyCustomKey Index 1 is applied since -AnswerKey MyCustomKey is specified. Final Status and Comments determined by <ValidationCode> output.

Answer File Maintenance and Migration

Maintenance:
Answer files must be maintained to ensure they stay appropriate with STIG updates. This is especially true for answer files that use the Rule ID from the STIG. Using Rule ID in your answer file instead of Group ID helps ensure answer files don't become stale and reduces the risk of inappropriate answers being applied. But this also means that your answer files will require more frequent updating. Any change to a STIG check results in a change to its Rule ID. Rule IDs configured in an answer file that no longer match the Rule ID in the current STIG are no longer applicable answers and are automatically ignored.

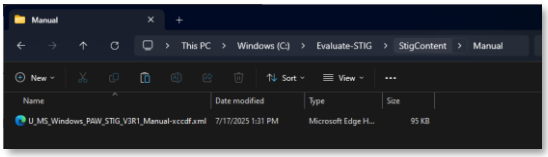
Migration:
Answer files that were developed for earlier versions of Evaluate-STIG must be migrated to the new format. Legacy answer files will fail schema validation and will not be used in scans. Be sure to migrate any older answer files to ensure they still work in your environment.

Both of the above needs can be addressed using either the `Maintain` button in the Evaluate-STIG GUI's Answer File Tools Tab or with the `Maintain-AnswerFiles.ps1` script located in the `Auxiliary` folder from the Evaluate-STIG distributable .zip.

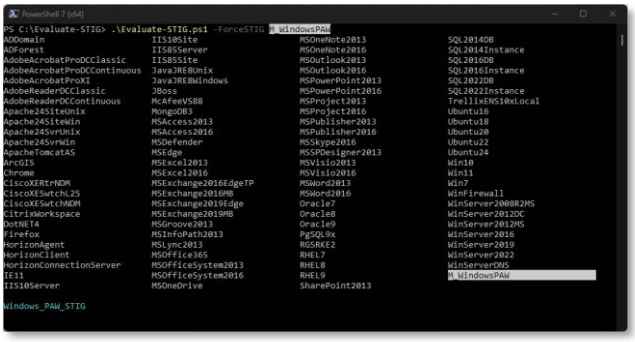
3.4 Unsupported STIGs

It is possible for Evaluate-STIG to create checklists for STIGs that are not natively supported using an answer file. To add an unsupported STIG:

1. Extract the `xccdf.xml` file from the unsupported STIG's .zip and place it in the `<Evaluate-STIG>\StigContent\Manual` folder.



2. Add `-ForceSTIG <ShortName>` to your command line. Unsupported STIG shortnames will always be prefixed with "M_". Pressing `[CTRL]+[SPACE]` after `ForceSTIG` will display the list of all possible shortnames to choose from.



3. Optionally, include the unsupported STIG in `-SelectSTIG` if you want to limit the scan to just specific STIG(s).

4. Create an answer file for the checks you wish to have automated in the unsupported STIG.

3.5 Preferences.xml

The `Preferences.xml` file may be used to preset some Evaluate-STIG, STIG Manager, and Manage-AnswerFile settings.

If a parameter is both configured in `Preferences.xml` and on the command line, the command line parameter will be used.

⇨ ***Note:*** *A backup of the default preferences file is located under ".\Evaluate-STIG\xml\Preferences.default.xml"*

EvaluateSTIG Section

The following scan settings may be configured:

- <ScanType>
- <Marking>
- <TargetComments>
- <VulnTimeout>
- <FileSearchTimeout>
- <AnswerKey>
- <AFPath>
- <Output>
- <JSON>
- <OutputPath>
- <PreviousToKeep>
- <AllowDeprecated>
- <AllowSeverityOverride>
- <ExcludeSTIG>
- <ExcludeVuln>
- <ApplyTattoo>
- <SMCollection>
- <SplunkHECName>

```
<EvaluateSTIG>
  <ScanType>Unclassified</ScanType>
  <Marking></Marking>
  <TargetComments></TargetComments>
  <VulnTimeout>15</VulnTimeout>
  <FileSearchTimeout>240</FileSearchTimeout>
  <AnswerKey>DEFAULT</AnswerKey>
  <AFPath></AFPath>
  <Output>Console</Output>
  <JSON>false</JSON>
  <OutputPath></OutputPath>
  <PreviousToKeep>1</PreviousToKeep>
  <AllowDeprecated>false</AllowDeprecated>
  <AllowSeverityOverride>false</AllowSeverityOverride>
  <ExcludeSTIG></ExcludeSTIG>
  <ExcludeVuln></ExcludeVuln>
  <ApplyTattoo>false</ApplyTattoo>
  <SMCollection></SMCollection>
  <SplunkHECName></SplunkHECName>
</EvaluateSTIG>
```

OutputPayload Section

The `<OutputPayload>` section configures the default fields to be outputted when using `-Output CSV|CombinedCSV|Splunk` or `-JSON` parameters. All are Boolean and must be lowercase “true” or “false”. If `-OutputPayload` is used, the settings here are ignored in favor of the fields specified on the command line.

```
<OutputPayload>
  <Title>true</Title>
  <Version>true</Version>
  <ReleaseDate>true</ReleaseDate>
  <Classification>true</Classification>
  <HostName>true</HostName>
  <Site>true</Site>
  <Instance>true</Instance>
  <IP>true</IP>
  <MAC>true</MAC>
  <FQDN>true</FQDN>
  <Role>true</Role>
  <GroupID>true</GroupID>
  <GroupTitle>true</GroupTitle>
  <RuleID>true</RuleID>
  <STIGID>true</STIGID>
  <Severity>true</Severity>
  <SeverityOverride>true</SeverityOverride>
  <Justification>true</Justification>

* Image truncated for display purposes *
```

STIGManager Section

The <STIGManager> section within **Preferences.xml** must be configured to your environment in order for Evaluate-STIG to send scan results to a STIG Manager instance:

```
<STIGManager>
  <SMImport_API_BASE></SMImport_API_BASE>
  <SMImport_AUTHORITY></SMImport_AUTHORITY>
  <SMImport_COLLECTION Name="">
    <SMImport_CLIENT_ID></SMImport_CLIENT_ID>
    <SMImport_CLIENT_CERT></SMImport_CLIENT_CERT>
    <SMImport_CLIENT_CERT_KEY></SMImport_CLIENT_CERT_KEY>
    <SMImport_COLLECTION_ID></SMImport_COLLECTION_ID>
  </SMImport_COLLECTION>
</STIGManager>
```

<SMImport_API_BASE>	**Required**. Base URL of the STIG Manager API service. The default value is your STIGManager instance's FQDN with "/api" appended. This is defined within your STIG Manager's settings via "STIGMAN_CLIENT_API_BASE".
<SMImport_AUTHORITY>	**Required**. Base URL of the OIDC authentication service that issues OAuth2 tokens for the API. This should match the value set for "STIGMAN_CLIENT_OIDC_PROVIDER" within STIG Manager's configuration.
<SMImport_COLLECTION Name>	**Required**. Name for the SMImport collection settings section that is called from "-SMCollection". Recommend this match your collection name within STIG Manager. Multiple SMImport_COLLECTION sections may be configured.
<SMImport_CLIENT_ID>	**Required**. OIDC client ID to authenticate. This should be created within your STIG Manager's backend OIDC Provider. The default provider used by STIG Manager is Keycloak, though your configuration may vary.
<SMImport_CLIENT_CERT>	The PEM encoded client certificate. An unencrypted private key may be included within this file so that you do not have to pass "-SMPassphrase", though this configuration is not recommended. File must exist in Certificates directory.
<SMImport_CLIENT_CERT_KEY>	Filename of PEM encoded encrypted private key. Required if SM_Import_CLIENT_CERT does not contain a plaintext private key. File must exist in Certificates directory.
<SMImport_COLLECTION_ID>	**Required**. The collection ID of your desired collection. A user with Manage permissions on the collection can find this. After selecting to manage the collection, reference the "ID" value in the Collection Properties window.

⇨ ***Note:*** *Defining* **SMImport_CLIENT_CERT_KEY** *will require the use of the "-SMPassphrase" parameter to decrypt the private key.*

Splunk Section

The <Splunk> section within **Preferences.xml** must be configured to your environment in order for Evaluate-STIG to send scan results to a Splunk instance:

```
<Splunk>
  <Splunk_URI></Splunk_URI>
  <Splunk_HECName Name="">
    <Splunk_token></Splunk_token>
    <Splunk_index></Splunk_index>
    <Splunk_source></Splunk_source>
    <Splunk_sourcetype></Splunk_sourcetype>
  </Splunk_HECName>
</Splunk>
```

<Splunk_URI>	**Required**. Base URL of the Splunk instance. The default value is your Splunk instances' FQDN, port 8088, with "/services/collector/event" appended.
<Splunk_HECName Name>	**Required**. Name for the HTTP Event Collector token. Multiple Splunk_HECName sections may be configured.
<Splunk_token>	**Required**. HEC Token value. This should be created within your Splunk instance.
<Splunk_index>	**Optional.** The name of the index by which the event data is to be indexed. The index you specify here must be within the list of allowed indexes if the token has the indexes parameter set.
<Splunk_source>	**Optional**. The source value to assign to the event data.
<Splunk_sourcetype>	**Optional**. The source type value to assign to the event data.

ManageAnswerFiles Section

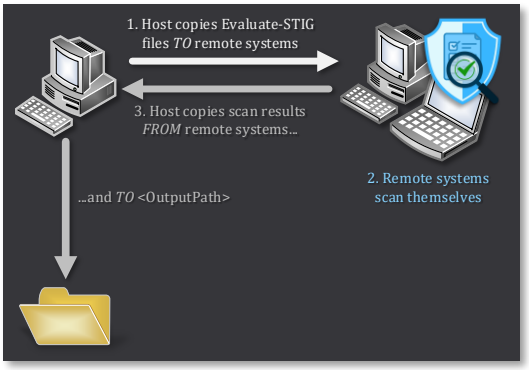
<ManageAnswerFiles> section is for configuring default settings in the Options portion of the Evaluate-STIG GUI Answer Files Tools tab

```
<ManageAnswerFiles>
  <EvaluateSTIG_Results></EvaluateSTIG_Results>
  <AnswerFileDirectory></AnswerFileDirectory>
  <DefaultAFKey>DEFAULT</DefaultAFKey>
  <StatusFormat>EvalSTIG</StatusFormat>
  <BackupOnSave>DEFAULT</BackupOnSave>
  <PowerShell_IDE>C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell_ISE.exe</PowerShell_IDE>
</ManageAnswerFiles>
```

<EvaluateSTIG_Results>	Path to checklist files.
<AnswerFileDirectory>	Path to answer files.
<DefaultAFKey>	Answer file key name to use when adding vulnerability ID to answer file.
<StatusFormat>	Format use for Status entries within the answer file.
<BackupOnSave>	To create a backup of the answer file when saving changes. Must be "true" or "false".
<PowerShell_IDE>	Path to preferred PowerShell editor.

