Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Unprotect-CmsMessage'

PS:\>Get-HELP Unprotect-CmsMessage -Full

NAME

Unprotect-CmsMessage

SYNOPSIS

Decrypts content that has been encrypted by using the Cryptographic Message Syntax format.

SYNTAX

Unprotect-CmsMessage [-Content] <System.String> [[-To] <System.Management.Automation.CmsMessageRecipient[]>] [-IncludeContext] [<CommonParameters>]

Unprotect-CmsMessage [-EventLogRecord] <System.Management.Automation.PSObject> [[-To] <System.Management.Automation.CmsMessageRecipient[]>] [-IncludeContext]

[<CommonParameters>]

Unprotect-CmsMessage [-LiteralPath] <System.String> [[-To] <System.Management.Automation.CmsMessageRecipient[]>] [-IncludeContext] [<CommonParameters>]

Unprotect-CmsMessage [-Path] <System.String> [[-To] <System.Management.Automation.CmsMessageRecipient[]>] [-IncludeContext] [<CommonParameters>]

DESCRIPTION

The `Unprotect-CmsMessage` cmdlet decrypts content that has been encrypted using the Cryptographic Message Syntax (CMS) format.

The CMS cmdlets support encryption and decryption of content using the IETF standard format for cryptographically protecting messages, as documented by RFC5652
(https://tools.ietf.org/html/rfc5652).

The CMS encryption standard uses public key cryptography, where the keys used to encrypt content (the public key) and the keys used to decrypt content (the private
key) are separate. Your public key can be shared widely, and isn't sensitive data. If any content is encrypted with this public key, only your private key can decrypt
it. For more information, see Public-key cryptography (https://en.wikipedia.org/wiki/Public-key_cryptography).

`Unprotect-CmsMessage` decrypts content that has been encrypted in CMS format. You can run this cmdlet to decrypt content that you have encrypted by running the
`Protect-CmsMessage` cmdlet. You can specify content that you want to decrypt as a string, by the encryption event log record ID number, or by path to the encrypted
content. The `Unprotect-CmsMessage` cmdlet returns the decrypted content.

Support for Linux and macOS was added in PowerShell 7.1.

PARAMETERS

-Content <System.String>

Specifies an encrypted string, or a variable containing an encrypted string.

Required?	true
Position?	0
Default value	None
Accept pipeline input?	True (ByPropertyName, ByValue)

Accept wildcard characters? false

-EventLogRecord <System.Management.Automation.PSObject>

Specifies an event log record that contains a CMS encrypted message.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByValue)

Accept wildcard characters? false

-IncludeContext <System.Management.Automation.SwitchParameter>

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-LiteralPath <System.String>

Specifies the path to encrypted content that you want to decrypt. Unlike Path , the value of LiteralPath is used exactly as it's typed. No characters are interpreted as wildcard characters. If the path includes escape characters, enclose it in single quotation marks. Single quotation marks tell PowerShell not to interpret any characters as escape sequences.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-Path <System.String>

Specifies the path to encrypted content that you want to decrypt.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-To <System.Management.Automation.CmsMessageRecipient[]>

Specifies one or more CMS message recipients, identified in any of the following formats:

- An actual certificate (as retrieved from the certificate provider).

- Path to the a file containing the certificate.

- Path to a directory containing the certificate.

- Thumbprint of the certificate (used to look in the certificate store).

- Subject name of the certificate (used to look in the certificate store).

Required? false

Position? 1

Default value None

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,
ErrorAction, ErrorVariable, WarningAction, WarningVariable,
OutBuffer, PipelineVariable, and OutVariable. For more information, see
about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.Diagnostics.Eventing.Reader.EventLogRecord

System.String

You can pipe an object containing encrypted content to this cmdlet.

OUTPUTS

System.String

This cmdlet returns the unencrypted message.

NOTES

----------------- Example 1: Decrypt a message -----------------

```
$parameters = @{
  LiteralPath = "C:\Users\Test\Documents\PowerShell\Future_Plans.txt"
  To = '0f 8j b1 ab e0 ce 35 1d 67 d2 f2 6f a2 d2 00 cl 22 z9 m9 85'
}
Unprotect-CmsMessage -LiteralPath @parameters
```

Try the new Break All command

------ Example 2: Decrypt an encrypted event log message ------

```
$event = Get-WinEvent Microsoft-Windows-PowerShell/Operational -MaxEvents 1 |
    Where-Object Id -eq 4104
Unprotect-CmsMessage -EventLogRecord $event
```

Example 3: Decrypt encrypted event log messages using the pipeline

```
Get-WinEvent Microsoft-Windows-PowerShell/Operational |
    Where-Object Id -eq 4104 |
    Unprotect-CmsMessage
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.security/unprotect-cmsmessage?view=powershell-5.1&WT.mc_id=ps-gethelp

about_Providers

Get-CmsMessage

Protect-CmsMessage