

Wydanie II

Helion 

Windows Server 2019 dla profesjonalistów

Tak stworzysz najnowocześniejsze centrum obliczeniowe!



Packt 

Jordan Krause

Tytuł oryginału: Mastering Windows Server 2019: The complete guide for IT professionals to install and manage Windows Server 2019 and deploy new capabilities, 2nd Edition

Tłumaczenie: Jacek Janusz

ISBN: 978-83-283-6485-1

Copyright © Packt Publishing 2019.

First published in the English language under the title 'Mastering Windows Server 2019 - Second Edition – (9781789804539)'

Polish edition copyright © 2020 by Helion SA

All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/ws19pr>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorze	11
O recenzentach	12
Przedmowa	13
Rozdział 1. Pierwsze kroki w systemie Windows Server 2019	19
Cel systemu Windows Server	20
Robi się pochmurno	22
Chmura publiczna	22
Chmura prywatna	23
Wersje systemu Windows Server i jego licencjonowanie	24
Wersje Standard i Datacenter	24
Desktop Experience, Server Core, Nano Server	25
Modele licencjonowania — SAC i LTSC	26
Przegląd nowych i zaktualizowanych funkcji	28
System Windows 10 wciąż działa	28
Infrastruktura hiperkonwergentna	28
Windows Admin Center	29
Windows Defender Advanced Threat Protection	29
Hasła zabronione	29
Miękki restart	30
Integracja z Linuksem	30
Ulepszona funkcja Shielded Virtual Machines	31
Azure Network Adapter	31
Always On VPN	32
Poruszanie się po interfejsie	32
Uaktualnione menu Start	33
Menu szybkich zadań administracyjnych	34

Używanie funkcji wyszukiwania	36
Przypinanie programów do paska zadań	37
Korzystanie z nowszego ekranu Settings	40
Dwie metody wykonania tej samej czynności	43
Menedżer zadań	46
Widok zadań	49
Podsumowanie	51
Pytania	52
Rozdział 2. Instalowanie systemu Windows Server 2019 i zarządzanie nim	53
Wymagania dotyczące instalacji	54
Instalowanie systemu Windows Server 2019	54
Wypalanie pliku ISO	55
Tworzenie rozruchowej pamięci USB (pendrive)	56
Uruchamianie instalatora	57
Instalowanie ról i funkcji	61
Instalowanie roli za pomocą kreatora	62
Instalowanie funkcji przy użyciu powłoki PowerShell	67
Scentralizowane zarządzanie i monitorowanie	69
Menedżer serwera	69
Narzędzia administracji zdalnej serwera (RSAT)	74
Czy to oznacza, że RDP jest martwy?	75
Windows Admin Center (WAC)	76
Instalacja Windows Admin Center	77
Uruchamianie Windows Admin Center	78
Dodawanie większej liczby serwerów do Windows Admin Center	80
Zarządzanie serwerem przy użyciu Windows Admin Center	81
Umożliwienie szybkiego wdrożenia serwera za pomocą narzędzia Sysprep	82
Instalacja systemu Windows Server 2019 na nowym serwerze	83
Konfigurowanie ustawień i aktualizacji na nowo utworzonym serwerze	83
Uruchomienie narzędzia Sysprep, aby przygotować i wyłączyć serwer główny	84
Tworzenie wzorcowego obrazu dysku	87
Wdrażanie nowych serwerów przy użyciu kopii obrazu wzorcowego	87
Podsumowanie	88
Pytania	89
Rozdział 3. Podstawowe usługi infrastrukturalne	91
Co to jest kontroler domeny?	92
Active Directory Domain Services	92
Używanie roli AD DS do zarządzania siecią	93
Active Directory Users and Computers	94
Active Directory Domains and Trusts	100
Active Directory Sites and Services	101
Active Directory Administrative Center	102
Kontrolery domeny tylko do odczytu (RODC)	104
Potęga zasad grupy	105
Domyślne zasady domeny	106
Tworzenie i podłączanie nowego obiektu zasad grupy	108
Filtrowanie obiektów zasad grupy z uwzględnieniem określonych urządzeń	111

System nazw domen (DNS)	112
Różne rodzaje rekordów DNS	114
DHCP a adresowanie statyczne	119
Zakres DHCP	120
Zastrzeżenia DHCP	122
Kopia zapasowa i jej przywracanie	124
Planowanie wykonywania regularnych kopii zapasowych	124
Przywracanie danych z systemu Windows	128
Przywracanie z płyty instalacyjnej	129
Skróty MMC i MSC	133
Podsumowanie	136
Pytania	136
Rozdział 4. Certyfikaty w systemie Windows Server 2019	137
Ogólnie używane typy certyfikatów	138
Certyfikaty użytkownika	138
Certyfikaty komputera	139
Certyfikaty SSL	139
Planowanie środowiska PKI	143
Usługi roli AD CS	143
Urząd certyfikacji przedsiębiorstwa czy autonomiczny?	144
Główny czy podrzędny urząd certyfikacji?	146
Nazwa serwera urzędu certyfikacji	147
Czy mogę zainstalować rolę CA na kontrolerze domeny?	147
Tworzenie nowego szablonu certyfikatu	148
Wydawanie nowych certyfikatów	152
Publikowanie szablonu	152
Żądanie wydania certyfikatu przy użyciu konsoli MMC	154
Żądanie wydania certyfikatu przy użyciu interfejsu WWW	156
Określanie sposobu automatycznej rejestracji certyfikatów	159
Uzyskanie certyfikatu SSL organu publicznego	164
Para kluczy publiczny-prywatny	164
Tworzenie żądania podpisania certyfikatu	165
Przesyłanie żądania certyfikatu	167
Pobieranie i instalowanie certyfikatu	168
Eksportowanie i importowanie certyfikatów	170
Eksportowanie z przystawki MMC	170
Eksportowanie z konsoli IIS	171
Importowanie w innym serwerze	172
Podsumowanie	172
Pytania	173
Rozdział 5. Obsługa sieci w Windows Server 2019	175
Wprowadzenie do protokołu IPv6	176
Jak działają adresy IP w wersji IPv6?	177
Twoje narzędzia sieciowe	181
Polecenie ping	181
Polecenie tracert	182
Polecenie pathping	184

Polecenie Test-Connection	185
Polecenie telnet	187
Polecenie Test-NetConnection	189
Śledzenie pakietów za pomocą programów Wireshark lub Message Analyzer	190
Narzędzie TCPView	191
Tworzenie tablicy routingu	192
Serwery o wielu adresach	192
Tylko jedna brama domyślna	193
Definiowanie trasy	194
Grupowanie kart sieciowych	198
Programowalna sieć komputerowa	201
Wirtualizacja sieci Hyper-V	202
Łączenie sieci lokalnej z usługą Azure	207
Azure Network Adapter	207
Podsumowanie	209
Pytania	209
 Rozdział 6. Użycie opcji zdalnego dostępu	 211
Always On VPN	212
Rodzaje tuneli AOVPN	213
Wymagania niezbędne do uruchomienia tunelu urządzenia	214
Wymagania klienta AOVPN	214
Wdrażanie ustawień	215
Serwerowe komponenty AOVPN	217
DirectAccess	219
Cała prawda o usłudze DirectAccess i protokole IPv6	220
Wymagania wstępne dotyczące usługi DirectAccess	222
Nie używaj kreatora Getting Started Wizard (GSW)!	230
Remote Access Management Console	231
Configuration	232
Dashboard	233
Operations Status	233
Remote Client Status	234
Reporting	235
Tasks	236
DirectAccess, VPN czy AOVPN? Jakie rozwiązanie jest najlepsze?	237
Dołączenie do domeny?	237
Uruchamianie automatyczne czy ręczne?	238
Oprogramowanie zewnętrzne czy wbudowane?	238
Problemy z hasłem i logowaniem w tradycyjnych sieciach VPN	239
Zapory z ograniczeniami portów	240
Ręczne rozłączanie	241
Natywne funkcje równoważenia obciążenia	242
Dystrybucja konfiguracji klienta	243
Web Application Proxy (WAP)	244
WAP jako serwer proxy AD FS	245
Wymagania dla WAP	245
Najnowsze ulepszenia WAP	246
Uwierzytelnienie wstępne dla autoryzacji HTTP Basic	246
Przekierowanie HTTP na HTTPS	246

Adresy IP klientów przekazywane do aplikacji	247
Dostęp do serwera Remote Desktop Gateway	247
Ulepszona konsola administracyjna	247
Podsumowanie	249
Pytania	249
Rozdział 7. Hardening i bezpieczeństwo	251
Windows Defender Advanced Threat Protection	252
Instalacja programu Windows Defender AV	253
Wykorzystanie interfejsu użytkownika	253
Wyłączanie usługi Windows Defender	254
Czym w ogóle jest ATP?	256
Windows Defender ATP Exploit Guard	257
Zapora systemu Windows Defender — bez żartów	258
Trzy konsole administracyjne zapory systemu Windows	259
Trzy różne profile zapory	262
Tworzenie w zaporze nowej reguły przychodzącej	263
Tworzenie reguły zezwalającej na wysyłanie pingów (ICMP)	266
Zarządzanie zaporą WFAS przy użyciu zasad grupy	269
Technologie szyfrowania	272
BitLocker i wirtualny układ TPM	272
Chronione maszyny wirtualne	273
Szyfrowane sieci wirtualne	274
Encrypting File System	274
Protokoły IPsec	275
Hasła zabronione	278
Zaawansowana analiza zagrożeń	279
Najważniejsze wskazówki dotyczące ogólnego bezpieczeństwa	282
Pozbycie się wiecznych administratorów	282
Korzystanie z odrębnych kont w celu uzyskania dostępu administracyjnego	283
Używanie innego komputera do wykonywania zadań administracyjnych	283
Nigdy nie przeglądaj internetu, będąc zalogowanym na serwerze	284
Kontrola dostępu oparta na rolach	284
Just Enough Administration	285
Podsumowanie	285
Pytania	286
Rozdział 8. Server Core	287
Dlaczego warto korzystać z wersji Server Core?	288
Zmiana wersji w locie jest już niemożliwa	289
Używanie systemu Server Core	290
PowerShell	291
Zdalna sesja PowerShell	296
Menedżer serwera	298
Narzędzia administracji zdalnej serwera	298
Przypadkowe zamknięcie okna z wierszem poleceń	300
Wykorzystanie aplikacji Windows Admin Center do zarządzania systemem Server Core	302
Narzędzie Sconfig	305
Role dostępne w wersji Server Core	309

Co się stało z systemem Nano Server?	309
Podsumowanie	310
Pytania	311
Rozdział 9. Redundancja w systemie Windows Server 2019	313
Równoważenie obciążenia sieciowego	314
Coś innego niż usługa DNS typu round-robin	315
Jakie role mogą korzystać z równoważenia obciążenia sieciowego?	315
Adresy IP wirtualne i dedykowane	316
Tryby pracy NLB	317
Konfigurowanie strony WWW z równoważeniem obciążenia	319
Włączanie opcji NLB	320
Konfigurowanie opcji NLB	321
Konfigurowanie usług IIS i DNS	325
Testowanie rozwiązania	327
Opróżnianie pamięci podręcznej ARP	328
Klaster pracy awaryjnej	329
Klastrowanie hostów Hyper-V	329
Klasy dla usług plikowych	330
Poziomy klastrowania	331
Klastrowanie na poziomie aplikacji	331
Klastrowanie na poziomie serwera	332
Połączenie obu poziomów klastrowania	332
Jak działa tryb pracy awaryjnej?	332
Konfigurowanie klastra pracy awaryjnej	333
Konfigurowanie serwerów	334
Instalowanie funkcji	335
Uruchamianie menedżera klastra pracy awaryjnej	335
Uruchamianie sprawdzania poprawności klastra	336
Uruchamianie kreatora tworzenia klastra	338
Najnowsze ulepszenia dotyczące klastrowania w systemie Windows Server	339
Prawdziwe dwuwęzłowe klastry ze świadkami wykorzystującymi medium USB	339
Wyższe bezpieczeństwo klastrów	340
Klastry korzystające z wielu lokalizacji	340
Klastry w wielu domenach lub grupie roboczej	340
Uaktualnienia stopniowe systemu operacyjnego klastra	341
Odporność maszyn wirtualnych	342
Storage Replica	342
Bezpośrednie miejsce do magazynowania	343
Nowości w systemie Windows Server 2019	345
Podsumowanie	345
Pytania	346
Rozdział 10. PowerShell	347
Dlaczego warto używać interfejsu PowerShell?	347
Polecenia cmdlet	348
PowerShell jest podstawą	349
Skrypty	350
Server Core	350