3.6 Remote Scanning

Evaluate-STIG can scan remote systems provided specific requirements are met. For managed environments, remote scanning is not a replacement for, nor recommended over pushing Evaluate-STIG configuration management tools as those are better suited for deploying to entire networks of computers.



Use the `-ComputerName` parameter to initiate a remote scan. Evaluate-STIG content and answer files are compressed on the Host machine, copied to the Remote(s), and extracted to a temporary folder. Then, the remote machine scans itself and the results are returned to the Host machine as either a PowerShell object or the output file type(s) if `-Output` is used. The Host is responsible for sending the results to the `[OutputPath]`.

⇨ ***Note:*** *`-OutputPath` is from the Host's perspective. If not specified, `[OutputPath]` will be on the Host – not the remote:*

- ***Windows:*** *`"C:\Users\Public\Documents\STIG_Compliance"`*
- ***Linux:*** *`"/opt/STIG_Compliance"`*

The following table outlines supported remote scan scenarios:

Host Operating System	Remote Operating System	Remote Scan Supported
Windows	Windows	✓
Windows	Linux	✓*
Linux	Linux	✓+
Linux	Windows	✗

* Windows to Linux supported when PowerShell 7.3.x or greater is installed on both the host and remote.
\+ Linux to Linux supported using optional Ansible script, which is available in our Auxiliary Files at the download locations.

Requirements

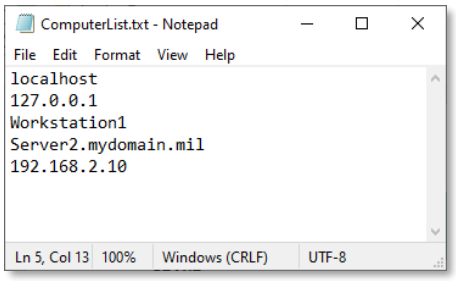
All remote scans are performed within an established PowerShell session between the host and remote. Therefore, requirements for allowing PowerShell session connections must be met before Evaluate-STIG can execute a remote scan. These requirements are out-of-scope for this guide but below are official Microsoft articles for ensuring that sessions can be created:

- **Windows to Windows Connection:**
  - https://learn.microsoft.com/en-us/powershell/scripting/learn/remoting/wsman-remoting-in-powershell
  - https://learn.microsoft.com/en-us/powershell/scripting/learn/remoting/winrmsecurity
  - **Additional Configuration for Non-Domain (Workgroup) Systems:**
    - https://woshub.com/using-psremoting-winrm-non-domain-workgroup/
- **Windows to Linux Connection:**
  - https://learn.microsoft.com/en-us/powershell/scripting/learn/remoting/ssh-remoting-in-powershell

For Windows to Windows scans, the Evaluate-STIG prerequisites apply to both the host and the remote. For Windows to Linux, PowerShell 7.3.x or greater must be installed on both the host and remote. Additionally, Linux prerequisites must be in place on the Linux remotes.

⇨ ***Note:*** *Remote systems must be resolvable in DNS or the HOSTS file.*

Parameter Notes



- `-ComputerName` accepts names, IP addresses, text files, PowerShell array objects or any combination. Multiple may be specified using comma separation. May also use "`localhost`" to include the host machine as part of the scan. If a text file, the content must contain one computer per line.
- `-AltCredential` will prompt for an alternate credential to use when establishing connection to remote Windows systems. Evaluate-STIG will first attempt to connect using the alternate credential. If it fails, connection will be attempted using the credential that launched PowerShell on the host computer before giving up.
- `-ThrottleLimit` will set the maximum number of concurrent connections allowed. Default is 10. If requested scan contains more computers than `[ThrottleLimit]`, Evaluate-STIG will start the maximum scans, wait for one to finish, then start the next until all are completed.

3.7 Cisco Scanning

Evaluate-STIG supports Cisco STIGs by parsing captured "`show tech-support`" output saved to a file. It does not make any connection to devices. To initiate a Cisco scan, use the `-CiscoConfig` parameter.

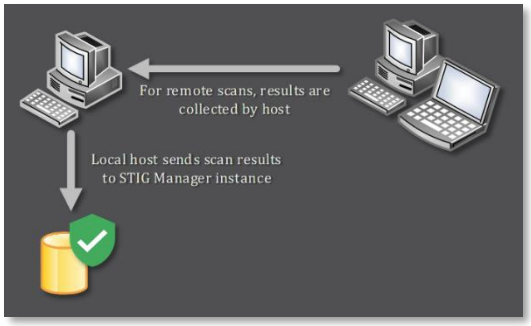
Parameter Notes

- `-CiscoConfig` must be the path to a file or a folder. If a folder, Evaluate-STIG will recursively search the folder for valid configuration files and add to scan. Multiple may be specified using comma separation. ***Important:*** *Only files that contain the full "show tech-support" output from a supported device will be accepted by Evaluate-STIG. Using output from "show running-config" will not suffice.*
- `-SelectDeviceType` will instruct Evaluate-STIG if the device is acting as a router, switch, or both for the following platforms:
  - Cisco IOS
  - Cisco IOS-XE Catalyst 9k series
  - Cisco IOS-XE ISR
- `-ThrottleLimit` will set the maximum number of concurrent files to parse. Default is 10. If requested scan contains more files than `[ThrottleLimit]`, Evaluate-STIG will start the maximum scans, wait for one to finish, then start the next until all are completed.

⇨ ***Note:*** *Refer to Appendix D for a list of Cisco STIGs that Evaluate-STIG supports.*

3.8 STIG Manager

Evaluate-STIG provides the ability to send scan results directly to a functioning STIG Manager instance. "STIG Manager is an Open-Source API and Web client for managing the assessment of Information Systems for compliance with security checklists published by the United States (U.S.) Defense Information Systems Agency (DISA). STIG Manager supports DISA checklists distributed as either a Security Technical Implementation Guide (STIG) or a Security Requirements Guide (SRG) in the XCCDF format."



When Evaluate-STIG is run with `-Output STIGManager`, the local machine will send the scan results to a STIG Manager instance and collection defined in `Preferences.xml` via STIG Manager's API. For remote scans, the remote scan results will be transferred back to the host machine and the host will send to STIG Manager.

Prerequisites

- A functioning STIG Manager instance and service account for connecting to STIG Manager. Refer to STIG Manager's documentation.
- Configured `<SMImport_API_BASE>` and `<SMImport_AUTHORITY>` within `Preferences.xml`.
- At least one configured `<SMImport_COLLECTION>` section.

Parameter Notes

- `-Output STIGManager` instructs Evaluate-STIG to send results to a STIG Manager instance.
- `-SMCollection` is required to direct Evaluate-STIG which `<SMImport_COLLECTION>` section to use within `Preferences.xml`.
- `-SMPassphrase` is required if the `<SMImport_CLIENT_CERT>` is used with a passphrase OR `<SMImport_CLIENT_ID>` is used with Client_Secret (SMPassphrase)

Usage

Consider the following example STIG Manager configuration in `Preferences.xml`:

```
<STIGManager>
  <SMImport_API_BASE>https://my.stig.manager.mil/api</SMImport_API_BASE>
  <SMImport_AUTHORITY>https://my.stig.manager.mil/kc/realms/stigman</SMImport_AUTHORITY>
  <SMImport_COLLECTION Name="MyCollection">
    <SMImport_CLIENT_ID>evaluate-stig</SMImport_CLIENT_ID>
    <SMImport_CLIENT_CERT>evaluate-stig-crt.pem</SMImport_CLIENT_CERT>
    <SMImport_CLIENT_CERT_KEY>evaluate-stig-key.pem</SMImport_CLIENT_CERT_KEY>
    <SMImport_COLLECTION_ID>1</SMImport_COLLECTION>
  </SMImport_COLLECTION>
</STIGManager>
```

The command line below will send the results to collection ID '**1**' within the STIG Manager instance. Since an encrypted key is configured in `<SMImport_CLIENT_CERT_KEY>`, the `-SMPassphrase` will be converted to a `SecureString` and used to decrypt the '**`evaluate-stig-key.pem`**' file required to authenticate to the STIG Manager instance.

```
.\Evaluate-STIG.ps1 -Output STIGManager -SMCollection MyCollection -SMPassphrase <.pem passphrase>
```

3.9 Splunk

Evaluate-STIG provides the ability to send scan results directly to a functioning Splunk instance. "Splunk Enterprise is a software product that enables you to search, analyze, and visualize the data gathered from the components of your IT infrastructure or business. Splunk Enterprise takes in data from websites, applications, sensors, devices, and so on. After you define the data source, Splunk Enterprise indexes the data stream and parses it into a series of individual events that you can view and search."

When Evaluate-STIG is run with `-Output Splunk`, the local machine will send the scan results to a Splunk HTTP Event Collector (HEC) defined in `Preferences.xml`. For remote scans, the remote scan results will be transferred back to the host machine and the host will send to Splunk.

Prerequisites

- A functioning Splunk HTTP Event Collector and token for authenticating to your Splunk instance. Refer to Splunk's documentation.
- Configured `<Splunk_URI>` within `Preferences.xml`.
- At least one configured `<Splunk_HECName>` section.

Parameter Notes

- `-Output Splunk` instructs Evaluate-STIG to send results to a Splunk HTTP Event Collector.
- `-SplunkHECName` is required to direct Evaluate-STIG which `<Splunk_HECName>` section to use within `Preferences.xml`.

