AlwaysOn VPN: tunel użytkownika z ProfileXML

Poziom: zaawansowany **Dotyczy: Windows 11 Pro/Enterprise, CSP VPNv2**

Czas: ~40 min

AlwaysOn VPN łączy tunel automatycznie, bez klikania „Połącz" w interfejsie, na podstawie jednego pliku ProfileXML wgranego przez most WMI. Ta strona pokazuje pełny profil tunelu użytkownika dla Zielonej, skrypt, który go wdraża, i sposób sprawdzenia, że wszystko wstało tak, jak powinno.

ZAŁOŻENIA SCENARIUSZA

jedynym punktem wejścia tunelu jest pojedynczy `fw01` z jednym łączem (`wan1`, adres stały 203.0.113.10), na który wskazuje `vpn.zielona.example`. Trasa do 10.20.30.0/24 w profilu zakłada, że magazyn jest osiągalny przez `fw01` (VLAN 30 na zaporze). Przy magazynie jako osobnej lokalizacji ta sama trasa prowadzi dalej przez tunel do `rtr-mag01`: IPsec site-to-site: Pamiątkowo i Swarzędz między FortiGate a routerem innej firmy.

Skróty użyte na tej stronie (12)

Skrót	Co znaczy
CSP	w Windows: Configuration Service Provider, mechanizm, przez który MDM albo RMM zapisuje ustawienia systemu, na przykład profil VPN
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
EAP	Extensible Authentication Protocol: rodzina metod uwierzytelniania, dzięki której VPN albo Wi-Fi mogą pytać o hasło albo o certyfikat na wiele różnych sposobów
IKEv2	Internet Key Exchange w wersji 2: nowsza odmiana IKE, wbudowana w klienta VPN Windows, szybciej wraca do życia po zerwaniu łącza niż starsze protokoły VPN
MDM	Mobile Device Management: system zdalnego zarządzania telefonami i tabletami firmowymi
NRPT	Name Resolution Policy Table: tabela reguł Windows mówiąca, dla której domeny pytać o adres inny serwer DNS niż domyślny
RMM	Remote Monitoring and Management: platforma do zdalnego zarządzania i monitorowania stacji roboczych
VLAN	Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie

Skrót	Co znaczy
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej
Wi-Fi	Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii
WMI	Windows Management Instrumentation: interfejs do odczytu stanu Windowsa
XML	eXtensible Markup Language: format tekstowy do zapisu ustrukturyzowanych danych w znacznikach, tutaj używany przez PowerShell do zapisu zrzutu obiektów poleceniem Export-Clixml

Fiszka

□□ ZMIENIA SYSTEM

Wdrożenie profilu AlwaysOn VPN przez WMI: skrypt CimSession.CreateInstance()

