# The beginning of this project

Since my new job (a few years ago), my company used to create pre-configured Endpoint Security packages with trac.config (an encrypted file that contains the VPN site config) included using AdminMode.bat.

Then we migrated to managed VPN clients by using SmartEndpoint (because of the firewall included)

Then we had to use vpnconfig (<https://support.checkpoint.com/results/sk/sk122574>) because we also edited trac.defaults file

But it was a difficult task to provide a new VPN client because of the steps amount:

* Generate a package from SmartEndpoint
* Install it (to generate trac.defaults new version)
* Connect one time
* Change several settings in trac.defaults and get trac.config
* Use vpnconfig to create a new msi
* Uninstall Endpoint Security
* Install the newly vpnconfig generated msi to verify everything works as expected.

# First versions of the script I made

So at the beginning I created a script to automate trac.defaults edition after Endpoint Security installation (bacically I ran a powershell script that launched EPS.msi and modified trac.defaults after EPS.msi installation finished).

This feature is still possible today, but I found a way to extract trac.defaults from the EC.cab included in the MSI, and I used autoit to automate the vpnconfig tool window.

Then I found that the embedded trac.config from a EPS.msi file generated through SmartEndpoint was not encrypted, but it did not contain the certificate fingerprint (so a popup appeared to accept the certificate).

Then I found the perfect registry key:

HKLM:\Software\WOW6432Node\CheckPoint\accepted\_cn

It contains all accepted certificates

I had everything ready to automate all what I was using inside vpnconfig.exe

# Current version of the script

Packages generated using a MSI were huge for a VPN client and firewall. It was around 450MB. For an issue we had to try the newest version of Endpoint Security and I discovered only a MSI of 900MB, and all packages for previous versions were removed to only provide a 900MB MSI file.

This script also manages dynamic packages. 7zip is used to unpack the dynamic package, and used to recreate an EXE package

Antimalware signature is always included by packages generated through SmartEndpoint, even if there is no antivirus included, same for a CAB file used by FDE. The script asks if you want to keep these items, and remove them if you want. It allows generating a 100MB package including VPN and firewall.

# Folder/files structure

This project is composed of two scripts:

* CheckPoint\_CustomizePackage.ps1
* Install-CheckPointEndpointSecurity.ps1

The first script is the one you wan to run. The second is copied automatically at the end of package generation, and you will use it to run setup on computers.

Here is the folder list:

|  |  |
| --- | --- |
| Folder | Details |
| Input | Contains response files used by the script |
| Sources | Will contain generated packets |
| Tools | Contains 7-Zip |
| UDF | Contains generic functions used by the script |

Let’s describe the most interesting folder, the “input” folder:

|  |  |
| --- | --- |
| Folder | Details |
| client\_configuration | Contains settings to edit trac.defaults |
| install\_general\_config | Contains settings like registry tagging and log folder for installer |
| install\_steps\_after | This folder contains actions that will be ran after installation of EPS.msi |
| install\_steps\_before | This folder contains actions that will be ran before installation of EPS.msi |
| package\_customization\_msi | Allows to change SDL and Fixed MAC (similar to VPNconfig.exe). NoKeep is the override trac.config option |
| package\_customization\_post\_actions | Actions ran at the end of package generation. Can be used to copy items ran during installation or generate an EXE runner that will run Install-CheckPointEndpointSecurity.ps1 |
| site | Contains VPN sites that can be included inside packages |

# How it works

Because of the way packages are made, steps are not exactly the same but similar

First step: open a powershell window

Then run

.\CheckPoint\_CustomizePackage.ps1

When you open the script for the first time, you will only be able to add an external file :

------- Please select an item: -------

No Check Point package in input folder

--------------------------------------

Select Another File

Press Enter or “o” to select another file

Then enter the file path of an Endpoint package generated through SmartEndpoint (or the webui version), and press enter:

------- Please select an item: -------

No Check Point package in input folder

--------------------------------------

Select Another File

Please enter a file path: "I:\Scripts\PowerShell\input\CheckPoint\_package\EPS\_E87.00\_SRV-CP\EPS.msi"

Double quotes are not mandatory. They are added when you copy a file path using “Ctrl” + “Shift” + “C” under Windows 11, so I added support for path with double quotes.