Usage

Consider the following example Splunk configuration in `Preferences.xml`:

```
<Splunk>
  <Splunk_URI>https://my.splunk.mil:8088/services/collector/event</Splunk_URI>
  <Splunk_HECName Name="MyHECName">
    <Splunk_token>1cea9f66-a7c9-4fcd-a20f-0b2c17d5d426</Splunk_token>
    <Splunk_index>es_events</Splunk_index>
    <Splunk_source>evaluate-stig</Splunk_source>
    <Splunk_sourcetype>weekly_scan</Splunk_sourcetype>
  </Splunk_HECName>
</Splunk>
```

This HTTP Event Collector (HEC) name would be used when running:

```
.\Evaluate-STIG.ps1 -Output Splunk -SplunkHECName MyHECName
```

3.10 Updating Evaluate-STIG

Evaluate-STIG is updated at least quarterly in line with the posted DoD Cyber Exchange quarterly STIG release schedule at https://www.cyber.mil/stigs/quarterly-release-schedule-and-summary. An updated version of Evaluate-STIG is typically released within 2 weeks after the STIG compilation for the quarter is posted to https://www.cyber.mil/stigs/compilations. It is not uncommon for additional Evaluate-STIG releases during the quarter to address bug fixes, out-of-band STIG releases, or new features.

There are two options for updating Evaluate-STIG:

1. If connected to the DODIN, use the `-Update` feature. This will pull down the current Evaluate-STIG content while preserving your `Preferences.xml` settings and any answer files located in the `.\Evaluate-STIG\AnswerFiles` path.
2. Download the latest version from IntelShare:
    - **NIPR:** https://intelshare.intelink.gov/sites/NAVSEA-RMF
    - **SIPR:** https://intelshare.intelink.sgov.gov/sites/NAVSEA-RMF

⇨ ***Note:*** *The above are the only supported mechanisms for updating Evaluate-STIG. Adding a newer STIG's .xccdf content to Evaluate-STIG is not supported as the code has not been updated for any changes to that STIG and could affect accuracy of the results. STIG content that fails an internal hash check will result in an error and that STIG ignored.*

4 Evaluate-STIG GUI

`Evaluate-STIG.ps1` included with Evaluate-STIG is a graphical interface to build and run scans. Run it from a PowerShell prompt:

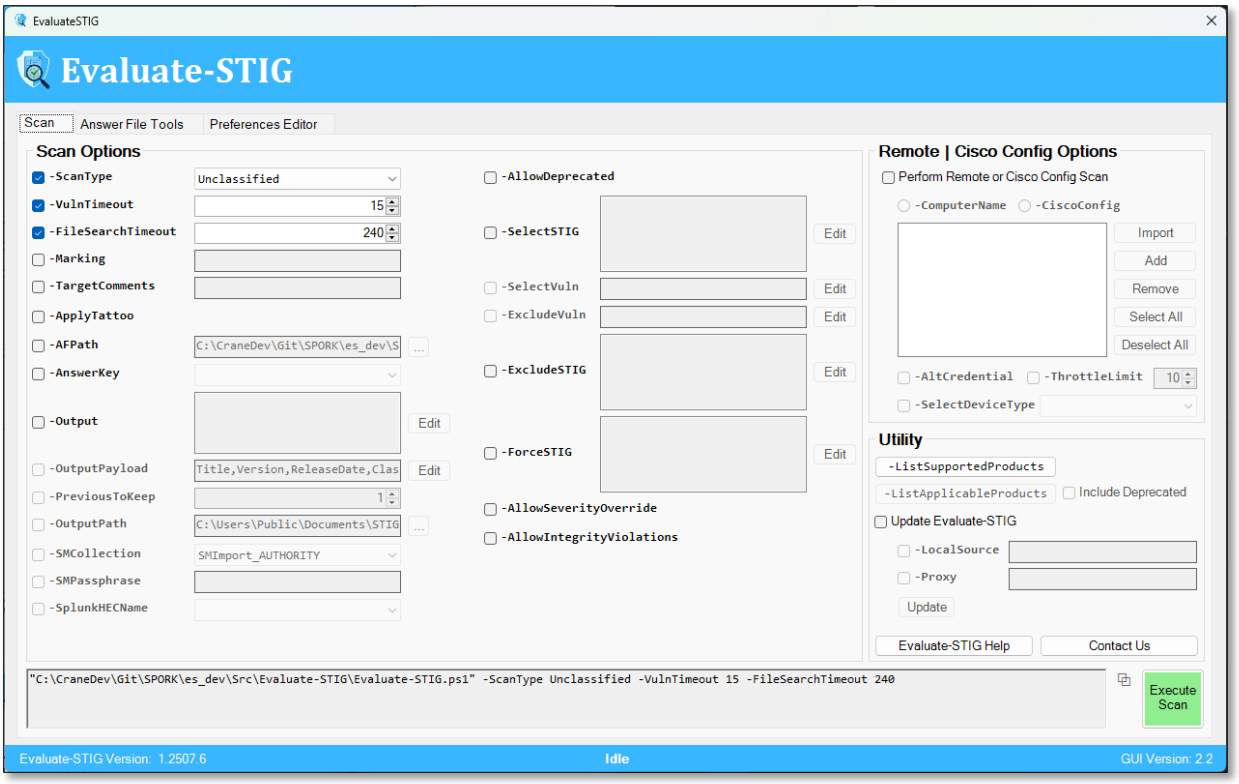
```
PS C:\Evaluate-STIG> .\Evaluate-STIG_GUI.ps1
```

⇨ ***Note:*** *If the local system is to be part of the scan, it must be run from an elevated PowerShell prompt on Windows.*

Requirements

- Windows operating system
- 1920 x 1080 screen resolution
- PowerShell 5.1 or greater

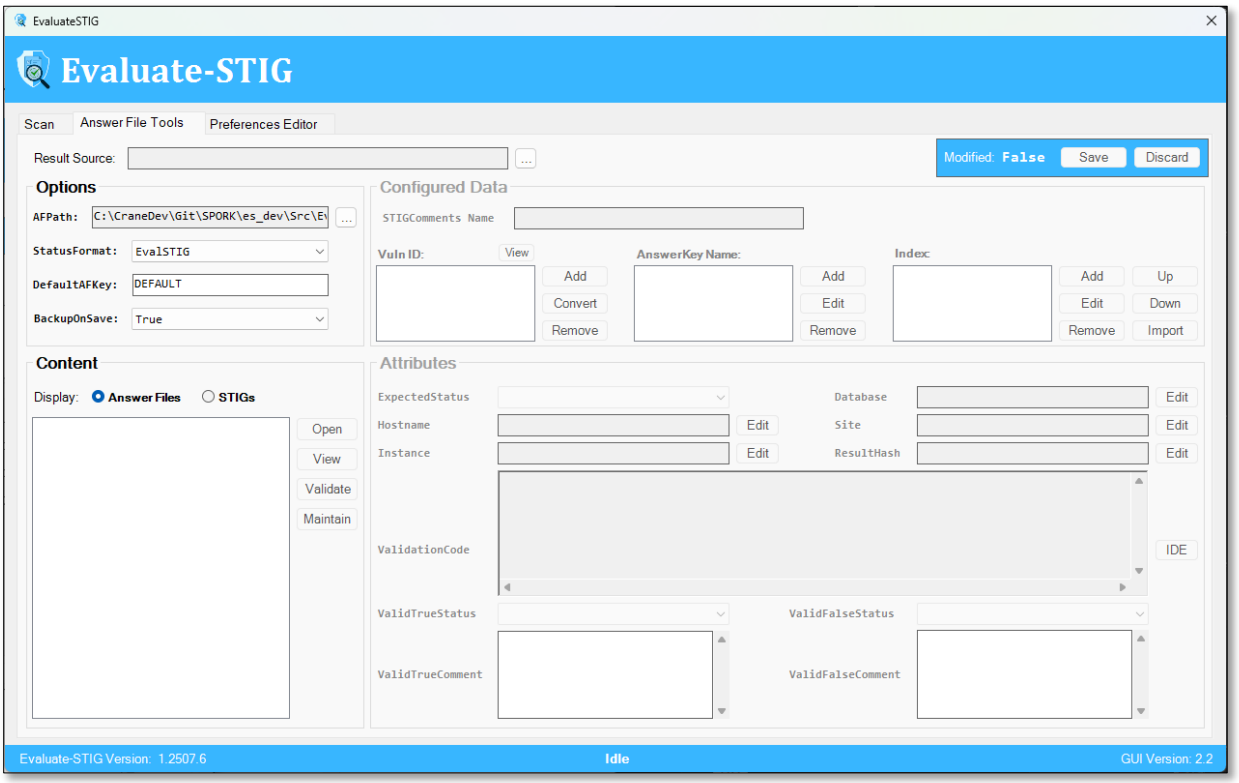
4.1 Scan Tab



The Scan tab is for performing a local or remote scan as well as running utility parameters. For scanning, the command line that will be executed is displayed at the bottom of the form. To enable a parameter, select the checkbox and then configure the parameter's setting if needed.