Komendy w tym bloku wykonujemy całościowo, jako jeden skrypt PowerShell uruchomiony z uprawnieniami administratora, bo poszczególne kroki (znalezienie zalogowanego użytkownika, usunięcie starego profilu, zapis nowego) mają sens tylko razem, w tej kolejności.

```
# Dane tunelu użytkownika Zielonej
$profileName = "Zielona AlwaysOn VPN"
$servers = "vpn.zielona.example"
$dnsSuffix = "ad.zielona.example"
$domainName = ".ad.zielona.example"
$dnsServers = "10.20.10.11,10.20.10.12"

# Pobieramy konfigurację EAP z wcześniej ręcznie założonego, testowego połączenia VPN
$szablon = Get-VpnConnection -Name "Zielona VPN Test"
$KonfiguracjaEap = $szablon.EapConfigXmlStream.InnerXml

$profilNazwaEscaped = $profileName -replace ' ', '%20'

# ProfileXML w kolejności wymaganej przez CSP VPNv2: DnsSuffix, NativeProfile,
# AlwaysOn, RememberCredentials, Route, DomainNameInformation
```

```
$profileXML = @"
<VPNProfile>
  <DnsSuffix>$dnsSuffix</DnsSuffix>
  <NativeProfile>
    <Servers>$servers</Servers>
    <NativeProtocolType>IKEv2</NativeProtocolType>
    <Authentication>
      <UserMethod>Eap</UserMethod>
      <Eap>
        <Configuration>
        $KonfiguracjaEap
        </Configuration>
      </Eap>
    </Authentication>
    <RoutingPolicyType>SplitTunnel</RoutingPolicyType>
  </NativeProfile>
  <AlwaysOn>true</AlwaysOn>
  <RememberCredentials>true</RememberCredentials>
  <Route>
    <Address>10.20.10.0</Address>
    <PrefixSize>24</PrefixSize>
  </Route>
  <Route>
    <Address>10.20.20.0</Address>
    <PrefixSize>24</PrefixSize>
  </Route>
  <Route>
    <Address>10.20.30.0</Address>
    <PrefixSize>24</PrefixSize>
  </Route>
  <DomainNameInformation>
    <DomainName>$domainName</DomainName>
    <DnsServers>$dnsServers</DnsServers>
  </DomainNameInformation>
</VPNProfile>
"@

# Znaki specjalne trzeba zakodować, zanim profil trafi jako wartość właściwości CIM
$profileXMLZakodowany = $profileXML -replace '<', '&lt;' -replace '>', '&gt;' -replace
```

```
'"', '&quot;'

$namespaceName = "root\cimv2\mdm\dmmap"
$className = "MDM_VPNv2_01"
$nodeCSPURI = "./Vendor/MSFT/VPNv2"

# Ustalamy SID zalogowanego użytkownika, bo tunel użytkownika zapisujemy w jego
kontekście
$ZalogowanyUser = (Get-WmiObject Win32_ComputerSystem).UserName
$ObjUser = New-Object System.Security.Principal.NTAccount($ZalogowanyUser)
$Sid = $ObjUser.Translate([System.Security.Principal.SecurityIdentifier]).Value

$session = New-CimSession
$options = New-Object Microsoft.Management.Infrastructure.Options.CimOperationOptions
$options.SetCustomOption("PolicyPlatformContext_PrincipalContext_Type",
"PolicyPlatform_UserContext", $false)
$options.SetCustomOption("PolicyPlatformContext_PrincipalContext_Id", "$Sid", $false)

# Usuwamy poprzednią wersję profilu o tej samej nazwie, jeśli istnieje
$IstniejaceProfile = $session.EnumerateInstances($namespaceName, $className, $options)
foreach ($istniejacy in $IstniejaceProfile) {
    if ($istniejacy.InstanceID -eq $profilNazwaEscaped) {
        $session.DeleteInstance($namespaceName, $istniejacy, $options)
    }
}

# Zapis nowego profilu przez CimSession.CreateInstance(), nie przez New-CimInstance
$NowaInstancja = New-Object Microsoft.Management.Infrastructure.CimInstance $className,
$namespaceName
$NowaInstancja.CimInstanceProperties.Add([Microsoft.Management.Infrastructure.CimPropert
y]::Create("ParentID", $nodeCSPURI, "String", "Key"))
$NowaInstancja.CimInstanceProperties.Add([Microsoft.Management.Infrastructure.CimPropert
y]::Create("InstanceID", $profilNazwaEscaped, "String", "Key"))
$NowaInstancja.CimInstanceProperties.Add([Microsoft.Management.Infrastructure.CimPropert
y]::Create("ProfileXML", $profileXMLZakodowany, "String", "Property"))

$session.CreateInstance($namespaceName, $NowaInstancja, $options)
Write-Host "Profil $profileName wdrożony."
```

Zapisuje profil `Zielona AlwaysOn VPN` w węźle CSP `VPNv2`, w kontekście zalogowanego użytkownika, czyli jako tunel użytkownika, nie urządzenia.

Przykład:

```
Profil Zielona AlwaysOn VPN wdrożony.
```

VPN = Virtual Private Network: szyfrowany tunel do sieci firmowej · **WMI** = Windows Management Instrumentation: interfejs do odczytu stanu Windowsa · **CSP** = w Windows: Configuration Service Provider, mechanizm, przez który MDM albo RMM zapisuje ustawienia systemu, na przykład profil VPN

TYLKO ODCZYT

Sprawdzenie zainstalowanego profilu VPN: Get-VpnConnection

```
Get-VpnConnection -Name "NazwaProfiluVpn"
```

Pokazuje stan zapisanego profilu: nazwę serwera, typ tunelu, czy AlwaysOn jest włączony i aktualny status połączenia.

Przykład:

```
Name                : Zielona AlwaysOn VPN
ServerAddress       : vpn.zielona.example
AllUserConnection   : False
TunnelType          : Ikev2
AlwaysOn            : True
ConnectionStatus    : Connected
```

VPN = Virtual Private Network: szyfrowany tunel do sieci firmowej

TYLKO ODCZYT

Ręczne wybudzenie tunelu i sprawdzenie stanu: rasdial

```
rasdial "NazwaProfiluVpn"
```

Wymusza ręczne połączenie tunelu bez czekania na automatyczne wyzwolenie przez AlwaysOn, przydatne przy pierwszym teście świeżo wdrożonego profilu. Samo `rasdial` bez argumentów wypisuje listę aktualnie aktywnych połączeń.