Next steps will depend on if you selected a MSI or an EXE file

## MSI

The script will guess a package folder name. The package will be VPN (if this is not a managed package) or EPS (if it’s a package generated by SmartEndpoint), the package version and the SmartEdpoint server name (if the package is managed by an Endpoint server):

Selected package to customize: other file (I:\Scripts\PowerShell\input\CheckPoint\_package\EPS\_E87.00\_SRV-CP\EPS.msi)

Selected outut package name is: EPS\_E87.00\_SRV-CP

Do you accept this package name as output folder?

Yes, use "EPS\_E87.00\_SRV-CP" as output folder

No, choose another folder name

You can press Y or N, or use arrows and press enter to validate

If you press N, you will be asked to type the output package folder name

Then you can copy the input file selected in a repository for future usage (for example if you need to generate different packages with the same MSI):

Selected package name = EPS\_E87.00\_SRV-CP

Do you want to copy the MSI in the input folder (for later use)?

Yes, copy package in input folder

No, don't copy for later use

You can press Y or N, or use arrows and press enter to validate

If you press Y, package will be copied to the “input\CheckPoint\_package” folder

You will see these lines during file copy:

Copy EPS.msi to repository - start

Copy EPS.msi to repository - end

Then you will need to choose which site you will put inside generated package:

------------- Please select a VPN site -----------

Site Authentication Method Details

vpn.example.com certificate Logs = basic

vpn.example.com username-password Logs = basic

---------------------- 1 / 1 ---------------------

Please use arrows to select the site, and press enter to validate.

You will have a confirmation about the site you selected:

Selected site:

Site Authentication Method Details

vpn.example.com certificate Logs = basic

This step is used to select some configuration used during Endpoint Security installation. At my company we use SCCM to deploy things and wanted to tag the registry at the end of installation, and we also wanted to change the path where logs are generated during installation

- Please select a general setup configuration file (log folders and SCCM tag) -

Description

Example

Example Consultant

------------------------------------ 1 / 1 ------------------------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected setup general config:

Description

Example

Then you have to select steps that will be ran before executing the final MSI:

- Please select steps to run before install -

Description

Example

Example (Reconnect to SRV-CP)

------------------- 1 / 1 -------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected steps to run before install:

Description

Example

Then you have to select step that will be ran after executing the MSI

- Please select steps to run after install -

Description

Example

Example Consultant

------------------- 1 / 1 ------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected steps to run after install:

Description

Example

Then please select a configuration file that contains trac.defaults properties you want to apply:

- Please select a Check Point configuration file (for trac.defaults customization) -

Name Details

External users

Example Filtered cerfiticates, machine auth enabled

Example Filtered cerfiticates, machine auth and always connect disabled

Example Filtered cerfiticates, machine auth disabled

--------------------------------------- 1 / 1 --------------------------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected Check Point configuration file:

Name Details

Example Filtered cerfiticates, machine auth enabled

The selected properties above can be integrated inside the MSI (the default version of trac.defaults will be extracted from the package, edited and put inside the new MSI), or can be set after installation (the service will be terminated after installation, file will be edited and the service will be restarted) :

Do you want to change trac.defaults inside MSI?

Yes, integrate it inside MSI

No, change trac.defaults after setup

I recommend editing the MSI. Please enter Y or N, or user arrows and Enter to validate.

When you use vpnconfig, several settings were possible: SDL\_Enabled and FIXED\_MAC (and an option to disable OfficeMode but I don’t think anybody will want to disable OfficeMode). This is also the step where you can choose if you want to override the currently installed trac.config.

- Please select a configuration file to customize the MSI Properties -

Description

Example (SDL Disabled)

Example (SDL Enabled)

-------------------------------- 1 / 1 -------------------------------

Please use arrows and press Enter to select a configuration file.

The option you selected will be displayed:

Selected customization for MSI file:

Description

Example (SDL Disabled)

This step allows to run actions at the end of package generation. I use it to copy the VPN session opening script, an exe to reconnect to a management and generate a runner:

- Please select custom actions to be ran after package generation -

Description

External EPS (Build runner only)

Example (Copy things that will be used during setup)

Example (Copy things, Generate runner)

------------------------------ 1 / 1 ------------------------------

Here is a documentation about how to create an EXE to reconnect to a management server:

<https://sc1.checkpoint.com/documents/Infinity_Portal/WebAdminGuides/EN/Harmony-Endpoint-Admin-Guide/Topics-HEP/Reconnect-Tool.htm>

What I call a runner is an EXE that can be used by users with admin rights to run the whole setup (by running the generated PS1). If you choose to do it (at the next step), the runner can also integrate the uninstall password (mandatory when you do upgrades).