⇨ ***Note:*** *If a parameter is greyed out (disabled), this means that the parameter's prerequisite has not been satisfied.*

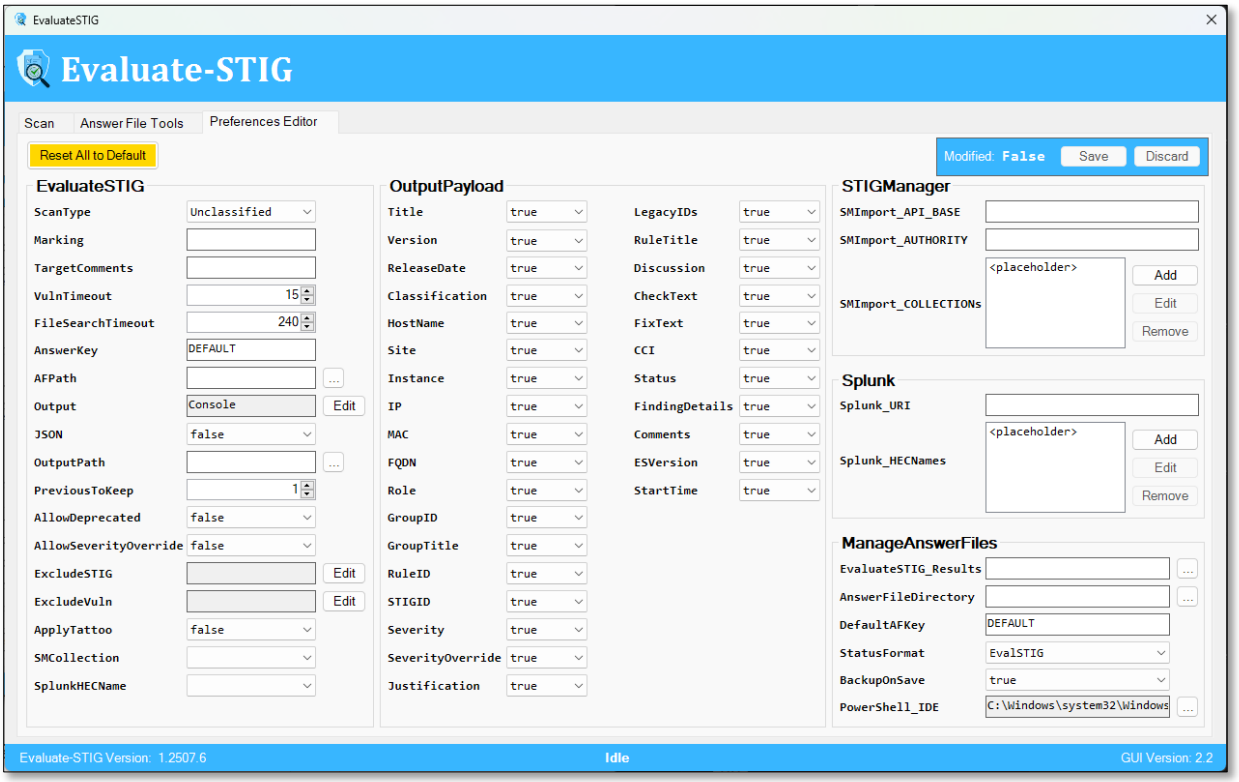
4.2 Answer File Tools Tab



Use the `Answer File Tools` tab to create, edit, and maintain your answer files. For routine maintenance or migration of legacy answer files, use the `Maintain` button.

⇨ ***Note:*** *Once a change to the answer file being edited is detected. The `Save|Discard` buttons will be enabled. Opening another answer file for editing will be disabled until the current changes are saved or discarded.*

4.3 Preferences Editor tab



The Preferences Editor tab is used for modifying the Evaluate-STIG Preferences.xml file. Any changes will require a restart of the GUI application.

5 Scan Processes

At the beginning, Evaluate-STIG will build a list of STIGs to scan by checking the applicability of each against the computer's configuration. Using `-SelectSTIG` does not guarantee applicability and Evaluate-STIG will still only scan for selected STIGs if they are required for the computer. The exception is `-ForceSTIG` in which Evaluate-STIG will attempt to scan the STIG regardless. Failed and/or inaccurate scans are possible when forcing STIGs so these results are not guaranteed and the user accepts the risk.

Evaluate-STIG utilizes a temporary folder to store data required for the scan. After scan completion, these files will be removed leaving just the `Evaluate-STIG.log` for reference or troubleshooting purposes. Location of the temporary folder:

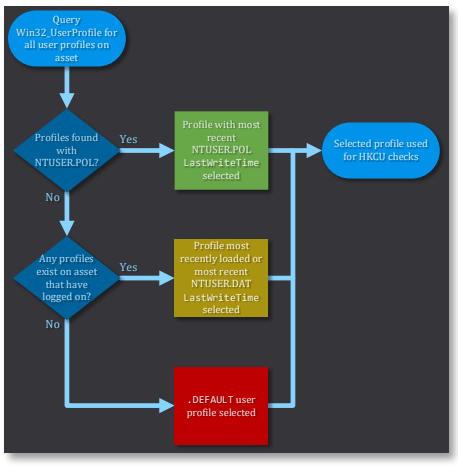
- **Windows:** $env:windir\Temp\Evaluate-STIG
- **Linux:** /tmp/Evaluate-STIG

Additionally, if a STIG requires the scan of Windows user settings, the preferred user's registry hive will be temporarily imported into the registry as a key under `HKEY_USERS\Evaluate-STIG_UserHive`. After scan completion, the key will be removed.

When the `-Output` parameter contains CKL, CKLB, CSV, `CombinedCKL`, `CombinedCKLB`, `CombinedCSV`, XCCDF, `Summary`, and/or OQE, the `-PreviousToKeep` parameter will dictate how many previous scan outputs to retain. Default is "1" (***retain one set of previous scan results***). Retained scan results will be moved into `[OutputPath]\Previous\[Results Date-Time] folder`.

5.1 Preferred User Selection Process

Nearly all user-based STIG settings instruct the reviewer to check the HKEY_CURRENT_USER (HKCU) registry hive on Windows systems. The STIG's `Fix Text` for user-based settings normally are group policy (GPO) configurations. Scanning all user profiles on a system can result in mass false positives due to how Windows applies user policy settings. It is highly recommended not to include Evaluate-STIG as part of your imaging process or run it on systems where no users have logged on.



To stay true to the STIG and simulate an administrator / auditor in-person session, Evaluate-STIG will logically select a "preferred user profile" for scanning user-based checks in the following order:

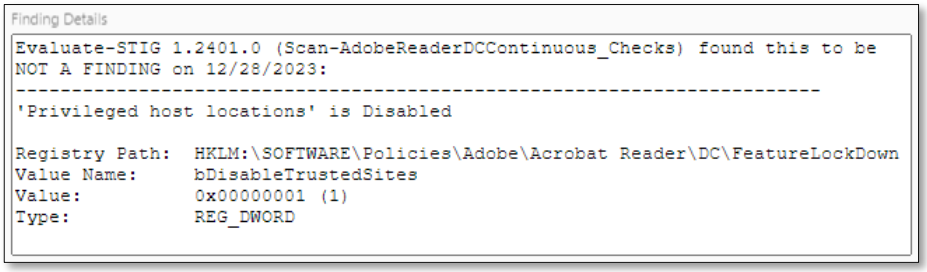
1. Profile that most recently applied GPO. This is the ideal scenario.
2. If no profiles have processed GPO, then the user that most recently logged on. Risk of mass false positives exists.
3. If no user has ever logged on, then the ".DEFAULT" user profile is used. This will most likely result in mass false positives.

⇨ ***Note:*** *For checks where the STIG states that each user on the system must be examined, Evaluate-STIG will examine every profile in these rare instances.*

5.2 CKL | CKLB Documentation

Evaluate-STIG has built-in support for formatting scan results into STIG Viewer 2.17 (`.CKL`) and STIG Viewer 3.x (`.CKLB`) checklist files. Use the `-Output` parameter to specify which output(s) to send the results to.

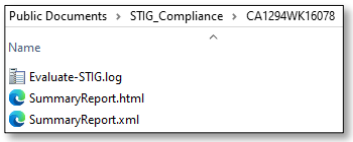
Evaluate-STIG will set the **`Status`** and document the computer's configuration for the check into **`Finding Details`**. If an answer file is used and applied, the text from the answer file will be placed into the **`Comments`** field of the checklist.



⇨ ***Note:*** *Evaluate-STIG reserves the Finding Details field for documenting the actual configuration as found. It is not possible for an answer file to update the Finding Details field. The only exception to this is for checks that Evaluate-STIG provided no data to Finding Details, at which point, the answer file Comment text will be duplicated to Finding Details. See Answer File Flowchart for more.*

5.3 Summary Reports

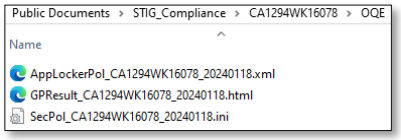
When specifying `Summary` in `-Output`, Evaluate-STIG will produce a summary of the scan in both .XML and .HTML formats.



- `SummaryReport.xml` – Useful for feeding scan summary data to external tools (e.g., SPLUNK)
- `SummaryReport.html` – Human readable report

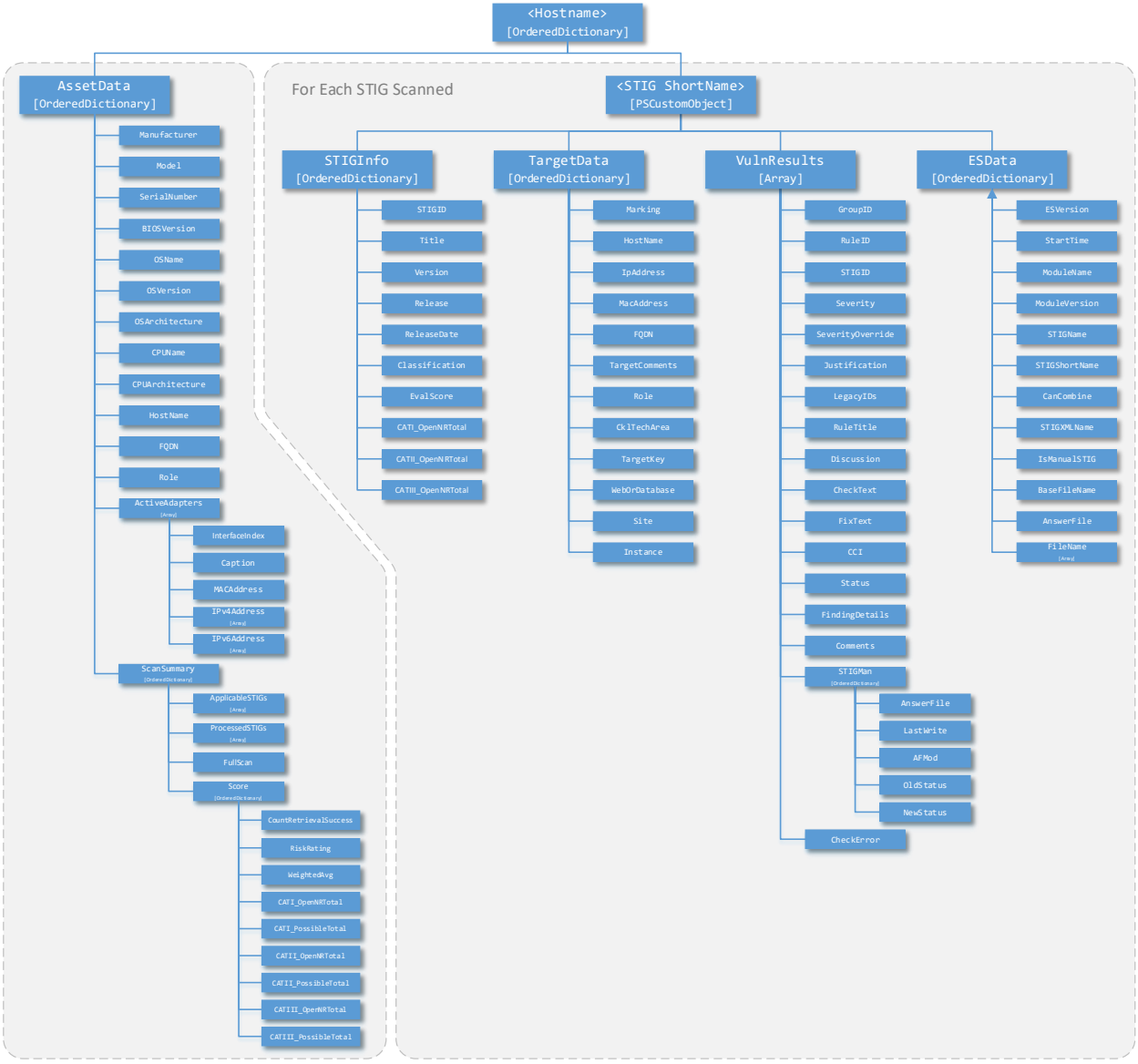
5.4 Objective Quality Evidence (OQE)