Praca z programem PowerShell	351
Uruchamianie środowiska PowerShell	351
Użycie klawisza Tab	356
Przydatne polecenia cmdlet używane do codziennych zadań	357
Użycie polecenia Get-Help	359
Formatowanie danych wyjściowych	360
Zintegrowane środowisko skryptowe PowerShell	363
Pliki PS1	364
Zintegrowane środowisko skryptowe PowerShell	365
Zdalne zarządzanie serwerem	368
Przygotowanie zdalnego serwera	369
Łączenie ze zdalnym serwerem	371
Konfiguracja żadanego stanu	375
Podsumowanie	377
Pytania	377
 Rozdział 11. Kontenery i Nano Server	 379
Co to są kontenery aplikacji?	380
Współdzielenie zasobów	380
Izolowanie	381
Skalowalność	382
Kontenery i Nano Server	383
Kontenery Windows Server a kontenery Hyper-V	384
Kontenery Windows Server	384
Kontenery Hyper-V	385
Docker i Kubernetes	385
Kontenery Linux	386
Docker Hub	386
Docker Trusted Registry	387
Kubernetes	388
Używanie kontenerów	389
Instalowanie roli i funkcji	389
Instalacja środowiska Docker for Windows	390
Pobieranie obrazu kontenera	392
Uruchamianie kontenera	393
Podsumowanie	395
Pytania	395
 Rozdział 12. Wirtualizacja centrum danych za pomocą hiperwizora Hyper-V	 397
Projektowanie i wdrażanie serwera Hyper-V	398
Instalowanie roli Hyper-V	399
Użycie przełączników wirtualnych	402
Zewnętrzny przełącznik wirtualny	404
Wewnętrzny przełącznik wirtualny	405
Prywatny przełącznik wirtualny	405
Tworzenie nowego przełącznika wirtualnego	405
Implementacja serwera wirtualnego	406
Uruchamianie maszyny wirtualnej i łączenie się z nią	410
Instalowanie systemu operacyjnego	411

Zarządzanie serwerem wirtualnym	412
Menedżer funkcji Hyper-V	413
Opcja Settings	415
Konsola Hyper-V, protokół pulpitu zdalnego (RDP) czy PowerShell	420
Windows Admin Center (WAC)	421
Chronione maszyny wirtualne	421
Szyfrowanie dysków VHD	424
Wymagania dotyczące infrastruktury dla chronionych maszyn wirtualnych	425
Poświadczenia hosta	426
Integracja z systemem Linux	427
Deduplikacja w systemie Resilient File System (ReFS)	428
System plików ReFS	428
Deduplikacja danych	429
Dlaczego jest to ważne dla środowiska Hyper-V?	429
Środowisko Hyper-V Server 2019	429
Podsumowanie	432
Pytania	433
Odpowiedzi na pytania	435

Obsługa sieci w Windows Server 2019

Jak wynika z informacji przedstawionych dotąd w tej książce, serwery są czymś w rodzaju pni drzew w naszych sieciach. Stanowią one infrastrukturę szkieletową, która umożliwia wykonywanie naszej pracy. Jeśli serwery są pniami, wówczas same sieci muszą być korzeniami. Twoja sieć to platforma obsługująca infrastrukturę firmy. Tworzy kanały, z których korzystają wszystkie urządzenia w firmie, aby się ze sobą komunikować.

Tradycyjnie w branży IT istnieli *specjaliści od serwerów* oraz *specjaliści od sieci*, a w wielu firmach nadal tak jest. Administrator, który przede wszystkim obsługuje serwery, zazwyczaj nie ma w ciągu dnia wystarczającej ilości czasu, aby zajmować się również infrastrukturą sieciową. Odwrotna sytuacja także jest prawdziwa. Administratorzy sieci do zarządzania używają zwykle specyficznego sprzętu i specyficznych narzędzi, więc nie byłoby zainteresowani zbyt głębokim zanurzeniem się w świat systemu Windows Server. Jednak wielu z nas pracuje w mniejszych firmach, w których trzeba się zajmować różnymi zagadnieniami. Czasami administratorzy serwerów i sieci muszą ściśle ze sobą współpracować, dlatego powinniśmy zrozumieć przynajmniej podstawowe zasady działania sieci i poznać narzędzia, których możemy użyć do rozwiązywania problemów z niedziałającymi połączeniami. Ponadto system Windows Server 2019 zmusza nas do innego sposobu myślenia o sieci — chodzi o wirtualizację. Zawsze będzie istnieć warstwa fizyczna sieci, wykorzystująca realne przełączniki i routery do przekazywania pakietów między różnymi pokojami i budynkami. Jednak w serwerach Windows Server pojawiła się możliwość wykorzystania *programowalnej sieci komputerowej* (ang. *software-defined networking* — SDN), co pozwala na zwirtualizowanie niektórych konfiguracji sieciowych. Oprócz tego wirtualizujemy ruch sieciowy i budujemy sieci przy użyciu konsoli serwera, zamiast, jak w przeszłości, używać interfejsu wiersza poleceń w celu podłączania się do routerów.

Stop, chyba się zagalopowałem. Najpierw porozmawiajmy o nowych przydatnych składnikach w systemie Windows Server 2019, które współpracują z fizycznymi lub całkowicie dowolnymi

sieciami komputerowymi, ponieważ będą one ważne dla każdego administratora. Później poświęcimy kilka chwil na bliższe zapoznanie Cię z wirtualizacją sieci.

Oto zagadnienia, które omówimy w tym rozdziale:

- Wprowadzenie do protokołu IPv6.
- Twoje narzędzia sieciowe.
- Tworzenie tablicy routingu.
- Grupowanie kart sieciowych.
- Programowalna sieć komputerowa.

Wprowadzenie do protokołu IPv6

Witamy po mrocznej stronie mocy! Niestety, tak właśnie wielu administratorów myśli o protokole IPv6. Chociaż protokół IPv6 nie jest niczym nowym, z mojego doświadczenia wynika, że prawie nikt nie wdrożył go w sieci lokalnej. Współpracując przez ostatnich kilka lat z setkami różnych firm na całym świecie, natknąłem się tylko na jedną organizację, która korzystała z protokołu IPv6 w całej sieci produkcyjnej, i nawet w tym przypadku nie była to jego prawdziwa, natywna wersja. Zamiast tego firma korzystała z technologii tunelowania zwanej ISATAP, aby wszystkie serwery i klienci porozumiewały się ze sobą za pomocą pakietów IPv6, które jednak wciąż przemieszczały się po fizycznej sieci IPv4. Nie zrozum mnie źle — znam wiele przypadków, w których firmy testują protokół IPv6 i nawet mają w tym celu odpowiednio skonfigurowane odizolowane fragmenty sieci. Czy jednak używasz protokołu IPv6 w całej sieci produkcyjnej? Większość z nas nie jest jeszcze gotowa na tak wielką zmianę. Dlaczego tak trudno jest wdrożyć protokół IPv6? Ponieważ od samego początku używamy protokołu IPv4, znamy go i rozumiemy, a poza tym naprawdę nie ma potrzeby przechodzenia w naszych sieciach na wersję IPv6. Chwila — wydawało mi się, że pojawiły się obawy przed wyczerpaniem adresów IPv4... Tak, dotyczy to adresów IP w publicznym internecie, ale nie ma nic wspólnego z naszymi sieciami wewnętrznymi. Otóż nawet jeśli jutro zabraknie nam publicznych adresów IPv4, nie wpłynie to wcale na sieci wewnętrzne w naszych firmach. Przez długi czas (a być może bezterminowo) będziemy mogli nadal używać protokołu IPv4 w swoich sieciach, o ile będziemy się czuli komfortowo, korzystając z technologii NAT do tłumaczenia zewnętrznego ruchu sieciowego na protokół IPv4. Wszyscy używamy translacji NAT w takiej czy innej formie prawie tak długo, jak istnieje protokół IPv4, więc jest to coś, z czego ludzie są bardzo zadowoleni.

Powiem wprost: nie próbuję Cię przekonywać, że trzymanie się protokołu IPv4 to sposób na przyszłość. Informuję tylko, że dla większości organizacji w ciągu najbliższych kilku lat będzie to po prostu prawda. Powodem, dla którego w tej książce chciałbym omówić protokół IPv6, jest to, że ostatecznie będziesz musiał sobie z nim poradzić. Ale kiedy to zrobisz, będziesz naprawdę podekscytowany! Protokół IPv6 ma kilka olbrzymich zalet w porównaniu z IPv4, w tym ogromną liczbę adresów IP, które można przechowywać w ramach jednej sieci. Zespoły administratorów sieciowych w firmach na całym świecie zmagają się codziennie z potrzebą budowania coraz większej liczby sieci IPv4 i ich łączenia. Pomyśl o tym: istnieje mnóstwo firm, w których liczba pracowników przekracza 10 tysięcy. W niektórych jest ich nawet o wiele

■ Adres IPv4 to adres o długości 32 bitów, który wygląda tak:

■ Adres IPv6 to adres o długości 128 bitów, który wygląda tak:

Jak widać, adres IPv4 jest znacznie krótszy, co oczywiście oznacza, że istnieje mniej możliwości uzyskania unikatowych adresów IP. W rzeczywistości nie widzisz, jak długie są adresy IPv6. W powyższych przykładach przedstawiono adresy IPv4 i IPv6 w takiej postaci, jaka została przyjęta do ich wyświetlania. Naprawdę jednak adres IPv4 jest prezentowany w postaci dziesiętnej, a IPv6 w systemie szesnastkowym. Adresy IPv6 są wyświetlane przy użyciu systemu szesnastkowego, dzięki czemu są maksymalnie skompresowane. Jeśli zajrzysz pod maskę, adres IPv6 w swojej natywnej 128-bitowej formie może wyglądać mniej więcej tak, jak pokazano poniżej (i właśnie tak wygląda wewnątrz pakietu danych):

To imponujący zestaw cyfr, ale nie jest on czymś bardzo użytecznym lub przyjaznym dla ludzkiego oka. Może więc zamiast w systemie dwójkowym zaprezentowalibyśmy adres IPv6 w formacie dziesiętnym — w taki sam sposób, w jaki są pokazywane adresy IPv4.⁹ W takim przypadku adres IPv6 będzie wyglądać mniej więcej tak:

Teraz w pełni rozumiesz, dlaczego adres IPv6 jest zawsze używany i wyświetlany w systemie szesnastkowym — jest on wystarczająco długi nawet w tym skompresowanym formacie!