Please use arrows and Enter to select an option. Then the option you selected will be displayed:

Selected post actions file:

Description

Example (Copy things, Generate runner)

Now you can integrate the uninstall password in the generated package. When you use a MSI, the password can only be integrated in the runner created (if the actions after package generation contains a task to create a runner):

If you upgrade a computer, please type the uninstall password

Password :

Ok, integrate it in PS1 launcher

Continue without password

You can choose not to integrate password. You can add the password by using the command line if you don’t integrate in the runner.

Then the package will be generated:

Copy MSI

Customize MSI - Add trac.config

Customize MSI - Apply MSI Properties

Copy install

Copy config.json

Running package customization post-actions

Copy vpn connection script

Generate EXE script to run the PS1 file

Package customization end

## EXE

Once an EXE has been selected, the chosen EXE will be unpacked using 7-Zip:

Selected package to customize: other file (I:\Scripts\PowerShell\input\CheckPoint\_package\EPS\_E88.60\_SRV-CP\EPS\_88\_60\_0087\_Laptop\_313.exe)

7-Zip (a) 24.08 (x86) : Copyright (c) 1999-2024 Igor Pavlov : 2024-08-11

Scanning the drive for archives:

1 file, 573207520 bytes (547 MiB)

Extracting archive: I:\Scripts\PowerShell\input\CheckPoint\_package\EPS\_E88.60\_SRV-CP\EPS\_88\_60\_0087\_Laptop\_313.exe

WARNINGS:

There are data after the end of archive

--

Path = I:\Scripts\PowerShell\input\CheckPoint\_package\EPS\_E88.60\_SRV-CP\EPS\_88\_60\_0087\_Laptop\_313.exe

Type = 7z

WARNINGS:

There are data after the end of archive

Offset = 281850

Physical Size = 572924173

Tail Size = 1497

Headers Size = 12275

Method = LZMA:24

Solid = +

Blocks = 1

52% 224 - AM2.Signatures\SPSFullSignature.exe

When it will be 100% unpacked, 7-Zip will say everything is OK

Everything is Ok

Archives with Warnings: 1

Warnings: 1

Folders: 106

Files: 977

Size: 762440155

Compressed: 573207520

At this step you need to select the VPN site to be included:

------------- Please select a VPN site -----------

Site Authentication Method Details

vpn.example.com certificate Logs = basic

vpn.example.com username-password Logs = basic

--------------------- 1 / 1 ----------------------

Please use arrows to select the site, and press enter to validate.

You will have a confirmation about the site you selected:

Selected site:

Site Authentication Method Details

vpn.example.com certificate Logs = basic

This step is used to select some configuration used during Endpoint Security installation. At my company we use SCCM to deploy things and wanted to tag the registry at the end of installation, and we also wanted to change the path where logs are generated during installation

- Please select a general setup configuration file (log folders and SCCM tag) -

Description

Example

Example Consultant

------------------------------------ 1 / 1 ------------------------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected setup general config:

Description

Example

Then you have to select steps that will be ran before executing the final MSI:

- Please select steps to run before install -

Description

Example

Example (Reconnect to SRV-CP)

------------------- 1 / 1 -------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected steps to run before install:

Description

Example

Then you have to select step that will be ran after executing the MSI

- Please select steps to run after install -

Description

Example

Example Consultant

------------------- 1 / 1 ------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected steps to run after install:

Description

Example

Then please select a configuration file that contains trac.defaults properties you want to apply:

- Please select a Check Point configuration file (for trac.defaults customization) -

Name Details

External users

Example Filtered cerfiticates, machine auth enabled

Example Filtered cerfiticates, machine auth and always connect disabled

Example Filtered cerfiticates, machine auth disabled

--------------------------------------- 1 / 1 --------------------------------------

Please use arrows and Enter to select the configuration you want. Then it will be displayed:

Selected Check Point configuration file:

Name Details

Example Filtered cerfiticates, machine auth enabled

The selected properties above can be integrated inside the MSI (the default version of trac.defaults will be extracted from the package, edited and put inside the new MSI), or can be set after installation (the service will be terminated after installation, file will be edited and the service will be restarted) :

Do you want to change trac.defaults inside MSI?