Scanning Windows systems and specifying **OQE** in `-Output`, Evaluate-STIG will generate additional artifacts as part of the scan and place these into `[OutputPath]\OQE`. These will include **AppLocker**, **Group Policy**, and **Local Security Policy** outputs.



6 Scan Results

Evaluate-STIG stores the scan result as a PowerShell object and will return this object to the console by default. Directing the console output to a variable (e.g. `$Obj = .\Evaluate-STIG.ps1`) allows for flexibility in filtering, sorting, and formatting the output to meet the user's need. **When using `-Output`, the results will only be sent to the console if "`Console`" is specified in the parameter.** Below is the structure of the PowerShell object:

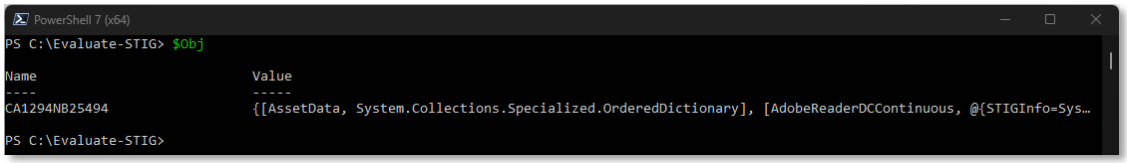


6.1 Walking the Object

The following are examples of walking the object within the console:

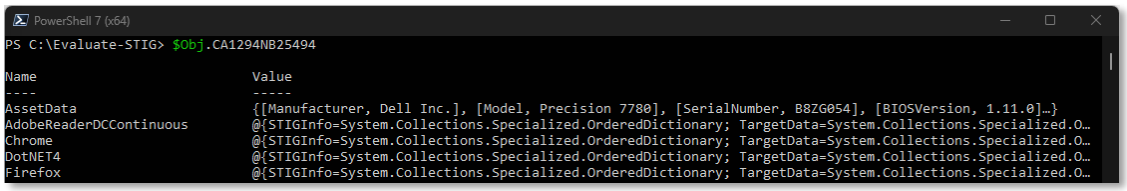
Parsing the top level (if remote or Cisco scan with multiple devices, there will be multiple entries here – one for each hostname):

```
PS C:\Evaluate-STIG> $Obj = .\Evaluate-STIG.ps1
```



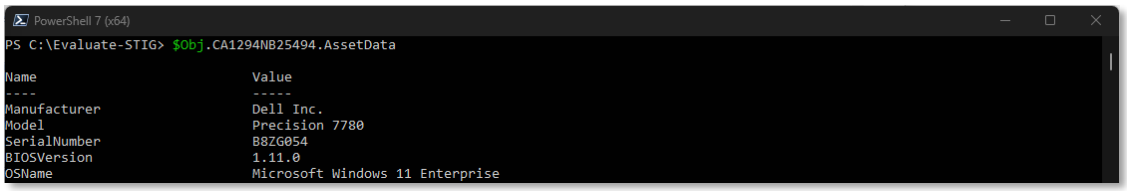
Examining the **<Hostname>** object (all STIGs scanned for the host will be present):

```
PS C:\Evaluate-STIG> $Obj.<Hostname>
```



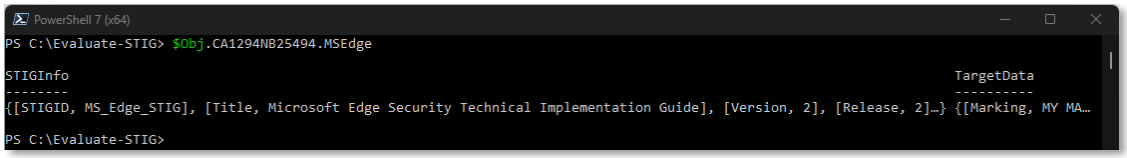
Examining the **AssetData** object:

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.AssetData
```



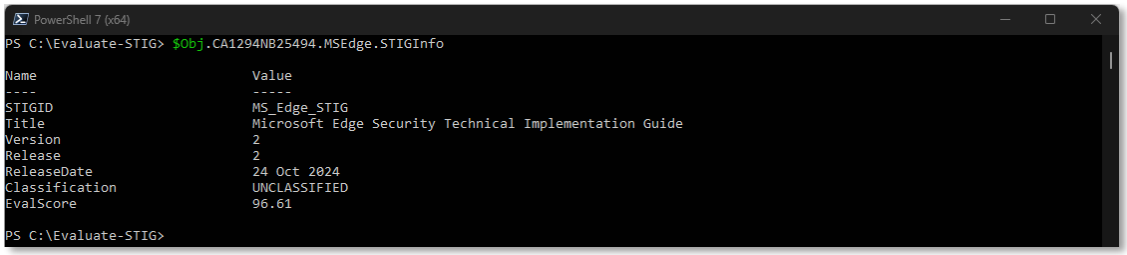
Examining the **<STIG ShortName>** object:

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.<STIGShortName>
```



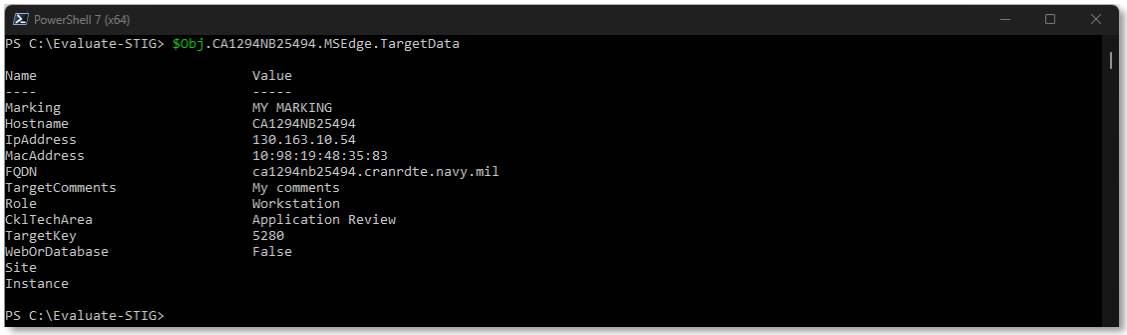
Examining the **STIGInfo** object (data from the STIG's `.xccdf`):

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.<STIGShortName>.STIGInfo
```



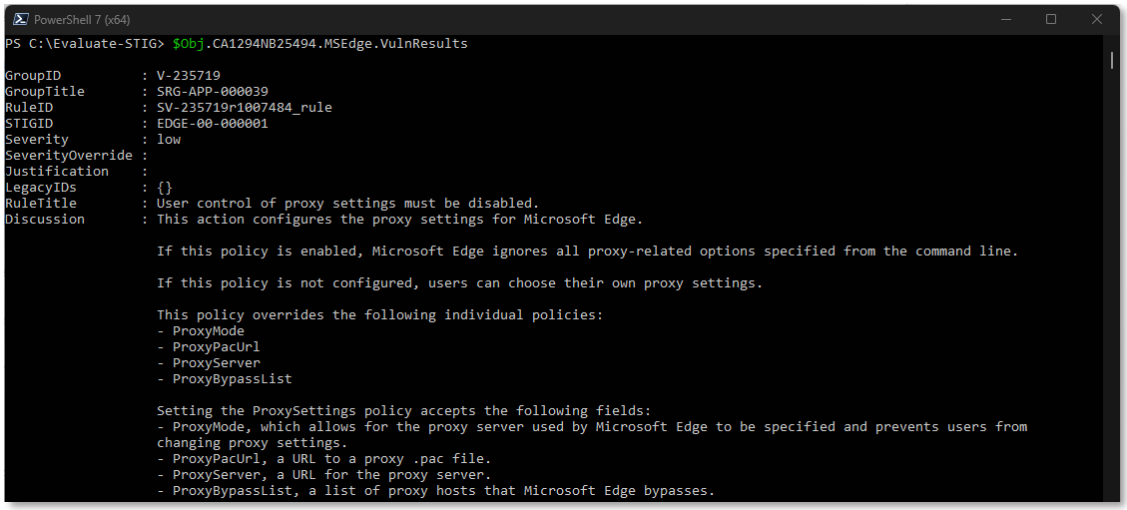
Examining the **TargetData** object:

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.<STIGShortName>.TargetData
```