Podczas tworzenia sieci w wersji IPv4 jej podsieci są niezwykle ważne, ponieważ umożliwiają nam posiadanie więcej niż jednego zbioru adresów IP. W przypadku najbardziej podstawowej formy sieci, w której konfigurujesz adresy i używasz 24-bitowej podsieci (maska podsieci równa 255.255.255.0), co jest bardzo powszechne na niewielkich obszarach, takich jak dom lub małe biuro, jesteś ograniczony do 254 unikatowych adresów IP. Ojej! Niektóre firmy mają tysiące różnych serwerów, a oprócz tego komputery klienckie i inne urządzenia, które również muszą się łączyć z siecią. Na szczęście możemy zbudować wiele różnych podsieci w ramach

jednej sieci IPv4 w celu zwiększenia swojego użytecznego zakresu adresów IP, ale wymaga to starannego planowania i obliczania przestrzeni adresowych. Jest to powód, dla którego polegamy na doświadczonych administratorach sieci, którzy zarządzają tym fragmentem infrastruktury. Jedna niepoprawna konfiguracja podsieci w tablicy routingu może zablokować cały ruch sieciowy. Administrowanie podsieciami w dużej sieci IPv4 nie jest zadaniem dla osób o słabym sercu.

Gdy myślimy o adresowaniu IPv6, możliwości są niemalże nieograniczone. Jeśli wyliczyłbyś wszystkie unikatowe adresy IPv6 dostępne w 128-bitowej przestrzeni, odkryłbyś, że istnieje ponad 340 undecylinów adresów możliwych do utworzenia. Innymi słowy, byłoby to 340 trylionów trylionów trylionów adresów. Jest to liczba dostępnych adresów w internecie obsługującym protokół IPv6, ale co to oznacza dla naszych sieci wewnętrznych?

Aby przeanalizować liczbę adresów, których moglibyśmy używać w typowej sieci wewnętrznej w wersji IPv6, przyjrzyjmy się jeszcze raz wcześniej zaprezentowanemu adresowi. Wymyśliłem go na poczekaniu, ale postaramy się go podzielić na części:

2001:AABB:CCDD:AB00:0123:4567:8901:ABCD

W porównaniu z prostym adresem 192.168.1.5 powyższy ciąg znaków wygląda jak potwór. Wynika to z tego, że generalnie nie jesteśmy przyzwyczajeni do obsługi formatu szesnastkowego — to po prostu inny sposób patrzenia na dane. Jak wspomniałem, jest to adres 128-bitowy. Został on podzielony na 8 różnych sekcji, a każda z nich jest oddzielona dwukropkiem i składa się z 16 bitów. Pierwsze 64 bity (pierwsza połowa) adresu to informacje o routingu, a ostatnie 64 bity to unikatowy identyfikator urządzenia w sieci. W pierwszej części mamy dwa różne komponenty. Pierwszych 48 bitów (3 grupy szesnastkowe) to prefiks organizacyjny, który będzie taki sam dla wszystkich naszych urządzeń w sieci. Następnie czwarty zestaw informacji, czyli dalszych 16 bitów, może być naszym identyfikatorem podsieci. Daje nam to możliwość posiadania w przyszłości wielu różnych podsieci. Po użyciu pierwszej połowy adresu pozostaje nam jeszcze druga część, czyli ostatnie 64 bity, które możemy wykorzystać do nadawania identyfikatorów urządzeniom. Ta część adresu będzie odmienna dla każdego urządzenia w sieci i pozwoli na nadawanie indywidualnych, statycznych adresów IPv6, które będą używane do komunikacji. Pokażę to wszystko dokładnie. Weźmy poprzedni przykładowy adres i podzielmy go na następujące części:

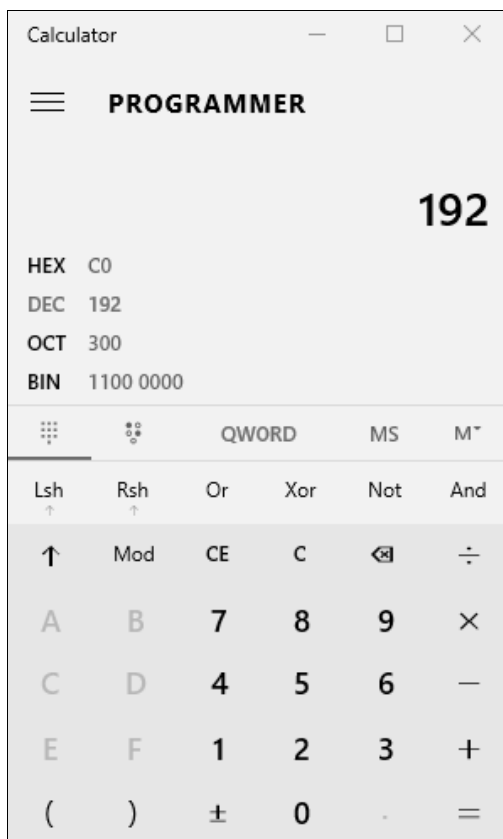
- **prefiks organizacyjny:** 2001:AABB:CCDD,
- **identyfikator podsieci:** AB00,
- **identyfikatory urządzeń:** 0123:4567:8901:ABCD.

Ile urządzeń możemy mieć w swojej sieci ze schematem IP takim jak powyższy? Cóż, nawet w naszym przykładzie, w którym przydzieliliśmy tylko jedną 16-bitową sekcję dla podsieci i 64 bity dla rzeczywistych adresów IP, uzyskalibyśmy możliwość posiadania ponad 65 000 podsieci i kwintylionów unikatowych identyfikatorów urządzeń w naszym zakresie IP. Imponujące, prawda?

Jeśli zastosujemy powyższy schemat i użyjemy tylko jednej podsieci w celu zarządzania wszystkimi urządzeniami, pierwsza połowa naszych adresów będzie zawsze taka sama, dzięki czemu będą one łatwiejsze do zapamiętania i obsługi. Zaskakujące jest to, że szybko przyzwyczajasz się w swoim środowisku do wyświetlania tych dużych liczb szesnastkowych. Nawet jeśli za-

czniesz je rozpoznawać, prawdopodobnie nadal nie będziesz chciał od razu logować się na serwery lub komputery w sieci przy użyciu ich statycznych adresów IP. Wiem, że wciąż wielu z nas ma w zwyczaju mówić: „Muszę szybko wskoczyć na mój serwer WWW, więc połączę się z adresem 192.168.1.5 przez zdalny pulpit”. Sam czas potrzebny na wpisanie adresów IPv6, nawet jeśli je pamiętasz, na ogół nie jest tego wart. Wdrożenie protokołu IPv6 pociągnie za sobą większe uzależnienie od usług DHCP i DNS, co sprawi, że stanie się on bardziej użyteczny.

Gdy już wiemy, do jakich celów są wykorzystywane określone fragmenty adresu, musimy się zastanowić, w jaki sposób moglibyśmy przypisać numery identyfikacyjne wszystkim komputerom, serwerom i innym urządzeniom w swojej sieci. Mógłbyś zacząć od cyfry 1 i po prostu za każdym razem zwiększać liczbę o jeden. Innym pomysłem jest zamiana starych adresów IPv4 na wartości szesnastkowe i użycie ich jako ostatnich 32 bitów adresu. Na swoim komputerze otwórz aplikację kalkulatora, a następnie przejdź w tryb programisty. Dzięki temu otrzymasz szybkie i łatwe w użyciu narzędzie, które można wykorzystać do konwersji liczb dziesiętnych na szesnastkowe i odwrotnie. Weźmy przykład mojego serwera WWW, który działa pod adresem 192.168.1.5. W sieci zamierzam wdrożyć protokół IPv6 i dlatego chciałbym, aby adres IPv6 mojego serwera odzwierciedlał w sekcji identyfikatora urządzenia oryginalny adres IPv4. Jeśli w kalkulatorze wprowadzisz wartość 192, wyświetli się odpowiednia liczba szesnastkowa, jak pokazano na poniższym zrzucie ekranu:



Jeśli postąpimy tak samo z każdym oktetem naszego adresu IPv4, otrzymamy następujące wyniki:

```
192 = C0
168 = A8
1  = 01
5  = 05
```

Zatem adresowi 192.168.1.5 odpowiada wartość C0A8:0105. Teraz mogę jej użyć w połączeniu z prefiksem organizacyjnym i identyfikatorem podsieci, aby utworzyć statyczny adres IPv6 dla swojego serwera WWW:

```
2001:AABB:CCDD:0001:0000:0000:C0A8:0105
```

Być może zauważyłeś, że na końcu adresu IPv6 umieściłem szesnastkowy identyfikator urządzenia, a oprócz tego wprowadziłem kilka innych zmian. Założyliśmy, że ostatnie 64 bity będą przeznaczone dla identyfikatora urządzenia. Jednakże mój stary adres IPv4 zużywa tylko 32 bity, więc w środku pozostały jeszcze 32 bity do wykorzystania. Byłoby dziwnie mieć w nich jakieś losowe dane, więc po prostu wyzerowałem je, aby uprościć schemat adresowania. Oprócz tej zmiany zainicjalizowałem swój identyfikator podsieci wartością 1, ponieważ jest to pierwsza podsieć w mojej sieci.

Nowe adresowanie zaczyna wyglądać bardziej przejrzyste i ma więcej sensu. Gdy spoglądam na ten nowy adres naszego serwera WWW, przychodzą mi do głowy kolejne pomysły na wprowadzenie ulepszeń. W tej chwili adres jest w stu procentach poprawny. Mogłbym wprowadzić go we właściwościach karty sieciowej swojego serwera i działałby bez problemu. Jednak jest tam wiele zer i nie muszę ich wszystkich przechowywać. Za każdym razem, gdy w 16-bitowym segmencie poprzedzającym rzeczywistą liczbę pojawiają się zbędne zera, można je po prostu usunąć. Na przykład nasz identyfikator podsieci i pierwsze 32 bity naszego identyfikatora urządzenia mają wiele niepotrzebnych zer, więc mogę skonsolidować adres w następujący sposób:

```
2001:AABB:CCDD:1:0:0:C0A8:0105
```

Idźmy dalej — za każdym razem, gdy 16-bitowe sekcje są złożone całkowicie z zer, można je zamienić na podwójny dwukropek. Tak więc pierwsze 32 bity naszego identyfikatora urządzenia, które są zerami, mogę zastąpić znakami ::. Poniżej zaprezentowałem adres pierwotny i po konsolidacji. Te liczby wyglądają zupełnie inaczej. Mimo że skonsolidowany adres jest znacznie łatwiejszy do odczytania, z technicznego punktu widzenia jest on dokładnie taki sam jak oryginalny:

```
2001:AABB:CCDD:0001:0000:0000:C0A8:0105
2001:AABB:CCDD:1::C0A8:0105
```

Jeśli tworzysz strukturę laboratorium lub chcesz szybko przetestować protokół IPv6, możesz użyć adresów tak prostych, jak podano w poniższym przykładzie. Dwa adresy, które tu przedstawię, są dokładnie takie same:

```
2001:0000:0000:0000:0000:0000:0000:0001
2001::1
```

Należy pamiętać, że podwójnego dwukropka można użyć tylko raz w obrębie adresu IP. Jeśli w tym samym adresie są dwa miejsca, w których można go zastosować, możesz wybrać tylko jedno z nich w celu uproszczenia zapisu. W drugim miejscu zapis musi pozostać w standardowej postaci.

Dzięki podanym powyżej informacjom powinieneś być w stanie stworzyć własny wzorzec adresów IPv6 i zacząć je przypisywać komputerom lub serwerom w sieci. Temu tematowi można poświęcić całą książkę i faktycznie powstało ich dużo.