Yes, integrate it inside MSI

No, change trac.defaults after setup

I recommend editing the MSI. Please enter Y or N, or user arrows and Enter to validate.

When you use vpnconfig, several settings were possible: SDL\_Enabled and FIXED\_MAC (and an option to disable OfficeMode but I don’t think anybody will want to disable OfficeMode). This is also the step where you can choose if you want to override the currently installed trac.config.

- Please select a configuration file to customize the MSI Properties -

Description

Example (SDL Disabled)

Example (SDL Enabled)

-------------------------------- 1 / 1 -------------------------------

Please use arrows and press Enter to select a configuration file.

The option you selected will be displayed:

Selected customization for MSI file:

Description

Example (SDL Disabled)

This step allows to run actions at the end of package generation. I use it to copy the VPN session opening script, an exe to reconnect to a management and generate a runner:

- Please select custom actions to be ran after package generation -

Description

External EPS (Build runner only)

Example (Copy things that will be used during setup)

Example (Copy things, Generate runner)

------------------------------ 1 / 1 ------------------------------

Here is a documentation to create an EXE to reconnect to a management server: <https://sc1.checkpoint.com/documents/Infinity_Portal/WebAdminGuides/EN/Harmony-Endpoint-Admin-Guide/Topics-HEP/Reconnect-Tool.htm>

What I call a runner is an EXE that can be used by users with admin rights to run the whole setup (by running the generated PS1). If you choose to do it (at the next step), the runner can also integrate the uninstall password (mandatory when you do upgrades).

Please use arrows and Enter to select an option. Then the option you selected will be displayed:

Selected post actions file:

Description

Example (Copy things, Generate runner)

The script will guess a package folder name. The package will be VPN (if this is not a managed package) or EPS (if it’s a package generated by SmartEndpoint), the package version and the SmartEdpoint server name (if the package is managed by an Endpoint server):

Selected outut package name is: EPS\_E88.60\_SRV-CP

Do you accept this package name as output folder?

Yes, use "EPS\_E88.60\_SRV-CP" as output folder

No, choose anotyher folder name

You can press Y or N, or use arrows and press enter to validate

If you press N, you will be asked to type the output package folder name

If you choose an EXE outside the repository, you will be asked if you want to copy the input file in a repository for future usage (for example if you need to generate different packages with the same MSI):

Selected package name = EPS\_E88.60\_SRV-CP

Do you want to copy the EXE in the input folder (for later use)?

Yes, copy package in input folder

No, don't copy for later use

If you select Yes, you will have two line two lines will be displayed to show the copy start and end

Copy EPS\_88\_60\_0087\_Laptop\_313.exe to repository - start

Copy EPS\_88\_60\_0087\_Laptop\_313.exe to repository – end

Several steps will be done based on the previous answers:

Customize trac.defaults in unpacked folder

Customize trac.config in unpacked folder

Customize MSI - Apply MSI Properties

Check Point EXE generated by the SmartEndpoint (even with the Web version) may contain useless things (we do not use the antimalware blade but the generated EXE contains antimalware signatures). We also do not deploy FDE, and the SmartEndpoint console add a CAB for FDE Smart preboot (the web version of SmartEndpoint can not include it, but it is always included with SmartEndpoint). Please use arrows or Y/N keys to validate removal, or keep files:

Creating output folder

The package does not contains antimalware but contains signature. Do you want to remove signatures?

Yes, remove antimalware signatures

No, keep them inside package

The package does not contains FDE but contains FDE Smart preboot. Do you want to remove FDE Smart preboot?

Yes, remove FDE Smart preboot

No, keep them inside package

You can integrate the uninstall password inside the package. It will be useful when you upgrade an already installed Endpoint Security package. There is two places where you can integrate this password:

* install.exe (it will be included in the SFX parameters when the EXE will be generated)
* PS1 launcher: the EXE that will run the PS1 at the output root

If you type a password and press enter, the password will be integrated in the new EXE generated. If you don’t want to include a password, or include it in the PS1 launcher, please use arrows and press Enter to validate:

If you upgrade a computer, please type the uninstall password

Password :

Ok, integrate it in install.exe (default)

Ok, integrate it in PS1 launcher

Continue without password

Next step: a 7-zip file is created including all sources:

Creating EPS.7z

7-Zip (a) 24.08 (x86) : Copyright (c) 1999-2024 Igor Pavlov : 2024-08-11

Scanning the drive:

104 folders, 975 files, 308327675 bytes (295 MiB)

Creating archive: I:\Scripts\PowerShell\output\CheckPoint\_CustomizePackage\2.1\Sources\CheckPoint\_package\EPS\_E88.60\_SRV-CPO\EPS.7z

Add new data to archive: 104 folders, 975 files, 308327675 bytes (295 MiB)

11% 135 + ComplianceData\WindowsPatchData.zip

One 7zip file is created, it should say “Everything is Ok”:

Files read from disk: 975

Archive size: 102860870 bytes (99 MiB)

Everything is Ok

Next step : create the install.exe file based on the SFX config and the 7z file create before:

Adding SFX items to output folder

Merging everything to install.exe

I:\Scripts\PowerShell\output\CheckPoint\_CustomizePackage\2.1\tools\7-Zip\7zSD.sfx

I:\Scripts\PowerShell\output\CheckPoint\_CustomizePackage\2.1\Sources\CheckPoint\_package\EPS\_E88.60\_SRV-CP\sfxConfig.txt

I:\Scripts\PowerShell\output\CheckPoint\_CustomizePackage\2.1\Sources\CheckPoint\_package\EPS\_E88.60\_SRV-CP\EPS.7z

1 fichier(s) copié(s).

Cleaning temporary 7z file and sfxConfig file

Copy install

Copy config.json

Running package customization post-actions

Generate EXE script to run the PS1 file

# Configuration files

## File structure

At the root of each files, there are several items:

* configuration
* properties used to display information during script execution
  + Name
  + Details
  + Authentication Method
  + …
* filter

Configuration contains what will be configured

Filter can be used to not display some json files based on previous answers

During the whole script, a variable (a hashtable) is filled by some important values like the Endpoint server name, the VPN site, the package type.

The filter content is a small piece of powershell (with some backslash to escape double quotes, because double quotes end strings) to test the $Variables content. Here is an example to test the selected vpn site:



All configuration files accept comments

## Site

First three properties are used when you display the different json files when using the script.

{

"Site": "vpn.example.com",

"Authentication Method": "certificate",

"Details": "Logs = basic",

"configuration" : {

"site": "vpn.example.com",

"displayName": "vpn.example.com",

// Valid values : username-password, certificate, p12-certificate, challenge-response, securIDKeyFob, securIDPinPad, SoftID

"authenticationMethod": "certificate",

"sdl\_enabled": "false",

// Valid values : basic, extended

"debug\_mode": "basic"

}

}

The “configuration” node will contain data to create the site.

| Property | Description |
| --- | --- |
| site | This where you put either the DNS name or an IP of your VPN site |
| displayName | If you put an IP above, displaying a friendly name would be great |
| authenticationMethod | Here are all supported authentication methods: username-password, certificate, p12-certificate, challenge-response, securIDKeyFob, securIDPinPad, SoftID |
| sdl\_enabled | If you use certificates, you will have to disable SDL. If you use username-password, you can enable SDL here |
| debug\_mode | This is where you configure the log level. Possible options are “basic” or “extended”. However, you should not use “extended” as it will decrease VPN bandwidth (sk177125) |

## client\_configuration

This is where you can choose a file that will configure trac.defaults. Here is an example I use:

{

"name": "Example CN",

"details": "Filtered cerfiticates, machine auth enabled",

"configuration": {

"trac\_defaults":{

// Always connect

"neo\_always\_connected": "true",

// Hotspot registration

"hotspot\_detection\_enabled": "true",

"hotspot\_registration\_enabled": "true",

"open\_default\_browser\_for\_hotspot": "true",

"global\_hotspot\_detection\_enabled": "true",

// DNS management

"restart\_dns\_service\_on\_vna\_init": "true",

"flush\_dns\_cache": "\"true\"",

"do\_proxy\_replacement": "\"false\"",

// Certificate filtering

"cert\_filter\_issuer": "CN=Example Europe Certification Authority,\*&#CN=Example America Certification Authority,\*&#CN=Example Asia Certification Authority,\*&#O=srv-cp.example.fr,\*&#",

"cert\_filter\_subject": "",

"cert\_filter\_template": "Example User EU SHA-2&#Example User AM SHA-2&#Example User CN SHA-2&#\*&#",

"cert\_filter\_enhanced\_key\_usage": "",

"cert\_filter\_condition": "or",

"cert\_filter\_check": "1",

// Machine auth disabled

"enable\_machine\_auth": "true",

"machine\_tunnel\_site": "\"vpn.example.cn\"",

// Other certificate options

"display\_capi\_friendly\_name": "1",

"display\_expired\_certificates": "0",

"display\_client\_auth\_certificates\_only": "1",

// Other options

"save\_vpn\_user\_per\_sid": "true"

}

},

"Filter": "$Variables[\"site\"] -in @(\"vpn.example.cn\")"

}

Properties you want to change in trac.defaults files should be configured in configuration.trac\_defaults section.