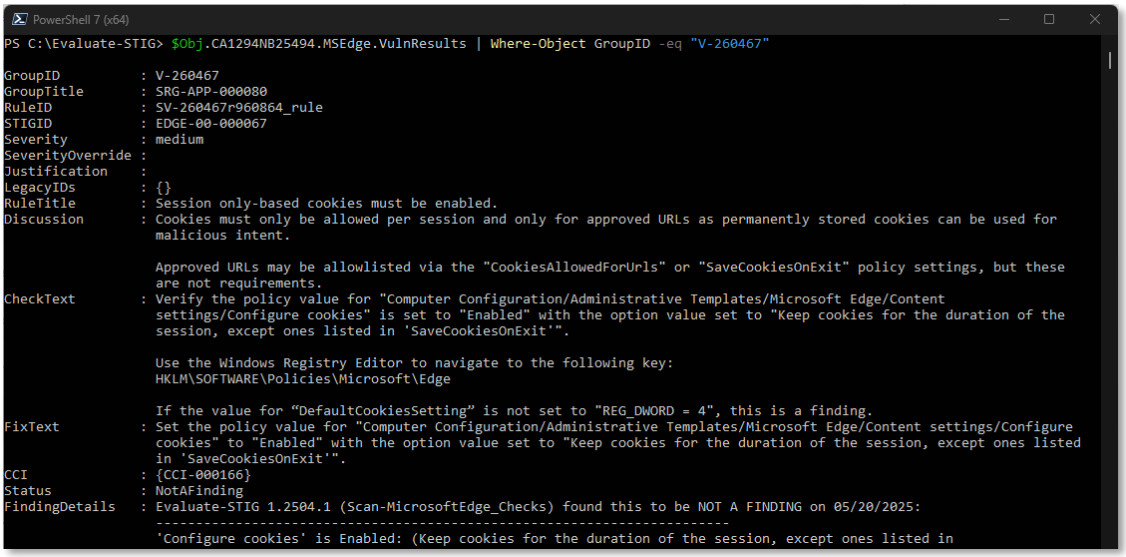
Examining the **VulnResults** object:

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.<STIGShortName>.VulnResults
```



Examining a specific check in the **VulnResults** object:

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.<STIGShortName>.VulnResults | Where-Object GroupID -eq "<VulnID>"
```



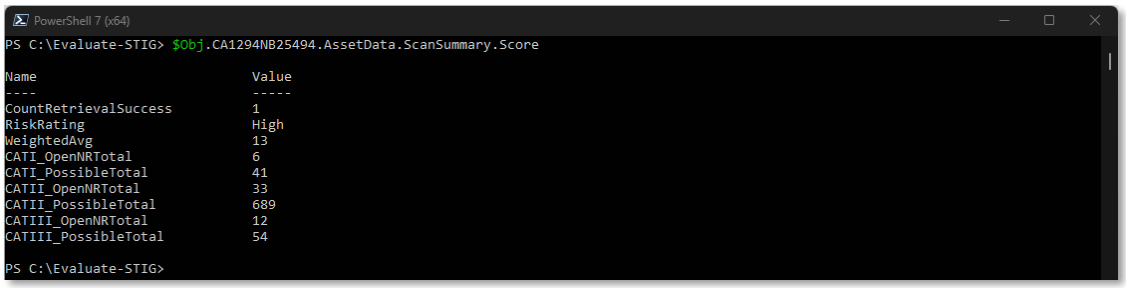
6.2 Scoring

Evaluate-STIG automatically calculates scores based on the results of appliable STIGs.

CORA Grading

All STIGs Evaluate-STIG deems as applicable to the asset are considered in these calculations. STIGs that may apply multiple times to an asset (e.g. multiple IIS web sites) will have CAT counts multiplied appropriately. **Deprecated STIGs are not considered applicable**.

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.AssetData.ScanSummary.Score
```



`RiskRating` is derived from the CORA Technology Areas matrix based on **`WeightedAvg`** (*Threshold*) percentage:

Technology Areas

Concern Indicator	Threshold
Very High Risk	≥ 20%
High Risk	≥ 10% and < 20%
Moderate Risk	> 0% and < 10%
Low Risk	0 CAT Is, <5% II & III
Very Low Risk	0.0%

`WeightedAvg` percentage is calculated from the CORA Scoring Methodology (note that *Not Reviewed* are considered *Open*):

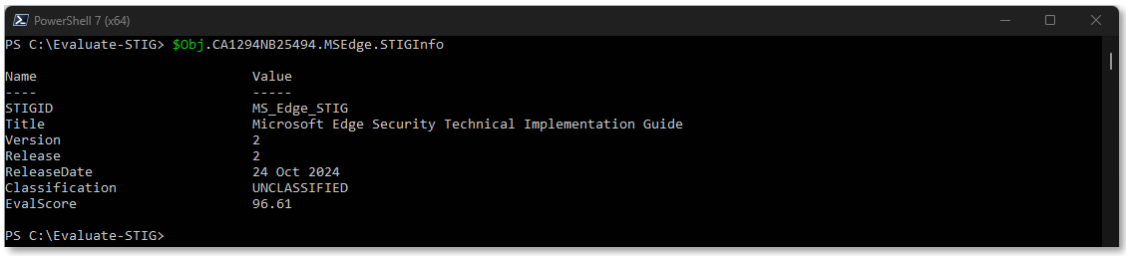
STIG Scan / Manual STIG:

$$Weighted\ Average = (p_1w_1 + p_2w_2 + p_3w_3) / (w_1 + w_2 + w_3)$$

$p_n = percent\ open$ $w_n = weight\ (10/4/1)$

Per STIG Score