Przykład:

```
Nawiązywanie połączenia z siecią Zielona AlwaysOn VPN...
Weryfikowanie nazwy użytkownika i hasła...
Rejestrowanie komputera w sieci...
Polecenie zostało wykonane pomyślnie.
```

rasdial = polecenie Windows łączące się z VPN i pokazujące status połączenia z linii poleceń

TYLKO ODCZYT

Sprawdzenie aktywnych reguł NRPT: Get-DnsClientNrptPolicy

```
Get-DnsClientNrptPolicy
```

Pokazuje reguły tabeli NRPT aktywne na komputerze, w tym te dołożone przez `DomainNameInformation` z profilu VPN. Sprawdzamy nią, czy zapytania o `.ad.zielona.example` rzeczywiście trafiają do wewnętrznych serwerów DNS, a nie do zwykłego DNS z sieci hotelowej.

Przykład:

```
Namespace             : .ad.zielona.example
NameServers           : {10.20.10.11, 10.20.10.12}
```

NRPT = Name Resolution Policy Table: tabela reguł Windows mówiąca, dla której domeny pytać o adres inny serwer DNS niż domyślny · **VPN** = Virtual Private Network: szyfrowany tunel do sieci firmowej · **DNS** = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP

Z życia wzięte

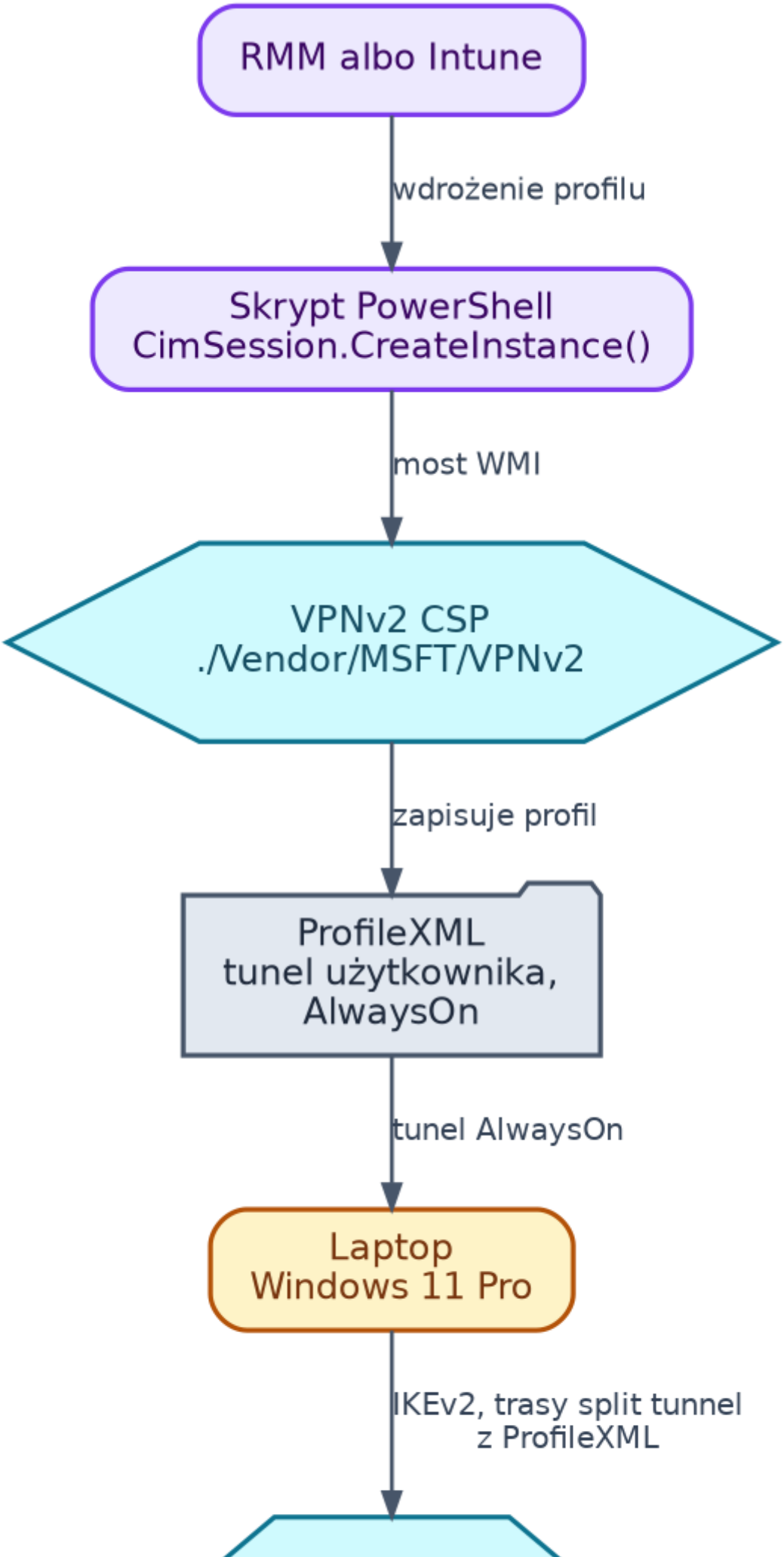
Z ŻYCIA WZIĘTE

Tomek dostaje zadanie: rozesłać profil AlwaysOn na trzydzieści laptopów handlowców przez RMM. Pierwsza wersja skryptu korzysta z `New-CimInstance` i na większości maszyn działa, ale na kilku kończy się błędem `MI RESULT 1` bez dalszego wyjaśnienia. Dopiero po przepisaniu zapisu profilu na `CimSession.CreateInstance()`, jak w fiszce wyżej, skrypt przechodzi identycznie na wszystkich maszynach.

Jak to działa

Rozwiń: jak to działa pod spodem

> **Pojęcie: tunel urządzenia i tunel użytkownika.** Tunel urządzenia (device tunnel) łączy się jeszcze przed zalogowaniem, w kontekście samego komputera, i uwierzytelnia się wyłącznie certyfikatem maszyny. Tunel użytkownika (user tunnel) łączy się dopiero po zalogowaniu konkretnej osoby i może uwierzytelniać się jej hasłem, certyfikatem użytkownika albo metodą EAP.


RMM albo Intune
wdrożenie profilu
Skrypt PowerShell
CimSession.CreateInstance()
most WMI
VPNv2 CSP
./Vendor/MSFT/VPNv2
zapisuje profil
ProfileXML
tunel użytkownika,
AlwaysOn
tunel AlwaysOn
Laptop
Windows 11 Pro