Since E84.10, Endpoint Security can filter certificates show in the connection dialog. Go check sk169453 if you want to know more.

You will find more info for other parameters in sk75221

## install\_general\_config

{

"Description": "Example",

"install": {

"Log":{

"Folder": "C:\\Windows\\Example\\logs\\",

"FallbackFolder": "%TEMP%\\"

},

"tag": {

"dotag":false,

"ignoretag":true,

"Manufactured": "Example IT",

"PackageVersion": "1.0",

"RegFolder": "Software\\Example\\Applications"

}

}

}

During the Install-CheckPointEndpointSecurity.ps1 script running, all lines displayed in console window are logged in a LOG file written in Log.Folder. If this folder does not exist, log will be written in Log.FallbackFolder

Tag section can be used to mark the registry in the tag.RegFolder (if dotag is true)

## package\_customization\_msi

{

"Description": "Example WW+CN EPS (SDL Disabled)",

"configuration": {

"MSI\_customization": {

"SDL\_ENABLED": "NO",

"FIXED\_MAC": "NO",

"NoKeep": "YES"

}

}

}

This folder contains properties modified inside the MSI

vpnconfig allowed to modify several properties:

|  |  |  |
| --- | --- | --- |
| Option | MSI property name | Possible values |
| Secure Domain Logon | SDL\_ENABLED | NO|YES |
| Fixed MAC | FIXED\_MAC | NO|YES |
| No Office Mode | NO\_OFFICE\_MODE | 0|1 |

I never understood the “No Office Mode” option but you can also configure it if you want.

NoKeep is another property modified by vpnconfig. It is set to “YES” when you ticked the “overwrite user’s configuration when upgrading”.

## package\_customization\_post\_actions

{

"Description": "Example WW (Copy Reconnect and Run\_VPN\_Startup\_Script.exe)",

"configuration": {

"post-actions": [

{

"Action": "Copy-Item",

"MessageBefore": "Copy Reconnect script",

"Arguments": {

"Source": "%InputDir%\\scripts\\Reconnect\\Reconnect.exe",

"Destination": "%outputFolder%\\"

}

},

{

"Action": "Copy-Item",

"MessageBefore": "Copy vpn connection script",

"Arguments": {

"Source": "%InputDir%\\scripts\\vpn\_startup\_script\\Run\_VPN\_Startup\_Script.exe",

"Destination": "%outputFolder%\\Sources\\"

}

},

{

"Action": "New-Runner",

"MessageBefore": "Generate EXE script to run the PS1 file"

}

]

}

}

These files are used at the end of package creation.

Post-actions are functions included in CheckPoint\_CustomizePackage at the end of package creation. MessageBefore is a string that will be written to host before running each action.

There are two actions possible:

|  |  |
| --- | --- |
| Function | Details |
| Copy-Item | If you need to run files when you are installing Endpoint Security, you may need to copy them from a repository to the output folder.  Copy-Item will do this. |
| New-Runner | As the MessageBefore suggests, New-Runner will create an EXE to run the Install-CheckPointEndpointSecurity.ps1 script. This EXE is built using AutoIt (needs to be installed to run) and can include the uninstall password. All arguments passed to the EXE will be assigned to the PS1 |

## install\_steps\_before

{

"Description": "Example WW (Reconnect to SRV-CP)",

"install": {

"steps\_before": [

{

"Action": "Write-ProductToHost"

},

{

"Action": "Stop-EndpointSecurity",

"Condition": "$Variables[\"Product\"] -eq \"Check Point VPN\""

},

{

"Action": "Invoke-ExternalCommand",

"MessageBefore": "Reconnect to management",

"Arguments": {

"Command": "%PSScriptRoot%\\Reconnect.exe"

},

"Condition": "$Variables[\"Product\"] -ne \"\""

}

]

},

"Filter": "$Variables[\"site\"] -in @(\"vpn.example.com\")"

}

This folder contains files with actions ran just before install.exe or EPS.msi

Steps\_before are functions included in Install-CheckPointEndpointSecurity.ps1 that will be ran just before install.exe or EPS.msi. MessageBefore is a string that will be written to host before running each action.