```
PS C:\Evaluate-STIG> $Obj.<Hostname>.<STIGShortName>.STIGInfo
```



`EvalScore` is the percentage of compliant checks (*Not A Finding* and *Not Applicable*) for that STIG.

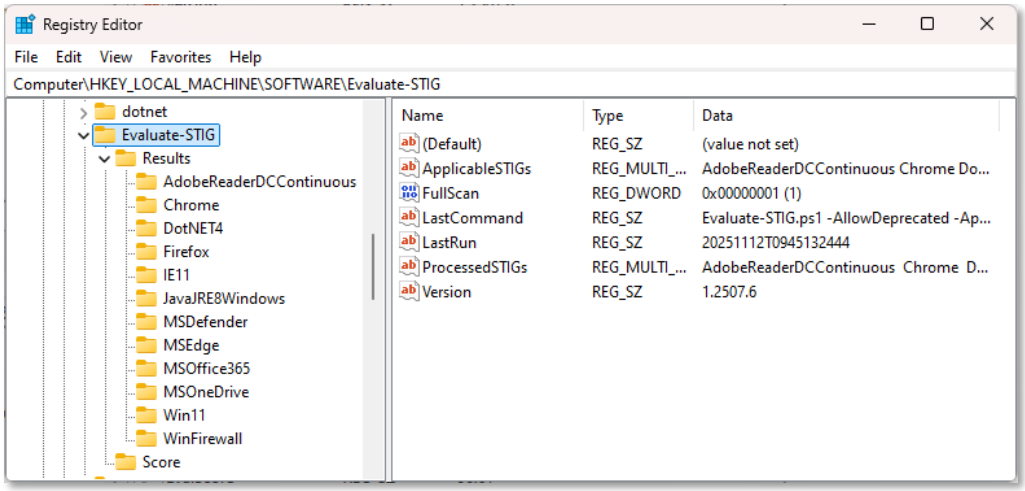
6.3 Tattooing

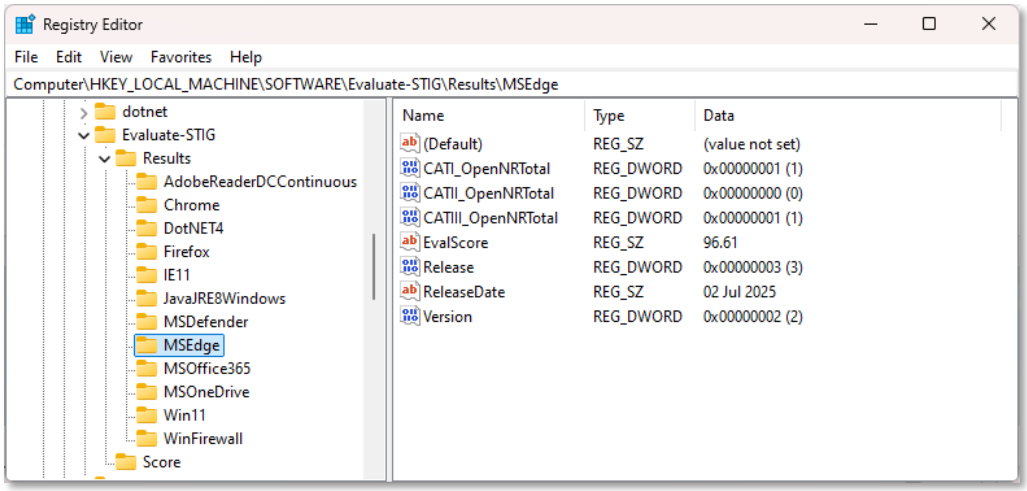
Evaluate-STIG scans can optionally tattoo the asset with data points from the scan using the -ApplyTattoo parameter. This information can be retrieved by external processes for inventory, verification, automated decision making, etc. For Windows systems, the tattoo is placed in the registry and Linux as a file. The tattoo consists of the following data:

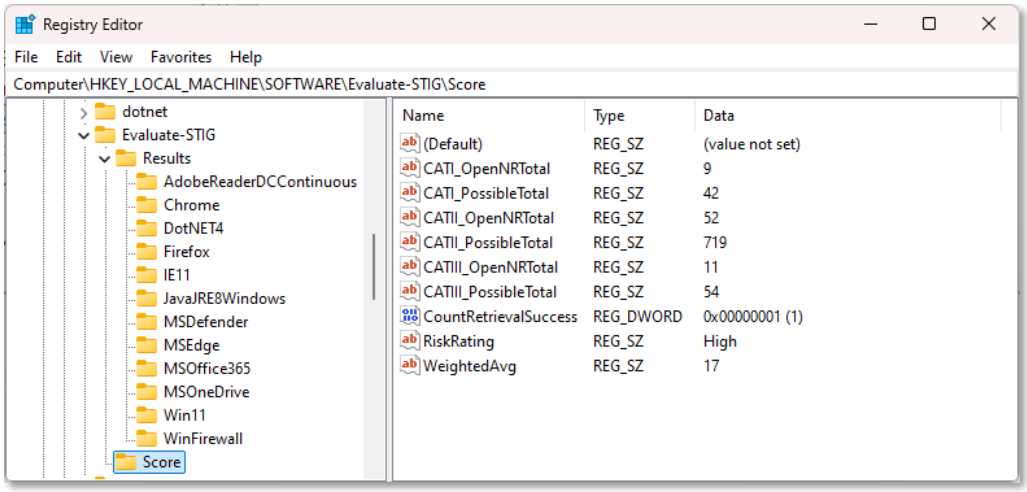
ApplicableSTIGs	List of STIG shortnames Evaluate-STIG determined as required
FullScan	Whether or not a full scan was conducted (all ApplicableSTIGs were processed)
LastCommand	Evaluate-STIG command line that was executed
LastRun	Date and time of last scan
ProcessedSTIGs	List of STIG shortnames that were included in the last scan
Version	Version of Evaluate-STIG used for last scan
Results\<STIG>\CATI_OpenNRTotal	Count of Open and Not Reviewed CAT I check results for STIG
Results\<STIG>\CATII_OpenNRTotal	Count of Open and Not Reviewed CAT II check results for STIG
Results\<STIG>\CATIII_OpenNRTotal	Count of Open and Not Reviewed CAT III check results for STIG
Results\<STIG>\EvalScore	Percent of completed checks (NA and NF status)
Results\<STIG>\Release	Release of the STIG
Results\<STIG>\ReleaseDate	Release date of the STIG
Results\<STIG>\Version	Version of the STIG
Score\CountRetrievalSuccess	Whether or not the full count of checks was retrieved for grading
Score\RiskRating	CORA risk rating for the asset based on Evaluate-STIG applicable STIGs
Score\WeightedAvg	CORA weighted average percentage based on Evaluate-STIG applicable STIGs
Score\CATI_OpenNRTotal	Total Open|Not Reviewed CAT I checks across all applicable STIGs
Score\CATI_PossibleTotal	Total CAT I checks across all applicable STIGs
Score\CATII_OpenNRTotal	Total Open|Not Reviewed CAT II checks across all applicable STIGs
Score\CATII_PossibleTotal	Total CAT II checks across all applicable STIGs
Score\CATIII_OpenNRTotal	Total Open|Not Reviewed CAT III checks across all applicable STIGs
Score\CATIII_PossibleTotal	Total CAT III checks across all applicable STIGs

Windows

Location: `HKLM:\SOFTWARE\Evaluate-STIG`

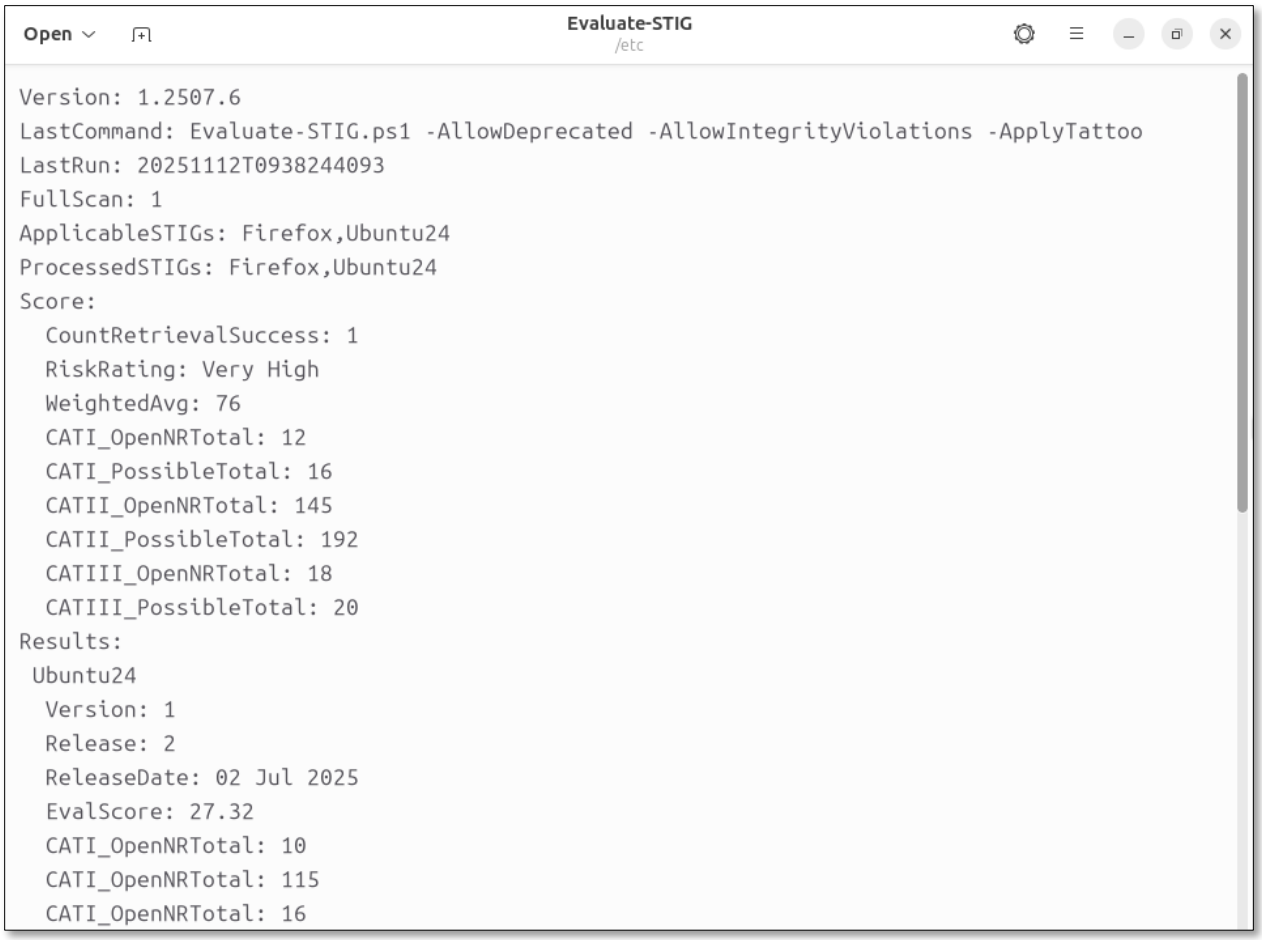






Linux

Location: `/etc/Evaluate-STIG`



Appendix A: Frequently Asked Questions

Q: Can Evaluate-STIG configure an asset to be compliant with the STIG?

A: No. Evaluate-STIG only documents the compliance state of an asset. It is not a tool to make your machine compliant.

Q: Why are the Active Directory Domain and Forest STIGs not scanned on a backup domain controller?

A: Much of our STIG applicability check is determined by special criteria outlined in the STIG's Overview.pdf. For Active Directory STIGs, the Overview.pdf has this note – "The requirements at the Active Directory Domain level are generally settings configured on a specific domain controller or replicated across domain controllers after configuration. They can typically be reviewed once per domain". With Evaluate-STIG, we will produce the AD STIGs on the domain controller holding the PDC Emulator role.

Q: Can Evaluate-STIG remotely scan non-domain (workgroup) Windows computers?

A: Yes, provided additional configurations are in place that allow WinRM connectivity between the hosts. These configurations are out-of-scope for this guide but this article should help - https://woshub.com/using-psremoting-winrm-non-domain-workgroup/. Once you can connect to workgroup computers with **`Enter-PSSession`**, you should then be able to scan with Evaluate-STIG using **`-ComputerName`** and **`-AltCredential`** as normal.

Appendix B: Troubleshooting

B-1 Logging

Evaluate-STIG uses several logs to assist in troubleshooting. All logs are designed to be viewed with CMTrace, a log-viewing tool that comes with Microsoft Configuration Manager (ConfigMgr). If you do not have ConfigMgr, instructions on ways to obtain `CMTrace.exe` can be found at https://scriptingnerd.com/2023/02/02/how-to-download-cmtrace-from-microsoft/.

Local Scan

All local scan logging is recorded in the `Evaluate-STIG.log` under the following paths:

- **Windows:** `%WINDIR%\Windows\Temp\Evaluate-STIG\Evaluate-STIG.log`
- **Linux:** `/tmp/Evaluate-STIG/Evaluate-STIG.log`

After scan completion, the `Evaluate-STIG.log` will also be copied to `[OutputPath]`.

Remote Scan

Remote scan logging consists of the `Evaluate-STIG_Remote.log` on the host and the local log on the remotes. The `Evaluate-STIG_Remote.log` contains session information for each remote and is located:

- **Windows:** `%TEMP%\Evaluate-STIG\Evaluate-STIG_Remote.log` (*this launching user's temp*)

After scan completion, the `Evaluate-STIG.log` for each remote will also be copied to its folder in `[OutputPath]`.

Cisco Scan

Cisco scan logging consists of the Evaluate-STIG_Cisco.log and an Evaluate-STIG.log for each Cisco hostname all on the host machine:

- **Windows:**
  - `%TEMP%\Evaluate-STIG\Evaluate-STIG_Cisco.log` (*this launching user's temp*)
  - `%TEMP%\Evaluate-STIG\CiscoScanTemp\<Hostname>\Evaluate-STIG.log`
- **Linux:**
  - `/tmp/Evaluate-STIG/Evaluate-STIG_Cisco.log`
  - `/tmp/Evaluate-STIG/CiscoScanTemp/<Hostname>/Evaluate-STIG.log`

After scan completion, the `Evaluate-STIG.log` for each Cisco host will also be copied to its folder in `[OutputPath]`.

B-2 Common Problems

Issue:
When trying to run Evaluate STIG, it will immediately fail and give ***Cannot index into a null array***

Resolution:
Could be that one or more files have the blocked attribute set. Run `Test-Prerequisites.bat` to check if any files are blocked.

Issue:
When trying to a remote scan using `-ComputerName`, file compression fails with ***Exception calling ".ctor" with "1" argument(s): "Stream was not readable."***

Resolution:
This is typically antivirus interfering with the compression process.

Issue:
No CKL files are produced.

Resolution:
Ensure you are specifying what output you want Evaluate-STIG to produce with the `-Output` parameter.

Issue:
Remote scan fails to connect or does not return results

Resolution:
Ensure WinRM is enabled that connections are not being blocked by a firewall. Test connectivity with the `Enter-PSSession` command. If connection with `Enter-PSSession` fails, this must be resolved before Evaluate-STIG can perform a remote scan.

Resolution:
Execution policy and/or code signing certificate issues on remote host. Verify prerequisites are met on remotes.

Resolution:
Check remote logs for clues.

Issue:
PostgreSQL scan hangs or fails with ***Local trust authentication method must be set in <path>\pg_hba.conf***

Resolution:
A "host" or "local" configuration line must exist in pg_hba.conf and be set to "trust" authentication. This configuration allows Evaluate-STIG to connect to the database without password prompting. After scanning the configuration line can be disabled (#) or removed.