Twoje narzędzia sieciowe

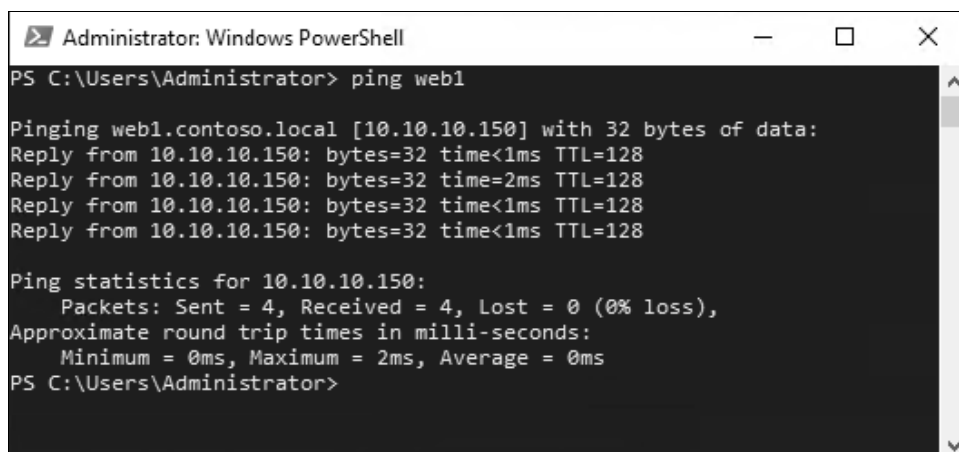
Niezależnie od tego, czy jesteś administratorem serwera, administratorem sieci, czy też łączysz ich obowiązki, w świecie systemu Windows Server możesz wykorzystać wiele narzędzi przydatnych do testowania i monitorowania połączeń sieciowych. Niektóre z tych narzędzi są zawarte w samym systemie operacyjnym i mogą być wywoływane z wiersza poleceń lub z programu PowerShell. Inne są bardziej rozbudowanymi interfejsami graficznymi, które wymagają instalacji przed uruchomieniem. Oto narzędzia sieciowe, które zamierzam zaprezentować:

- ping,
- tracert,
- pathping,
- Test-Connection,
- telnet,
- Test-NetConnection.

Wszystkie te narzędzia są bezpłatne, więc nie masz żadnej wymówki, aby się z nimi nie zapoznać.

Polecenie ping

Nawet najbardziej nowocześni specjaliści IT są zwykle zaznajomieni z poleceniem ping. Można je wywołać z wiersza polecenia lub z programu PowerShell. Służy ono do wysłania zapytania o nazwę DNS i (lub) adres IP, po którym następuje oczekiwanie na odpowiedź na to wywołanie. Polecenie ping zawsze było i wciąż jest naszym głównym narzędziem do testowania łączności między dwoma urządzeniami w sieci. W swoim kliencie Windows 10 podłączonym do sieci LAN mogę uruchomić wiersz poleceń i wykonać instrukcję `ping <ADRES_IP>`. Ponieważ w środowisku używam usługi DNS, które zamienia nazwy na adresy IP, mogę również użyć polecenia `ping <NAZWA_SERWERA>`, jak pokazano na poniższym przykładzie. Możesz zauważyć, że mój serwer odpowiada na polecenie ping, informując mnie, że jest w sieci i pracuje:

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command prompt shows a user at the C:\Users\Administrator directory running the command "ping web1". The output shows four successful replies from 10.10.10.150 with 32 bytes of data, response times less than 1ms, and a TTL of 128. Below the replies, the ping statistics for 10.10.10.150 are displayed: 4 packets sent and received, 0% loss, and approximate round trip times of 0ms minimum, 2ms maximum, and 0ms average. The prompt returns to "PS C:\Users\Administrator>".

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> ping web1

Pinging web1.contoso.local [10.10.10.150] with 32 bytes of data:
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128
Reply from 10.10.10.150: bytes=32 time=2ms TTL=128
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.150:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms
PS C:\Users\Administrator>
```

Ruch sieciowy generowany przez polecenie ping jest technicznie zwany **ruchem ICMP**. Jest to ważna informacja, ponieważ protokół ICMP jest obecnie coraz częściej blokowany, a zapory są domyślnie włączone w wielu naszych systemach i urządzeniach. ping był zawsze narzędziem, na które mogliśmy liczyć, gdy chcieliśmy dość dokładnie ustalić, czy między dwoma urządzeniami istnieje połączenie sieciowe. Niestety, już tak nie jest. Jeśli na komputerze zainstalujesz system Windows i podłączysz go do sieci, może się on poprawnie komunikować z internetem i wszystkimi serwerami. Jeśli jednak z innego urządzenia w sieci spróbujesz wysłać do tego nowego komputera polecenie ping, prawdopodobnie nie zostanie wykonane. Dlaczego tak się dzieje? Wynika to stąd, że system Windows ma wbudowane pewne zabezpieczenia, w tym zdefiniowane w zaporze blokowanie ruchu ICMP. W takim przypadku musisz wyłączyć zaporę albo dodać do niej regułę dostępu, która zezwoli na ruch ICMP. Po włączeniu takiej reguły nowy komputer zacznie odpowiadać na polecenie ping. Pamiętaj, że w dzisiejszym świecie podczas tworzenia nowych systemów lub serwerów w sieci polecenie ping nie zawsze jest narzędziem, na którym można polegać.

Na odpowiedzi ICMP można łatwo zezwolić przez dodanie reguły do zapory Windows Defender z zabezpieczeniami zaawansowanymi. Nadal jednak musiałbyś pamiętać, aby tę operację wykonywać ręcznie na każdym nowym systemie, który podłączasz do sieci. Na szczęście już wiesz, jak korzystać z zasad grupy w celu zbudowania obiektu GPO i umieszczenia go na wszystkich komputerach. Jak się zapewne domyślasz, w obiekcie GPO możesz bez problemu umieścić reguły zapory. Dodanie reguły zapory za pomocą zasad grupy to powszechny sposób na dopuszczenie lub zablokowanie protokołu ICMP w całej organizacji.

Polecenie tracert

Polecenie tracert (skrót od słów *Trace Route*) służy do śledzenia pakietu sieciowego, który przemieszcza się po Twojej sieci. Tak naprawdę to polecenie obserwuje wszystkie miejsca, które napotyka pakiet, zanim dotrze do miejsca docelowego. Te „nierówności na drodze”, przez które musi przejść pakiet sieciowy, nazywane są **przeskokami** (ang. *hops*). Trasa śledzenia pokazuje wszystkie przeskoky, które odwiedza Twój pakiet, gdy zbliża się do serwera docelowego lub dowolnego innego urządzenia, z którym próbuje się skontaktować. Moja sieć laboratoryjna

jest bardzo płaska i nudna, więc wykonanie polecenia `tracert` nie pokazałoby nam wiele. Jeśli jednak otworzę program PowerShell w komputerze podłączonym do internetu i wykonam polecenie `tracert` do serwera sieciowego takiego jak Bing, otrzymam kilka interesujących wyników:

```
PS C:\WINDOWS\system32> tracert www.bing.com

Tracing route to any.edge.bing.com [204.79.197.200]
over a maximum of 30 hops:

  1  <1 ms  <1 ms  <1 ms  192.168.8.1
  2   1 ms  <1 ms  <1 ms  192.168.128.1
  3   8 ms   7 ms   5 ms  172.17.224.1
  4  11 ms   9 ms  15 ms  172.19.253.1
  5  10 ms   9 ms  11 ms  172.31.255.1
  6  20 ms   9 ms  13 ms  htl-max1-1.iserv.net [206.114.55.1]
  7  15 ms  12 ms   8 ms  69.87.144.9
  8  23 ms  18 ms  19 ms  888-2.iserv.net [206.114.40.2]
  9  23 ms  20 ms  15 ms  g5-0-0.core3.grr.iserv.net [206.114.51.20]
 10 19 ms  11 ms  19 ms  g5-0-0.core1.grr.iserv.net [206.114.51.2]
 11 21 ms  28 ms  19 ms  GigabitEthernet4-1.GW5.DET5.ALTER.NET [152.179.10.81]
 12 25 ms  28 ms  28 ms  0.ae1.XL3.CHI13.ALTER.NET [140.222.225.179]
 13 27 ms  37 ms  54 ms  TenGigE0-6-0-1.GW2.CHI13.ALTER.NET [152.63.65.133]
 14 36 ms  34 ms  34 ms  microsoft-gw.customer.alter.net [152.179.105.130]
 15 58 ms  50 ms  46 ms  104.44.81.58
 16 34 ms  33 ms  36 ms  10.201.194.219
 17 26 ms  29 ms  29 ms  a-0001.a-msedge.net [204.79.197.200]

Trace complete.
PS C:\WINDOWS\system32>
```

Jeśli wykorzystujesz polecenie `tracert`, ale w danych wyjściowych nie chcesz widzieć żadnych informacji z usługi DNS, użyj opcji `tracert -d`, aby zachować wyłącznie adresy IP.

Informacje te mogą być bardzo przydatne podczas diagnozowania niedziałającego połączenia. Jeśli ruch, zanim dotrze do miejsca docelowego, przechodzi przez wiele przeskoków, takich jak routery i zapory sieciowe, polecenie `tracert` może być niezbędne w ustaleniu, w którym miejscu strumienia pojawia się problem. Biorąc pod uwagę, że powyższy zrzut ekranu pokazuje udaną trasę śledzenia do serwera Bing, zobaczmy teraz, jak się sprawy mają, gdy coś się zepsuje. Odlączę router internetowy i ponownie uruchomię to samo polecenie `tracert www.bing.com`. Teraz widzimy, że nadal mogę się komunikować ze swoim lokalnym routerem, ale poza niego pakiety już nie przechodzą:

```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\jkrause> tracert www.bing.com

Tracing route to any.edge.bing.com [204.79.197.200]
over a maximum of 30 hops:

  1   9 ms   1 ms   1 ms  192.168.8.1
  2  *      *      *      192.168.8.1 reports: Destination host unreachable.

Trace complete.
PS C:\Users\jkrause>
```

Polecenie pathping

Polecenie `tracert` jest użyteczne i wydaje się faktycznie standardem przy śledzeniu pakietów w sieci, jednak moim zdaniem `pathping` jest jeszcze potężniejsze. Polecenie `pathping` zasadniczo wykonuje dokładnie to samo co `tracert`, z tym wyjątkiem, że zapewnia jeszcze jedną istotną informację. Podczas używania zaprezentowanych narzędzi najczęściej chciałbyś się tylko dowiedzieć, w którym miejscu w łańcuchu przeskoków coś się psuje. Konfigurując serwery do zastosowań sieciowych, często używam urządzeń, które mają wiele różnych kart sieciowych. Gdy w systemie mamy do czynienia z wieloma kartami sieciowymi, lokalna tabela routingu jest tak samo ważna jak zewnętrzne routery i przełączniki, więc często chciałbym sprawdzić ścieżkę pakietu, aby zobaczyć, z której lokalnej karty jest wysyłany. W takiej sytuacji polecenie `pathping` jest lepsze od `tracert`. Pierwszą informacją, którą prezentuje polecenie `tracert`, jest przeskok na następne urządzenie sieciowe. Polecenie `pathping` pokazuje również, z którego interfejsu sieciowego Twojego komputera są wysyłane pakiety.

Oto przykład: często konfiguruję serwery dostępu zdalnego z wieloma kartami sieciowymi. Podczas tego procesu tworzę wiele tras na serwerze lokalnym, dzięki czemu wie on, jaki ruch należy wysłać w danym kierunku — na przykład jaki ruch powinien wychodzić z wewnętrznej karty sieciowej i jaki ruch musi wchodzić przez zewnętrzną kartę sieciową. Po wypełnieniu wszystkich tabel routingu dla wewnętrznej karty sieciowej testuję je przez wysłanie polecenia `ping` do serwera w sieci. Być może wykonanie tego polecenia nie powiedzie się, a ja nie będę wiedział, jaki jest tego powód. Mogę spróbować użyć polecenia `tracert`, ale nie dowiem się niczego nowego, ponieważ ono po prostu nie widzi pierwszego przeskoku, więc następuje przekroczenie limitu czasu. Jeśli jednak spróbuję zastosować polecenie `pathping`, pierwszy przeskok co prawda nadal nie będzie dostępny, ale teraz dowiem się, że mój ruch próbuje się wydostawać przez **ZEWNEĘTRZNĄ kartę sieciową**. Ojej! Na serwerze musieliśmy nieprawidłowo skonfigurować swoją statyczną trasę. Wiem więc, że muszę ją usunąć, a następnie utworzyć ponownie, aby ruch odbywał się przez wewnętrzną kartę sieciową.

Poniżej przedstawiono wiersz poleceń programu PowerShell uruchomiony na tym samym komputerze, którego użyłem do wykonania zrzutu ekranu dla polecenia `tracert`. Widzimy, że `pathping` wyświetla adres IP lokalnej karty sieciowej mojego laptopa, przez którą odbywa się ruch. Polecenie `tracert` nie wyświetlało tych informacji:

```
PS C:\Users\jkrause> pathping www.bing.com

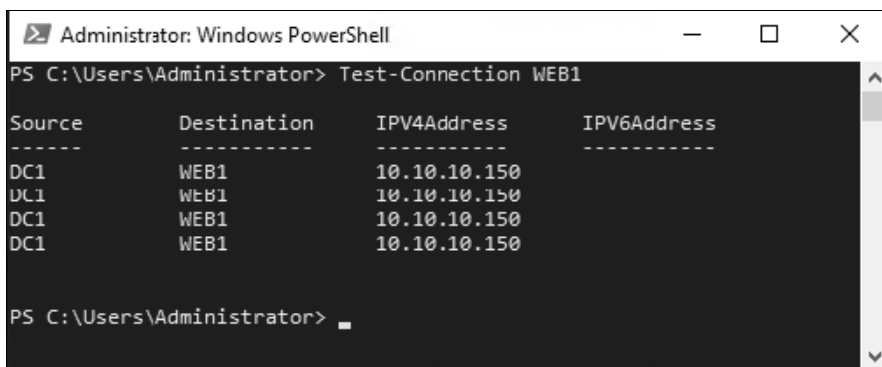
Tracing route to any.edge.bing.com [204.79.197.200]
over a maximum of 30 hops:
 0  IVO-PC-328 [192.168.8.113]
 1  192.168.8.1
 2  192.168.128.1
 3  * 192.168.8.1 reports: Destination host unreachable.

Computing statistics for 75 seconds...
Source to Here   This Node/Link
Hop  RTT      Lost/Sent = Pct  Lost/Sent = Pct  Address
0      ---      0/ 100 = 0%      0/ 100 = 0%      IVO-PC-328 [192.168.8.113]
1    1ms      0/ 100 = 0%      0/ 100 = 0%      192.168.8.1
2    ---     100/ 100 =100%   100/ 100 =100%   192.168.128.1
3    ---     100/ 100 =100%   0/ 100 = 0%      IVO-PC-328 [0.0.0.0]

Trace complete.
PS C:\Users\jkrause>
```

Polecenie Test-Connection

Instrukcje, które omówiliśmy do tej pory, można było uruchomić z wiersza poleceń lub programu PowerShell. Nadszedł jednak czas, aby zaprezentować nowsze narzędzie, które można wywołać tylko w powłoce PowerShell. Mam na myśli polecenie o nazwie `Test-Connection` — jest ono czymś w rodzaju polecenia `ping` na sterydach. Jeśli utworzymy wiersz poleceń PowerShell w naszym środowisku laboratoryjnym, a następnie uruchomimy instrukcję `Test-Connection WEB1`, otrzymamy wynik, który jest bardzo podobny do tego, jaki uzyskalibyśmy po użyciu zwykłego polecenia `ping`. Informacje są jednak wyświetlone w sposób, który moim zdaniem jest bardziej przyjazny dla użytkownika. Pojawiła się również nowa kolumna danych o nazwie *Source* (Źródło):



```

Administrator: Windows PowerShell
PS C:\Users\Administrator> Test-Connection WEB1

Source      Destination    IPv4Address    IPv6Address
-----
DC1         WEB1           10.10.10.150
DC1         WEB1           10.10.10.150
DC1         WEB1           10.10.10.150
DC1         WEB1           10.10.10.150

PS C:\Users\Administrator>
  
```

Ciekawa sprawa. Gdy uruchomiłem to polecenie, byłem zalogowany na serwerze DC1, więc był on moim urządzeniem źródłowym. Czy to oznacza, że w poleceniu `Test-Connection` mam możliwość zmiany komputera źródłowego? Tak, jest to prawda. Podobnie jak w przypadku wszystkich innych elementów zarządzających systemem Windows Server 2019, nie ma potrzeby logowania się na serwerze lokalnym. W przypadku polecenia `Test-Connection` oznacza to, że możesz otworzyć wiersz poleceń programu PowerShell w dowolnym komputerze podłączonym do sieci i przetestować połączenia między dwoma różnymi punktami końcowymi, nawet jeśli nie jesteś zalogowany do żadnego z nich. Sprawdźmy to.

Nadal jestem zalogowany na serwerze DC1, ale zamierzam użyć polecenia `Test-Connection`, aby przetestować połączenia między kilkoma serwerami w sieci. Nie tylko możesz określić komputer źródłowy inny niż ten, na którym jesteś obecnie zalogowany, ale też możesz pójść o krok dalej i za pomocą tego potężnego polecenia określić wiele źródeł i miejsc docelowych. Jeśli więc chcesz przetestować połączenia z kilku różnych maszyn źródłowych do kilku różnych miejsc docelowych, mogą to łatwo zrobić za pomocą następującego polecenia:

```
Test-Connection -Source DC1, DC2 -ComputerName WEB1, BACK1
```

Na poniższym zrzucie ekranu widać, że otrzymałem odpowiednie statystyki dotyczące komunikowania się źródłowych serwerów DC1 i DC2 z docelowymi maszynami WEB1 i BACK1. Polecenie `Test-Connection` może więc być bardzo potężnym narzędziem do monitorowania:

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Test-Connection -Source DC1, DC2 -ComputerName WEB1, BACK1

```

Source	Destination	IPV4Address	IPV6Address	Bytes	Time(ms)
DC1	WEB1	10.0.0.150		32	1
DC1	WEB1	10.0.0.150		32	0
DC1	WEB1	10.0.0.150		32	4
DC1	WEB1	10.0.0.150		32	1
DC1	BACK1	10.0.0.10		32	0
DC1	BACK1	10.0.0.10		32	4
DC1	BACK1	10.0.0.10		32	2
DC1	BACK1	10.0.0.10		32	1
DC2	WEB1	10.0.0.150		32	0
DC2	WEB1	10.0.0.150		32	2
DC2	WEB1	10.0.0.150		32	1
DC2	WEB1	10.0.0.150		32	0
DC2	BACK1	10.0.0.10		32	1
DC2	BACK1	10.0.0.10		32	1
DC2	BACK1	10.0.0.10		32	1
DC2	BACK1	10.0.0.10		32	1

```

PS C:\Users\Administrator>

```

Jeszcze jedną przydatną funkcjonalnością, na którą należy zwrócić uwagę, jest to, że za pomocą przełącznika `-Quiet` można dość łatwo zmniejszyć ilość informacji wyjściowych. Przez dodanie opcji `-Quiet` do polecenia `Test-Connection` likwidujesz prawie wszystkie dane wyjściowe i otrzymujesz tylko proste potwierdzenia `True` lub `False` w zależności od tego, czy połączenie się powiodło czy nie. Niestety, nie można łączyć przełączników `-Source` i `-Quiet`, ale jeśli używasz polecenia `Test-Connection` z oryginalnego komputera źródłowego, na którym jesteś zalogowany (jak większość z nas i tak robi), wówczas opcja `-Quiet` działa świetnie. W większości przypadków tak naprawdę zależy nam na zwykłej odpowiedzi *Tak* lub *Nie*, by dowiedzieć się, czy połączenia działają, a niekoniecznie chcielibyśmy oglądać na ekranie dokładne wyniki wszystkich prób. Po użyciu opcji `-Quiet` otrzymujemy dokładnie to, o co nam chodzi:

```
Test-Connection -Quiet -ComputerName WEB1, BACK1, DC2, CA1
```

Gdybyśmy użył polecenia `Test-Connection` w standardowy sposób, aby spróbować się połączyć ze wszystkimi serwerami w swojej sieci, otrzymałbym bardzo dużą liczbę wyników. Dzięki wykorzystaniu parametru `-Quiet` otrzymuję tylko krótkie odpowiedzi `True` (*Prawda*) lub `False` (*Falsz*), jednoznacznie informujące, czy można się skontaktować z danym serwerem:

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Test-Connection -Quiet -ComputerName WEB1, BACK1, DC2, CA1
True
True
True
True
PS C:\Users\Administrator>

```

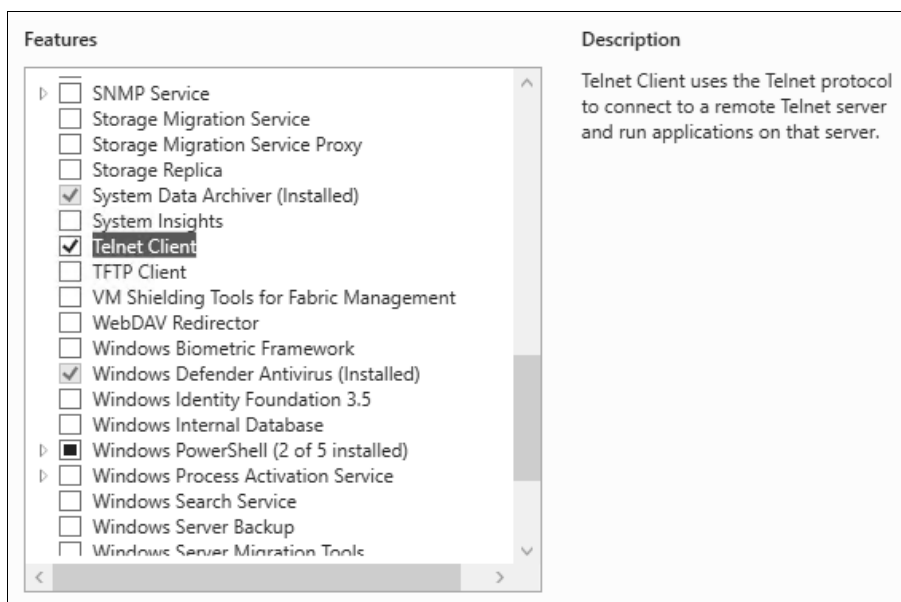
Polecenie telnet

Polecenie telnet zapewnia sporo możliwości zdalnego zarządzania. Zasadniczo umożliwia ono nawiązanie połączenia między dwoma komputerami w celu zarządzania maszyną zdalną za pośrednictwem wirtualnego połączenia terminalowego. Nie będziemy jednakże omawiać tu żadnej rzeczywistej funkcjonalności zapewnianej przez telnet, ponieważ w kontekście sieci uważam, że jest to całkiem przydatne, proste narzędzie służące do testowania połączeń i nie trzeba nic wiedzieć o tym, jakie ma zaawansowane opcje.

Gdy analizowaliśmy polecenie ping, wspominaliśmy o wadach protokołu ICMP. Można go łatwo zablokować i przez to w obecnie tworzonych sieciach coraz częściej polecenie ping nie działa prawidłowo. Jest to smutne, ponieważ było ono zawsze najczęstszą formą testowania połączenia sieciowego. W rzeczywistości jednak, jeśli ułatwiało nam życie, ułatwiało też życie hakerom. Skoro nie możemy już polegać na poleceniu ping, gdy chcemy uzyskać informację, czy istnieje połączenie ze zdalnym systemem, czego powinniśmy używać w zamian? Innym często spotykanym przypadkiem jest sytuacja, w której sam serwer może reagować poprawnie, ale określona usługa działająca na nim ma problem i nie odpowiada. Proste polecenie ping może informować, że serwer jest co prawda w trybie online, ale nie może nam nic powiedzieć na temat konkretnej usługi. Z wykorzystaniem poleceń klienta Telnetu możemy w prosty sposób wysłać zapytania do serwera. Co ważniejsze, możemy sprawdzić pojedynczą usługę na tym serwerze, aby upewnić się, że działa ona poprawnie.

Oto praktyczny przykład. Często konfiguruje serwery WWW wymagające dostępu do internetu. Po uruchomieniu nowego serwera warto sprawdzić, czy można się do niego dostać z internetu, i upewnić się, że odpowiada. Być może jednak sama witryna nie jest jeszcze dostępna, więc nie mogę jej wczytać za pomocą przeglądarki Internet Explorer lub innej. Jest też całkiem prawdopodobne, że na tym serwerze lub na poziomie firmowej zapory sieciowej wyłączyliśmy odpowiadanie na polecenie ping, ponieważ w internecie często blokuje się protokół ICMP w celu zmniejszenia podatności na ataki. Tak więc mój nowy serwer działa i wydaje się nam, że sieć jest poprawnie skonfigurowana, ale nie mogę użyć polecenia ping, serwer bowiem z założenia nie odpowie na nie. Czego mogę więc użyć do przeprowadzenia testów? Polecenia telnet. Używając go, mogę nakazać komputerowi, aby sprawdził określony port na moim nowym serwerze internetowym i dowiedział się, czy istnieje odpowiednie połączenie. W ten sposób ustanawia się połączenie gniazda z portem na tym serwerze, co jest znacznie bardziej zbliżone do rzeczywistego ruchu generowanego przez użytkownika niż użycie polecenia ping. Jeśli polecenie telnet połączy się pomyślnie, będziesz już wiedział, że ruch dociera do serwera, a usługa działająca na porcie, którego dotyczy zapytanie, wydaje się odpowiadać poprawnie.

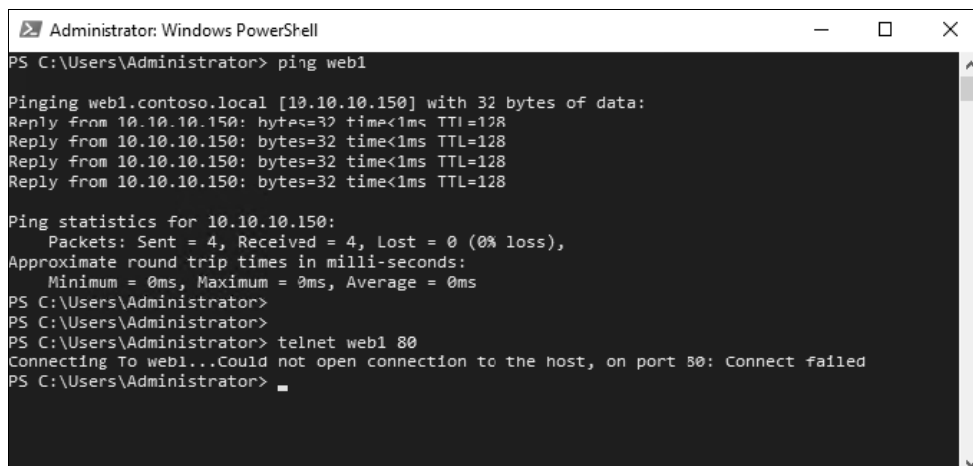
Możliwość korzystania z usługi Telnet nie jest domyślnie dostępna w systemie Windows Server 2019 ani w żadnym innym systemie operacyjnym Windows, dlatego najpierw musimy przejść do Menedżera serwera i wybrać opcję *Add Roles and Features* (*Dodaj rolę i funkcje*), aby zainstalować funkcję o nazwie *Telnet Client* (*Klient Telnetu*):



Klienta Telnetu powinieneś zainstalować tylko na komputerze, na którym użyjesz wiersza poleceń. Nie musisz nic robić na zdalnym serwerze, z którym się łączysz.

Gdy funkcja klienta Telnetu zostanie już zainstalowana, będziemy mogli jej użyć z wiersza poleceń lub programu PowerShell, dzięki czemu spróbujemy się połączyć ze swojego komputera do usługi zdalnej. Wystarczy podać nazwę (lub adres IP) serwera docelowego i jego port. Następnie telnet po prostu połączy się poprawnie lub przekroczy limit czasu, a my na podstawie otrzymanego wyniku będziemy mogli stwierdzić, czy dana usługa na serwerze odpowiada. Wypróbujmy to narzędzie na naszym serwerze WWW. Właśnie wyłączyłem stronę internetową w konsoli IIS, więc mamy obecnie sytuację, w której co prawda serwer jest online, ale sama strona nie działa. Jeśli spróbuję użyć polecenia ping z serwerem WEB1, uzyskam poprawną odpowiedź. Jak widać, narzędzia do monitorowania serwera oparte na protokole ICMP mogłyby przekazać fałszywe informacje, ponieważ stwierdziłyby, że serwer działa, nawet jeśli nasza strona jest niedostępna. Na poniższym zrzucie ekranu widać, że po wykonaniu polecenia ping próbowałem również wysłać zapytanie na port 80 serwera WEB1. W tym celu użyłem polecenia telnet web1 80. Okazało się, że upłynął limit czasu. Dzięki temu możemy stwierdzić, że strona internetowa działająca na porcie 80 nie odpowiada (zobacz pierwszy rysunek na następnej stronie).

Po włączeniu strony możemy ponownie spróbować wykonać polecenie telnet web1 80 i teraz już nie otrzymamy komunikatu o przekroczeniu limitu czasu. Tym razem wewnątrz okna programu PowerShell zostanie wyczyszczone, a pod jego górną krawędzią pojawi się migający kursor. Choć nie otrzymałem jawnego komunikatu „Super, połączyłeś się!”, ten migający kursor wskazuje, że połączenie z serwerem zostało pomyślnie nawiązane po porcie 80, co oznacza, że witryna jest w trybie online i odpowiada (zobacz drugi rysunek na następnej stronie).



```

Administrator: Windows PowerShell
PS C:\Users\Administrator> ping web1

Pinging web1.contoso.local [10.10.10.150] with 32 bytes of data:
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128
Reply from 10.10.10.150: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.150:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator> telnet web1 80
Connecting To web1...Could not open connection to the host, on port 80: Connect failed
PS C:\Users\Administrator>

```



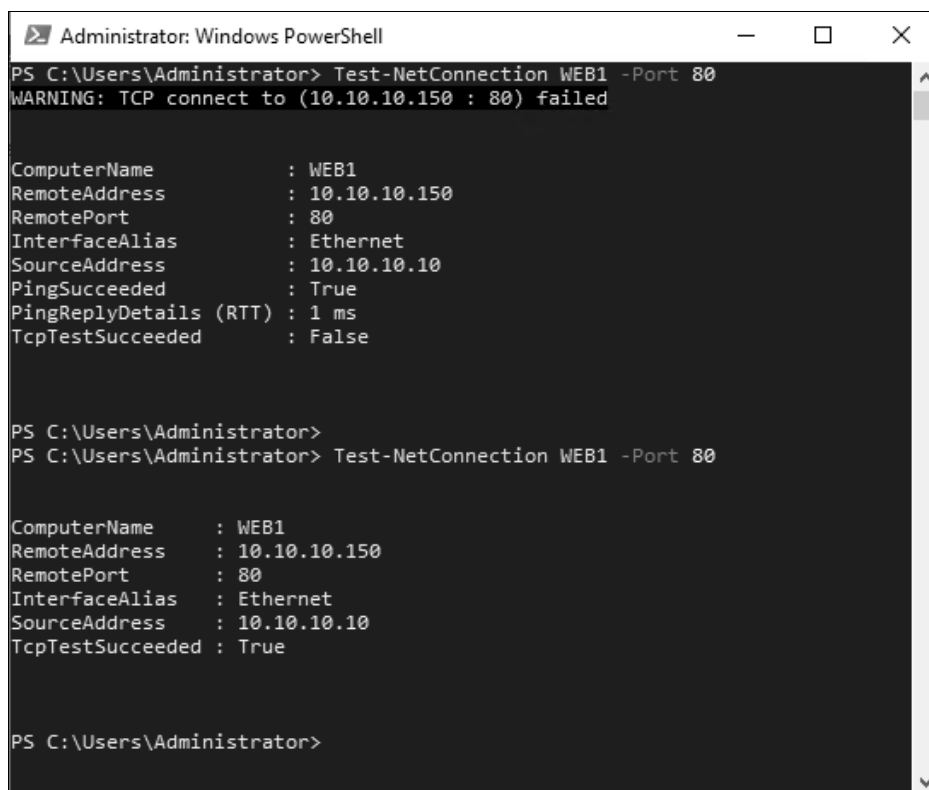
Po utworzeniu udanego połączenia za pomocą klienta Telnetu możesz się zastanawiać, jak wrócić do zwykłego interfejsu programu PowerShell. Naciśnij jednocześnie klawisze **Ctrl+]** (ten drugi to klawisz prawego nawiasu kwadratowego, znajdujący się zazwyczaj na klawiaturze obok klawisza ukośnika odwrotnego), wpisz słowo **quit**, a następnie naciśnij klawisz **Enter**. Powinieneś powrócić do wiersza poleceń programu PowerShell.

Polecenie Test-NetConnection

Jeśli poleceniu ping odpowiada ulepszone polecenie programu PowerShell o nazwie **Test-NetConnection**, moglibyśmy zapytać, czy istnieje również poprawione narzędzie, które działa podobnie jak **telnet** i służy do testowania połączeń po określonych portach. Odpowiedź jest twierdząca. Polecenie **Test-NetConnection** programu PowerShell to kolejny sposób na sprawdzenie określonych portów lub usług w systemie zdalnym, przy czym uzyskane dane wyjściowe są bardziej przyjazne niż w przypadku usługi Telnet.

Wykonajmy te same testy co poprzednio, jeszcze raz sprawdzając port 80 na serwerze WEB1. Na poniższym zrzucie ekranu widać, że polecenie zostało uruchomione dwukrotnie. Za pierwszym razem witryna WWW serwera WEB1 została wyłączona, a moje połączenie z portem 80 nie powiodło się. Za drugim razem włączyłem ponownie witrynę i uzyskałem udane połączenie.

```
Test-NetConnection WEB1 -Port 80
```



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Test-NetConnection WEB1 -Port 80
WARNING: TCP connect to (10.10.10.150 : 80) failed

ComputerName      : WEB1
RemoteAddress     : 10.10.10.150
RemotePort        : 80
InterfaceAlias    : Ethernet
SourceAddress     : 10.10.10.10
PingSucceeded     : True
PingReplyDetails (RTT) : 1 ms
TcpTestSucceeded  : False

PS C:\Users\Administrator>
PS C:\Users\Administrator> Test-NetConnection WEB1 -Port 80

ComputerName      : WEB1
RemoteAddress     : 10.10.10.150
RemotePort        : 80
InterfaceAlias    : Ethernet
SourceAddress     : 10.10.10.10
TcpTestSucceeded  : True

PS C:\Users\Administrator>
```

Śledzenie pakietów za pomocą programów Wireshark lub Message Analyzer

Może także zajść potrzeba głębszego przeanalizowania pakietów sieciowych. Teraz wkraczamy na terytorium, na którym dobrze się czuje zespół administratorów sieciowych. Jeśli jednak poznasz odpowiednie narzędzia, być może przed wezwaniem pomocy zdołasz samodzielnie rozwiązać problem. Korzystanie z narzędzi wiersza poleceń do sprawdzania stanu serwerów i usług jest bardzo przydatne, ale czasami może nie wystarczyć. Na przykład masz aplikację kliencką, która nie łączy się z serwerem, i nie wiesz, jaki jest tego powód. Narzędzia takie jak ping, a nawet telnet, mogą być w stanie połączyć się pomyślnie, co wskazuje na poprawne skonfigurowanie routingu sieciowego, ale sam program po uruchomieniu nie łączy się. Jeśli dzienniki zdarzeń aplikacji nie pomagają w rozwiązaniu problemu, możesz się dokładniej przyjrzeć pakietom sieciowym, które program próbuje wysłać w kierunku serwera.