Here are some actions available in the script:

|  |  |
| --- | --- |
| Function | Details |
| Write-ProductToHost | At the beginning or the installation, this function will display on the host the already installed product |
| Stop-EndpointSecurity | A very long time ago we upgraded from unmanaged Endpoint Security VPN to SmartEndpoint managed Endpoint Security  If I remember correctly, sometimes upgrade failed. We had to stop the Endpoint Security VPN service to allow upgrade. It might not be necessary anymore. The condition allows not to run this action if the already installed package is a managed package |
| Invoke-ExternalCommand | A few years ago, when we had to renew our internal CA, it broke the agent/server communication. We had to create and deploy a reconnect tool.  More information here for how to create such a tool : <https://sc1.checkpoint.com/documents/Infinity_Portal/WebAdminGuides/EN/Harmony-Endpoint-Admin-Guide/Topics-HEP/Reconnect-Tool.htm>  We also had several Endpoint versions at the same time, and upgrading an Endpoint where communication was broken did not solve the issue. That’s why I created this action (and the Copy-Item action in the “package\_customization\_post\_actions” step). |

## install\_steps\_after

{

"Description": "Example WW",

"install": {

"steps\_after": [

{

"Action": "Copy-Item",

"MessageBefore": "Copying VPN connection script",

"Arguments": {

"Source": "%PSScriptRoot%\\Sources\\ Run\_VPN\_Startup\_Script.exe",

"OtherSource": "%PSScriptRoot%\\input\\CheckPoint\_Scripts\\vpn\_startup\_script\\ Run\_VPN\_Startup\_Script.exe",

"Destination": "%windir%\\EXAMPLE\\",

"MessageCopySuccess": "VPN connection script copied successfully",

"MessageCopyError": "File copy failed",

"MessageSourceNotFound": "Script file does not exists"

}

},

{

"Action": "Wait-TracSrvWrapperServiceRunning",

"MessageBefore": "Waiting service TracSrvWrapper running",

"Arguments": {

"Timeout": 60000

}

},

{

"Action": "Set-EndpointSDLState",

"MessageBefore": "Disabling SDL",

"Arguments": {

"Value": "disabled"

}

},

{

"Action": "Set-EndpointLogLevel",

"MessageBefore": "Changing log level to basic",

"Arguments": {

"Value": "basic"

}

},

{

"Action": "Add-CertificateFingerprint",

"MessageBefore": "Adding \"srv-cp VPN Certificate\" fingerprint",

"Arguments": {

"AcceptedCN": "srv-cp VPN Certificate",

"Fingerprint": "APS MET NEED SNAG WAYS…"

}

},

{

"Action": "Update-TracDefaults",

"MessageBefore": "Changing trac.defaults file"

}

]

},

"Filter": "$Variables[\"site\"] -in @(\"vpn.example.com\")"

}

install\_steps\_after are functions included in Install-CheckPointEndpointSecurity.ps1 that will be ran just after install.exe or EPS.msi. MessageBefore is a string that will be written to host before running each action.

Here are some actions available in the script:

|  |  |
| --- | --- |
| Action | Details |
| Copy-Item | We use post connection script. This function will allow copy from the source to the destination you want.  There are two options to select source because of the way I develop scripts. |
| Wait-TracSrvWrapperServiceRunning | If you want to run commands like trac.exe with arguments, you need the service to be running |
| Set-EndpointSDLState | This command can enable or disable Secure Domain Logon |
| Set-EndpointLogLevel | This command can change the log level |
| Add-CertificateFingerprint | You can find the fingerprint in this registry key:  HKLM\SOFTWARE\WOW6432Node\CheckPoint\accepted\_cn\  In the function arguments, “AcceptedCN“ will be the subkey name, and “Fingerprint” will contain the property value of the “--Fingerprint--” value: |
| Update-TracDefaults | This function will be used to change installed trac.defaults content. What will be changed is in the config.json configuration file.  If you choose to edit the trac.defaults inside the MSI, config.json will contains nothing and nothing will be done by this function.  If you have a lot of files inside “install\_steps\_after”, I recommend to keep this function call even if it does nothing |