IKEv2, trasy split tunnel
z ProfileXML

Droga wdrożenia profilu AlwaysOn: od RMM albo Intune, przez skrypt WMI, do CSP VPNv2 i tunelu na laptopie (kliknięcie otwiera pełny rozmiar)

> ⚠ Tunel urządzenia wymaga Windows 11 Enterprise albo Education. Windows 11 Pro, którego używa większość floty laptopów Zielonej, sam z siebie nie uruchomi tunelu urządzenia, nawet jeśli profil zostanie na niego poprawnie wgrany. Próby obejścia tego przez PLAP (Pre-Logon Access Provider) z certyfikatem maszyny bywają na Windows 11 zawodne, dlatego Zielona stawia wyłącznie na tunel użytkownika.

Pojęcie: CSP VPNv2. Configuration Service Provider, przez który system MDM albo mechanizm WMI zapisuje w Windows kompletny profil VPN jednym blokiem XML zamiast osobnych, ręcznie klikanych ustawień. Profil trafia do drzewa `./Vendor/MSFT/VPNv2/<NazwaProfilu>/...` i stamtąd Windows odczytuje wszystkie parametry tunelu.

W samym pliku ProfileXML kolejność elementów najwyższego poziomu ma znaczenie: `DnsSuffix`, potem `NativeProfile`, potem `AlwaysOn`, potem `RememberCredentials`, potem `Route`, na końcu `DomainNameInformation`. Zapis w innej kolejności bywa po cichu odrzucany przez CSP, bez czytelnego komunikatu błędu, co jest najbardziej mylącą częścią całej procedury.

> Zanim wyślemy plik ProfileXML do skryptu wdrożeniowego, sprawdzamy jego poprawność składniową lokalnie, na przykład jednym poleceniem `python3 -c "import xml.etree.ElementTree as ET; ET.parse('profil.xml')"`. Błąd składni w tym miejscu oznacza błąd w każdym profilu wysłanym dalej, więc taniej złapać go przed wdrożeniem niż po nim.

Sekcja `Route` w profilu wyżej definiuje trzy sieci Zielonej: serwery (`10.20.10.0/24`), biuro w Pamiątkowie (`10.20.20.0/24`) i magazyn w Swarzędzu (`10.20.30.0/24`). Te trasy muszą być zdefiniowane po stronie klienta, we właśnie tym miejscu profilu, a nie tylko na `fw01`. Wbudowany klient IKEv2 Windows w praktyce ignoruje trasy split-include przysyłane przez FortiGate podczas negocjacji fazy 2, więc bez sekcji `Route` w ProfileXML cały ruch klienta zostałby poprowadzony poza tunel, mimo że brama próbowała ustawić inaczej.