W takiej sytuacji przydają się narzędzia Wireshark i Message Analyzer. Oba są bezpłatne i oba można łatwo pobrać. Mają także te same funkcje. Zostały zaprojektowane do przechwytywania ruchu sieciowego opuszczającego system lub docierającego do niego, a także wyświetlania informacji zawartych w samych pakietach, abyś mógł dokładniej zapoznać się z sytuacją. W przypadku naszej przykładowej aplikacji, która nie może się połączyć, mógłbyś uruchomić jedno z tych narzędzi na komputerze klienckim, aby obserwować ruch wychodzący. Możesz je także uruchomić na serwerze aplikacji, aby śledzić pakiety przychodzące od klienta.

Każde z narzędzi działa jednak trochę odmiennie, a ponieważ nie mamy tutaj miejsca, aby je dokładnie omówić, poniżej podam jedynie odnośniki, za pomocą których możesz je pobrać w celu przetestowania:

1. **Wireshark:** <https://www.wireshark.org/download.html>,
2. **Microsoft Message Analyzer:** <https://www.microsoft.com/en-us/download/details.aspx?id=44226>.

Narzędzie TCPView

Narzędzia, które omówiliśmy do tej pory, są świetne i mogą być używane na co dzień do testowania określonych zasobów, jednakże zdarzają się sytuacje, w których musisz się wycofać i przede wszystkim dowiedzieć, czego w ogóle szukasz. Być może obsługujesz aplikację na komputerze i nie masz pewności, z jakim serwerem się ona komunikuje. A może podejrzewasz, że komputer „złapał” wirusa, który próbuje się połączyć z zarządzającym nim serwerem, a Ty chciałbyś zidentyfikować jego lokalizację lub sam proces nawiązujący połączenie. W takich sytuacjach pomocne byłoby narzędzie, które można by uruchomić na komputerze lokalnym w celu zaprezentowania w jasny i zwięzły sposób wszystkich aktywnych strumieni ruchu sieciowego. Właśnie taką funkcję ma narzędzie **TCPView**. Zostało ono pierwotnie stworzone przez zespół programistów Sysinternals. Być może słyszałeś o innych stworzonych przez niego narzędziach, takich jak ProcMon i FileMon. Program TCPView wyświetla w czasie rzeczywistym wszystkie aktywne połączenia TCP i UDP używane na danym komputerze. Ważne jest również to, że nie musisz go instalować. Jest to samodzielny plik wykonywalny, dzięki czemu można go łatwo użyć, a przy tym po zakończeniu pracy sam usuwa swoje ślady z pamięci.

Program TCPView możesz pobrać z adresu <https://docs.microsoft.com/en-us/sysinternals/downloads/tcpview>.

Po prostu skopiuj plik na komputer lub serwer, który chcesz monitorować, a następnie kliknij go dwukrotnie. Poniżej zamieszczono zrzut ekranu z interfejsem programu TCPView, który działał na moim lokalnym komputerze i wyświetlał wszystkie połączenia tworzone w danym momencie przez system Windows i inne oprogramowanie. Możesz zatrzymać przewijanie ekranu, aby przyjrzeć się bliżej wynikom, a także zdefiniować filtry, by ograniczyć ilość danych i znaleźć to, czego naprawdę szukasz. Filtry pozwalają pozbyć się *śmieci*, dzięki czemu można się przyjrzeć bliżej określonemu docelowemu miejscu lub identyfikatorowi procesu:

Process	Protocol	Local Address	Remote Address	State
dasHost.exe:2012	UDP	IV0-PC-328:ws-discovery	...	
dasHost.exe:2012	UDP	IV0-PC-328:ws-discovery	...	
dasHost.exe:2012	UDP	IV0-PC-328:56988	...	
dasHost.exe:2012	UDPV6	ivo-pc-328:3702	...	
dasHost.exe:2012	UDPV6	ivo-pc-328:3702	...	
dasHost.exe:2012	UDPV6	ivo-pc-328:56989	...	
chrome.exe:10708	TCP	ivo-pc-328:61556	r1.ycpi.vip.nyc.yahoo.net:https	ESTABLISHED
chrome.exe:10708	TCP	ivo-pc-328:61562	ne1.onepush.vip.ne1.yahoo.com:https	ESTABLISHED
chrome.exe:10708	TCP	ivo-pc-328:61580	pr.comet.vip.bf1.yahoo.com:https	ESTABLISHED
chrome.exe:10708	TCP	ivo-pc-328:63472	bf1.onepush.vip.bf1.yahoo.com:https	ESTABLISHED
chrome.exe:10708	TCP	ivo-pc-328:63475	lga15s42-in-47.1e100.net:https	ESTABLISHED
chrome.exe:10708	TCP	ivo-pc-328:63476	pr.comet.vip.bf1.yahoo.com:https	ESTABLISHED
AppleMobileDeviceService.exe...	TCP	IV0-PC-328:27015	IV0-PC-328:0	LISTENING
AppleMobileDeviceService.exe...	UDP	IV0-PC-328:49664	...	
AppleMobileDeviceService.exe...	UDP	IV0-PC-328:49665	...	
[System Process]:0	TCP	ivo-pc-328:63449	134.170.188.139:https	TIME_WAIT
[System Process]:0	TCP	ivo-pc-328:60231	132.245.247.210:https	TIME_WAIT
[System Process]:0	TCP	ivo-pc-328:60277	pr.comet.vip.bf1.yahoo.com:https	TIME_WAIT
[System Process]:0	TCP	ivo-pc-328:63465	132.245.247.210:https	TIME_WAIT
[System Process]:0	TCP	ivo-pc-328:63469	207.46.7.252:http	TIME_WAIT

Endpoints: 87 Established: 18 Listening: 16 Time Wait: 5 Close Wait: 0

Tworzenie tablicy routingu

Gdy słyszysz termin „**tablica routingu**”, możesz łatwo wpaść w nawyk przyjmowania, że dotyczy on czegoś, z czym muszą się zmagać jedynie administratorzy sieci i co jest konfigurowane w routerach i zaporach sieciowych. Przecież to zagadnienie nie dotyczy administratora serwera, mam rację? Podłączanie serwerów do sieci stało się bardzo łatwe, ponieważ wymaga podania jedynie adresu IP, maski podsieci i domyślnej bramy. Po spełnieniu tych warunków możemy natychmiast komunikować się ze wszystkimi urządzeniami wewnątrz swojej sieci lokalnej. Chociaż sprzęt sieciowy i administratorzy ukrywają wiele szczegółów związanych z magicznym działaniem sieci, ważne jest, aby zrozumieć, w jaki sposób w środowisku Windows funkcjonuje routing. Mogą pojawić się sytuacje, w których będziesz musiał zmodyfikować lub zbudować tablicę routingu bezpośrednio w samym systemie Windows Server.

Serwery o wielu adresach

Uruchamianie serwerów z wieloma adresami to przypadek, dla którego z pewnością będziesz musiał zrozumieć działanie lokalnej tablicy routingu i ją zmodyfikować. Jeśli uważasz, że to zagadnienie Cię nie dotyczy, ponieważ nigdy wcześniej nie słyszałeś o *wieloadresowości*, zastanów się ponownie. To słowo po prostu oznacza, że Twój serwer ma więcej niż jedną kartę sieciową. Z pewnością może się tak zdarzyć, nawet jeśli prowadzisz niewielki sklep, który nie używa wielu serwerów. Często serwery Small Business lub Essentials mają wiele interfejsów sieciowych oddzielających ruch wewnętrzny od ruchu internetowego. Innym przykładem urządzenia z wieloma adresami może być serwer dostępu zdalnego, który udostępnia funkcje pośrednika (proxy), DirectAccess lub VPN. Kolejnym powodem, dla którego warto zrozumieć wielo-

adresowość, są serwery Hyper-V. Bardzo często mają one wiele kart sieciowych, ponieważ działające na nich maszyny wirtualne mogą wymagać połączenia z różnymi sieciami fizycznymi w organizacji.

Gdy już ustaliliśmy, czym jest serwer o wielu adresach, możesz nadal się zastanawiać, dlaczego o tym wspominam. Jeśli mam więcej niż jedną kartę sieciową, czyż konfigurując każdą z nich w systemie Windows, nie nadam im po prostu określonych adresów IP, tak jak zrobiłbym w przypadku pojedynczej karty sieciowej na dowolnym serwerze? Tak i nie. Tak — konfigurujesz adresy IP na każdej karcie sieciowej, ponieważ jest to niezbędne do odpowiedniej identyfikacji i transportu pakietów w sieci. Nie — nie konfigurujesz wszystkich kart sieciowych na swoim serwerze w ten sam sposób. Jest jeden kluczowy element, o którym należy pamiętać i który należy uwzględnić, aby pakiety na serwerze wieloadresowym były poprawnie przesyłane.

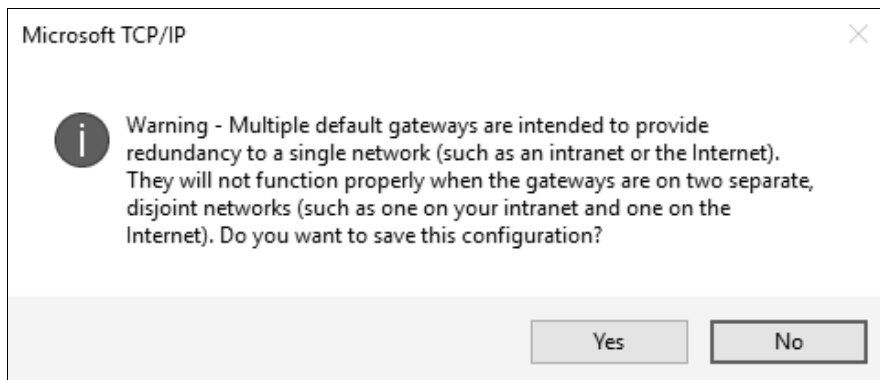
Tylko jedna brama domyślna

Oto najważniejsza zasada. Gdy serwer zawiera wiele kart sieciowych, można mieć tylko jedną bramę domyślną. Jedną dla całego serwera. Oznacza to, że będziesz mieć jedną kartę sieciową z domyślną bramą, natomiast pozostałe karty *NIE* będą miały zdefiniowanej domyślnej bramy w swoich ustawieniach TCP/IP. Jest to bardzo ważne. Celem istnienia domyślnej bramy jest zdefiniowanie *ścieżki ostatniej szansy*. Gdy system Windows chce wysłać pakiet do miejsca docelowego, przegląda lokalną tablicę routingu (tak, istnieje tablica routingu, nawet jeśli jej nie skonfigurowałeś ani nie widziałeś) i sprawdza, czy dla docelowej podsieci, do której musi dotrzeć ten pakiet, istnieje określona trasa statyczna. Jeśli tak, wysyła tą trasą pakiet do miejsca docelowego. Gdy w tablicy routingu nie ma trasy statycznej, Windows próbuje skorzystać z bramy domyślnej i przekierowuje ruch na jej adres. Na wszystkich serwerach z pojedynczą kartą sieciową domyślną bramą jest router, który zawiera wszystkie informacje o routingu dla Twojej sieci. Tak więc serwer po prostu przekazuje pakiet danych routerowi, który wykonuje resztę pracy.