```
#An example of local scanning:
local    all    all                    trust
host     all    all   127.0.0.1/32     trust
host     all    all   ::1/128          trust

#To allow remote scanning:
host     all    all   [ip-of-evaluate-stig-system]    trust
host     all    all   192.168.0.123/32    trust
```

Issue:
Remote scan fails with ***The WinRM client sent a request to the remote WS-Management service and was notified that the request size exceeded the configured MaxEnvelopeSize quota.***

Resolution:
Configure the source and remote machines to the same WSMan MaxEnvelope size (default is 500):

```
PS C:\> Set-Item WSMan:\localhost\MaxEnvelopeSizekb –Value 500
```

https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-wsman/8a6b1967-ff8e-4756-9a3b-890b4b439847

Appendix C: Technical Support

- **SPORK Ticket** (best way to raise an issue, report a bug):
  https://spork.navsea.navy.mil/nswc-crane-division/evaluate-stig/-/issues
  (You can register for a SPORK account at https://reg.fusion.navy.mil)

- **Microsoft Teams** (Navy Flank Speed):
  https://dod.teams.microsoft.us/l/channel/19%3adod%3a5d219fc1ee444d86a4db8f325ba51ceb%40thread.skype/Evaluate%2520STIG?groupId=f8d63861-f4a7-4af4-bb9a-35433e24f6f1&tenantId=e3333e00-c877-4b87-b6ad-45e942de1750

- **Fusion Chat**:
  https://chat.navsea.navy.mil/channel/evaluate-stig
  (You can register for a SPORK account at https://reg.fusion.navy.mil)

- **Email**:
  eval-stig_spt@us.navy.mil

Appendix D: Supported STIGs

STIG	Version	Date	DisaStatus *
Active Directory Domain	V3R5	13-Sep-24	Active
Active Directory Forest	V3R2	2-Jul-25	Active
Adobe Acrobat Pro XI	V1R2	26-Jan-18	Deprecated
Adobe Acrobat Professional DC Classic Track	V2R1	23-Oct-20	Deprecated
Adobe Acrobat Professional DC Continuous Track	V2R1	23-Jul-21	Active
Adobe Acrobat Reader DC Classic Track	V2R1	23-Oct-20	Deprecated
Adobe Acrobat Reader DC Continuous Track	V2R1	23-Jul-21	Active
Amazon Linux 2023	V1R1	14-Jul-25	Active
Apache Server 2.4 UNIX Server	V3R2	30-Jan-25	Active
Apache Server 2.4 UNIX Site	V2R6	2-Apr-25	Active
Apache Server 2.4 Windows Server	V3R3	2-Apr-25	Active
Apache Server 2.4 Windows Site	V2R2	2-Apr-25	Active
Apache Tomcat Application Server 9	V3R2	2-Apr-25	Active
ArcGIS for Server 10.3	V2R1	26-Jul-23	Sunset
Canonical Ubuntu 16.04 LTS	V2R3	23-Apr-21	Deprecated
Canonical Ubuntu 18.04 LTS	V2R15	24-Jul-24	Sunset
Canonical Ubuntu 20.04 LTS	V2R3	2-Jul-25	Active
Canonical Ubuntu 22.04 LTS	V2R5	2-Jul-25	Active
Canonical Ubuntu 24.04 LTS	V1R2	2-Jul-25	Active
Cisco IOS Router NDM	V3R5	2-Jul-25	Active
Cisco IOS Router RTR	V3R3	30-Jan-25	Active
Cisco IOS Switch L2S	V3R1	24-Jul-24	Active
Cisco IOS Switch NDM	V3R5	2-Jul-25	Active
Cisco IOS Switch RTR	V3R1	24-Jul-24	Active
Cisco IOS XE Router NDM	V3R5	2-Jul-25	Active
Cisco IOS XE Router RTR	V3R4	2-Jul-25	Active
Cisco IOS XE Switch L2S	V3R2	2-Jul-25	Active
Cisco IOS XE Switch NDM	V3R4	2-Jul-25	Active
Cisco IOS XE Switch RTR	V3R2	2-Jul-25	Active
Citrix Virtual Apps and Desktop 7.x Workspace App	V1R3	2-Jul-25	Sunset
Google Chrome Current Windows	V2R11	2-Jul-25	Active
JBoss Enterprise Application Platform 6.3	V2R6	2-Apr-25	Active
McAfee VirusScan 8.8 Local Client	V6R1	27-Jan-22	Deprecated
Microsoft Access 2013	V1R7	24-Jul-24	Deprecated
Microsoft Access 2016	V1R1	14-Nov-16	Active
Microsoft Defender Antivirus	V2R6	24-Sep-25	Active
Microsoft DotNet Framework 4.0	V2R7	2-Jul-25	Active

Microsoft Edge	V2R3	2-Jul-25	Active
Microsoft Excel 2013	V1R8	24-Jul-24	Deprecated
Microsoft Excel 2016	V2R1	24-Apr-24	Active
Microsoft Exchange 2016 Edge Transport Server	V2R6	30-Jan-25	Active
Microsoft Exchange 2016 Mailbox Server	V2R6	24-Jan-24	Active
Microsoft Exchange 2019 Edge Server	V2R2	30-Jan-25	Active
Microsoft Exchange 2019 Mailbox Server	V2R3	2-Jul-25	Active
Microsoft Groove 2013	V1R4	30-Jan-25	Deprecated
Microsoft IIS 10.0 Server	V3R4	2-Jul-25	Active
Microsoft IIS 10.0 Site	V2R12	2-Jul-25	Active
Microsoft IIS 8.5 Server	V2R7	25-Oct-23	Deprecated
Microsoft IIS 8.5 Site	V2R9	25-Oct-23	Deprecated
Microsoft InfoPath 2013	V1R6	24-Jul-24	Deprecated
Microsoft Internet Explorer 11	V2R5	24-Jan-24	Active
Microsoft Lync 2013	V1R5	24-Jul-24	Deprecated
Microsoft Office 365 ProPlus	V3R3	2-Apr-25	Active
Microsoft Office System 2013	V2R2	24-Jul-24	Deprecated
Microsoft Office System 2016	V2R4	30-Jan-25	Active
Microsoft OneDrive	V2R4	2-Jul-25	Sunset
Microsoft OneNote 2013	V1R4	30-Jan-25	Deprecated
Microsoft OneNote 2016	V1R2	19-Jan-17	Active
Microsoft Outlook 2013	V1R14	30-Jan-25	Deprecated
Microsoft Outlook 2016	V2R3	27-Apr-22	Active
Microsoft PowerPoint 2013	V1R7	24-Jul-24	Deprecated
Microsoft PowerPoint 2016	V1R1	14-Nov-16	Active
Microsoft Project 2013	V1R5	24-Jul-24	Deprecated
Microsoft Project 2016	V1R1	14-Nov-16	Active
Microsoft Publisher 2013	V1R6	24-Jul-24	Deprecated
Microsoft Publisher 2016	V1R3	27-Apr-18	Active
Microsoft SharePoint 2013	V2R4	30-Jan-25	Deprecated
Microsoft Skype for Business 2016	V1R1	14-Nov-16	Active
Microsoft SQL Server 2022 Database	V1R1	27-May-25	Active
Microsoft SQL Server 2022 Instance	V1R1	27-May-25	Active
Microsoft Visio 2013	V1R5	24-Jul-24	Deprecated
Microsoft Visio 2016	V1R1	14-Nov-16	Active
Microsoft Windows 10	V3R4	2-Apr-25	Active
Microsoft Windows 11	V2R4	2-Jul-25	Active
Microsoft Windows Defender Firewall with Advanced Security	V2R2	9-Nov-23	Active
Microsoft Windows Server 2012/2012 R2 Domain Controller	V3R7	9-Nov-23	Deprecated
Microsoft Windows Server 2012/2012 R2 Member Server	V3R7	9-Nov-23	Deprecated
Microsoft Windows Server 2016	V2R10	15-Jan-25	Sunset

Microsoft Windows Server 2019	V3R5	2-Jul-25	Active
Microsoft Windows Server 2022	V2R5	2-Jul-25	Active
Microsoft Windows Server Domain Name System (DNS)	V2R3	2-Apr-25	Active
Microsoft Word 2013	V1R7	24-Jul-24	Deprecated
Microsoft Word 2016	V1R1	14-Nov-16	Active
MongoDB Enterprise Advanced 3.x	V2R3	24-Jul-24	Sunset
Mozilla Firefox	V6R6	2-Apr-25	Active
MS SharePoint Designer 2013	V1R3	27-Apr-18	Deprecated
MS SQL Server 2014 Database	V1R7	24-Jul-24	Sunset
MS SQL Server 2014 Instance	V2R4	24-Jul-24	Sunset
MS SQL Server 2016 Database	V3R3	2-Jul-25	Active
MS SQL Server 2016 Instance	V3R5	2-Jul-25	Active
Oracle Java JRE 8 for Unix	V1R3	27-Oct-17	Deprecated
Oracle Java Runtime Environment (JRE) Version 8 for Windows	V2R1	22-Jan-21	Deprecated
Oracle Linux 7	V3R3	2-Jul-25	Active
Oracle Linux 8	V2R5	2-Jul-25	Active
Oracle Linux 9	V1R2	2-Jul-25	Active
PostgreSQL 9.x	V2R5	24-Jul-24	Sunset
Rancher Government Solutions RKE2	V2R3	30-Jan-25	Active
Red Hat Enterprise Linux 7	V3R15	24-Jul-24	Sunset
Red Hat Enterprise Linux 8	V2R4	2-Jul-25	Active
Red Hat Enterprise Linux 9	V2R5	2-Jul-25	Active
Trellix ENS 10.x Local Client	V2R4	2-Jul-25	Active
VMware Horizon 7.13 Agent	V1R1	13-Jul-21	Sunset
VMware Horizon 7.13 Client	V1R1	13-Jul-21	Sunset
VMware Horizon 7.13 Connection Server	V1R2	24-Apr-24	Sunset
Windows 7	V1R32	17-Jun-20	Deprecated
Windows Server 2008 R2 Member Server	V1R33	17-Jun-20	Deprecated

* STIGs no longer available on https://www.cyber.mil are considered Deprecated.