Sekcja `DomainNameInformation` odpowiada za regułę NRPT: dla domeny `.ad.zielona.example` Windows ma pytać wyłącznie serwery `10.20.10.11` i `10.20.10.12`, nie domyślny DNS z sieci, w której akurat znajduje się laptop. Bez tej sekcji nazwy wewnętrzne Zielonej, na przykład `sql01.ad.zielona.example`, nie rozwiążą się poprawnie z hotelowego Wi-Fi, nawet gdy sam tunel

VPN już stoi.

Profil wdrażamy skryptem z fiszki przez RMM albo przez Intune jako niestandardowy profil VPN typu ProfileXML: obie drogi w tle robią to samo, zapisują ten sam blok XML do tego samego węzła CSP, różni się tylko mechanizm dostarczenia skryptu na docelowy laptop.

Częste błędy

Objaw	Przyczyna	Co robimy
`New-CimInstance` kończy się błędem `MI RESULT 1` bez dalszego opisu	cmdlet `New-CimInstance` bywa zawodny przy zapisie do przestrzeni nazw `root\cimv2\mdm\dmmap` używanej przez CSP	zapisujemy profil metodą `$session.CreateInstance()` z klasy `CimSession`, jak w fiszce wyżej, zamiast cmdletem
Profil wdrożony poprawnie, ale cały ruch klienta, także do sieci Zielonej, omija tunel	brak albo pusta sekcja `Route` w ProfileXML, klient zignorował trasy split-include z `fw01`	dopisujemy jawne wpisy `Route` dla każdej z trzech sieci Zielonej po stronie klienta
Tunel się łączy, ale nazwy wewnętrzne, na przykład `sql01.ad.zielona.example`, nie rozwiązują się	brak sekcji `DomainNameInformation`, więc nie powstała reguła NRPT dla domeny wewnętrznej	dopisujemy `DomainNameInformation` ze wskazaniem `10.20.10.11` i `10.20.10.12` jako serwerów DNS
Profil w ogóle się nie zapisuje albo znika po restarcie, bez błędu w konsoli	elementy ProfileXML w złej kolejności, CSP po cichu odrzuca cały blok	poprawiamy kolejność na `DnsSuffix`, `NativeProfile`, `AlwaysOn`, `RememberCredentials`, `Route`, `DomainNameInformation`
Tunel urządzenia w ogóle się nie łączy na laptopach z Windows 11 Pro	Windows 11 Pro nie obsługuje tunelu urządzenia, wymaga Enterprise albo Education	zostajemy przy tunelu użytkownika dla tej floty, tunel urządzenia planujemy dopiero po migracji licencji

□□ Ćwiczenia

1. Skrypt wdrożeniowy zapisuje profil komendą `New-CimInstance` i działa na testowym laptopie administratora, ale na laptopach handlowców kończy się błędem `MI RESULT 1`. Co zmieniamy w pierwszej kolejności?

Odpowiedź

Zamieniamy zapis profilu z cmdletu `New-CimInstance` na metodę `$session.CreateInstance()` obiektu `CimSession`, otwartego z odpowiednimi opcjami kontekstu użytkownika. To dokładnie ten sam most WMI do CSP, ale inna droga zapisu, która w praktyce działa tam, gdzie `New-`

`CimInstance` zawodzi.

1. Profil ProfileXML ma sekcję `Route` z trzema sieciami Zielonej, ale ktoś przypadkiem wkleił ją przed sekcją `NativeProfile`. Co się stanie po wdrożeniu takiego profilu?

Odpowiedź

CSP VPNv2 wymaga konkretnej kolejności elementów najwyższego poziomu w ProfileXML. Profil ze zmienioną kolejnością bywa po cichu odrzucony w całości, bez wyraźnego komunikatu błędu, więc zamiast błędnie działającego tunelu najczęściej nie zobaczymy żadnego profilu wcale.

Źródła: Tutorial: Create Always On VPN Connection on Windows Client Devices, Microsoft Learn · VPNv2 CSP, Microsoft Learn · VPNv2 DDF file, Microsoft Learn · Configure the VPN device tunnel in Windows client, Microsoft Learn

---

Revision #27
Created 2026-09-28 06:15:12 UTC by Tomasz Konieczny
Updated 2026-10-26 14:18:11 UTC by Tomasz Konieczny