Gdy mamy wiele kart sieciowych w systemie Windows Server, nie możemy zdefiniować dla każdej z nich bramy domyślnej, ponieważ zakłóci to przepływ ruchu z Twojego serwera. Nie da się przewidzieć, która z kart zostanie wybrana podczas każdej transmisji sieciowej wykorzystującej bramę domyślną. Pomogłem wielu osobom, które miały właśnie taki problem w swoich serwerach. Musiały one używać serwera jako mostu między dwiema sieciami lub z jakiegoś powodu podłączyły go do wielu różnych sieci, a teraz miały problem, ponieważ połączenie czasami wydaje się działać, a czasem nie. Zaczynam więc przeglądać właściwości kart sieciowych i odkrywam, że każda z nich ma wprowadzony domyślny adres bramy we właściwościach protokołu TCP/IP. No i proszę, znaleźliśmy przyczynę. Gdy system próbuje wysłać pakiety, jest całkowicie zdezorientowany, ponieważ nie wie, z której bramy powinien skorzystać w danym momencie.

Jeśli kiedykolwiek próbowałeś konfigurować domyślne bramy w większej liczbie kart sieciowych na tym samym serwerze, prawdopodobnie znasz już komunikat ostrzegawczy wyświetlany po wykonaniu tej czynności. Przeprowadźmy test. Dodałem kolejną kartę sieciową do jednego ze swoich serwerów i skonfigurowałem ustawienia IP tylko na jednej z nich. Teraz dodam

nowy adres IP, maskę podsieci i domyślną bramę do swojej drugiej karty sieciowej. Aby zapisać zmiany, klikam przycisk OK i pojawia się następujące okno:



Okno zawiera jedno z tych ostrzeżeń, które łatwo pominąć z powodu jego nieco tajemniczej natury, ale i tak rozumiesz sedno: działasz na własne ryzyko! A co w tym momencie robi większość administratorów? Po prostu klika i zapisuje zmiany. Następnie zaczynają się pojawiać problemy z routingiem. Może nie dzisiaj, ale następnym razem, gdy ponownie uruchamiasz ten serwer, a być może dopiero 3 tygodnie później. Na pewno jednak w pewnym momencie Twój serwer zacznie wysyłać pakiety do niewłaściwych miejsc i powodować problemy.

Definiowanie trasy

Co więc należy zrobić w takiej sytuacji? Zbudować statyczną tablicę routingu. Jeśli na serwerze zainstalowano wiele kart sieciowych, co powoduje, że ma on wiele adresów IP, musisz w tablicy routingu poinformować system Windows, której karty sieciowej należy używać do jakiego rodzaju ruchu. Gdy pakiety będą musiały zostać wysłane do określonego miejsca docelowego, tablica routingu będzie świadoma istnienia różnych ścieżek sieciowych i odpowiednio przekieruje ruch. Nadal będziesz polegać na routerach, które zajmą się resztą ruchu, ale dostarczenie pakietów do właściwego routera przez wysłanie ich przez odpowiednią fizyczną kartę sieciową jest kluczem do tego, by cała operacja przebiegła poprawnie w Twoim serwerze wieloadresowym.

Gdy już rozumiesz, dlaczego tablica routingu jest ważna, i wiesz, w jaki sposób należy jej używać, zabierzmy się do pracy i dodajmy kilka tras na serwerze z podwójną kartą sieciową. Działania wykonamy za pomocą wiersza poleceń oraz programu PowerShell, jednakże zastosowana składnia będzie odmienna w zależności od tego, jakiego narzędzia użyjesz.

Dodawanie trasy za pomocą wiersza poleceń

Zanim będziemy mogli zaplanować nową trasę, powinniśmy uzyskać więcej informacji o konfiguracji sieci na naszym serwerze. Ma on dwie karty sieciowe: jedna jest podłączona do sieci wewnętrznej, a druga jest podłączona do strefy DMZ z dostępem do internetu. Ponieważ mogą mieć tylko jeden domyślny adres bramy, został on zdefiniowany tylko w karcie sieciowej DMZ, nie ma bowiem możliwości, abym dodał trasy dla każdej podsieci, z którą trzeba się

będzie połączyć przez internet. Przez umieszczenie domyślnej bramy w swojej karcie DMZ sprawiam, że wewnętrzna karta sieciowa jest jej pozbawiona i przez to ma bardzo ograniczone możliwości kontaktowania się z innymi sieciami. Wewnętrzna podsieć, do której jestem fizycznie podłączony, ma adres 10.10.10.0/24, więc mogę się teraz kontaktować z dowolnym jej elementem, poczynając od adresu 10.10.10.1, a kończąc na 10.10.10.254. Te adresy są dostępne lokalnie i nie wymagają żadnej bramy (trasa typu **on-link**). Jestem podłączony bezpośrednio do tej podsieci, dlatego mój serwer automatycznie wie, jak należy w niej kierować ruchem. Jednakże za pośrednictwem swojej wewnętrznej karty sieciowej nie mogę się skontaktować z niczym innym, gdyż tablica routingu nie wie nic o pozostałych podsieciach, które istnieją w mojej sieci wewnętrznej. Na przykład mam dodatkową podsieć 192.168.16.0/24 zawierającą urządzenia, z którymi muszę się łączyć ze swojego nowego serwera. Gdybym musiał się teraz połączyć z jednym z tych urządzeń, pakiety zostałyby wysłane przez kartę sieciową DMZ. Ponieważ tablica routingu na moim serwerze nie ma pojęcia, jak radzić sobie z ruchem o adresach 192.168.16, wysłałaby go do bramy domyślnej. Poniżej zaprezentowałem ogólną składnię instrukcji route, której musimy użyć, aby ruch został skierowany z naszego serwera do nowej podsieci:

```
route add -p <ID_PODSIECI> mask <MASKA_PODSIECI> <BRAMA> IF <ID_INTERFEJSU>
```

Zanim będziemy mogli użyć unikatowej instrukcji route dotyczącej dodania sieci 192.168.16, musimy wykonać trochę pracy detektywistycznej i dowiedzieć się, co oznaczają poszczególne opcje. Poniżej przedstawiono opis poszczególnych elementów, które są wymagane do poprawnego uruchomienia instrukcji route:

- **-p**: ta opcja sprawi, że polecenie zdefiniuje trasę na czas niekreślony. Jeśli zapomnisz wstawić opcję **-p** do instrukcji route add, nowa trasa zniknie przy następnym uruchomieniu serwera. Niedobrze.
- **ID_PODSIECI**: jest to podsieć, którą dodajemy; w naszym przypadku jest równa 192.168.16.0.
- **MASKA_PODSIECI**: maska podsieci dla nowej trasy (255.255.255.0).
- **BRAMA**: ten parametr jest trochę mylący. Często zdaje się oznaczać, że należy wprowadzić adres bramy dla nowej podsieci, jednakże nie jest to prawdą. W rzeczywistości za pomocą tego parametru definiujesz pierwszy przeskok, na który trafią wysyłane przez serwer dane. Jaki byłby domyślny adres bramy w wewnętrznej karcie sieciowej? W przypadku naszej sieci jest to 10.10.10.1.
- **ID_INTERFEJSU**: podanie numeru interfejsu nie jest konieczne do utworzenia trasy, ale jeśli tego nie zrobisz, jest szansa, że Twoja trasa zostanie powiązana z niewłaściwą kartą sieciową i ruch zostanie wysłany w złym kierunku. Doświadczyłem już czegoś takiego, więc od tego czasu zawsze podaję numer identyfikacyjny interfejsu karty sieciowej. Zazwyczaj jest to jedno- lub dwucyfrowa liczba, która jest identyfikatorem zdefiniowanym przez system Windows dla wewnętrznej karty sieciowej. Odczytując wyniki polecenia route print, możemy się dowiedzieć, jaki jest numer identyfikacyjny interfejsu:

```
Administrator: C:\Windows\system32\cmd.exe

C:\Users\Administrator>route print

=====
Interface List
 5...00 15 5d 08 58 07 .....Microsoft Hyper-V Network Adapter
 6...00 15 5d 08 58 09 .....Microsoft Hyper-V Network Adapter #2
 1.....Software Loopback Interface 1
=====
```

Na samym początku listingu zawierającego wynik działania polecenia `route print` zostają wyświetlone wszystkie karty sieciowe dostępne w wymienionym systemie. W naszym przypadku wewnętrzna karta sieciowa jest położona na najwyższym miejscu listy. Zidentyfikowałem ją przez porównanie adresu MAC uzyskanego z danych wyjściowych polecenia `ipconfig /all`. Jak widać, moja wewnętrzna karta interfejsu sieciowego ma numer 5, więc w instrukcji `route add` zamierzam użyć parametru `IF 5`, aby upewnić się, że nowa trasa zostanie powiązana z tą kartą.

Oto nasza pełna instrukcja `route add`:

```
route add -p 192.168.16.0 mask 255.255.255.0 10.10.10.1 if 5
```

```
Administrator: C:\Windows\system32\cmd.exe

C:\>route add -p 192.168.16.0 mask 255.255.255.0 10.10.10.1 if 5
OK!

C:\>_
```

Jeśli teraz wykonasz polecenie `route print`, będziesz mógł zobaczyć nową trasę 192.168.16.0 wymienioną w sekcji *Persistent Routes (Trasy statyczne)* tablicy routingu. Z naszego serwera możemy obecnie wysyłać pakiety danych do tej podsieci. Ilekroć w serwerze pojawi się pakiet, który będzie musiał zostać skierowany do podsieci 192.168.16.x, będzie on przesyłany przez wewnętrzną kartę sieciową w kierunku routera 10.10.10.1. Router następnie odbierze ten pakiet i prześle do podsieci 192.168.16:

```
=====
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
      0.0.0.0             0.0.0.0      10.10.10.1    Default
  192.168.16.0    255.255.255.0      10.10.10.1         1
=====
```

Usuwanie trasy

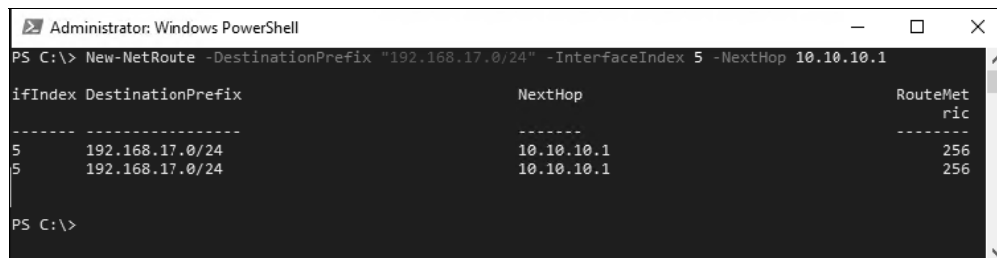
Czasami możesz błędnie wprowadzić instrukcję route. Najlepszym wyjściem jest wtedy po prostu usunięcie złej trasy, a następnie ponowne użycie instrukcji route add z poprawną składnią. Być może istnieją też inne powody, dla których od czasu do czasu konieczne będzie usuwanie tras, dlatego warto zapoznać się z tym poleceniem. Usuwanie tras jest znacznie prostsze niż ich dodawanie. Musisz tylko znać identyfikator podsieci dla trasy, którą chcesz usunąć, a następnie po prostu wykonać polecenie route delete <ID_PODSIECI>. Na przykład, aby pozbyć się naszej trasy 192.168.16.0, którą utworzyliśmy za pomocą wiersza poleceń, użyłbym takiej instrukcji:

```
route delete 192.168.16.0
```

Dodawanie trasy za pomocą programu PowerShell

Ponieważ PowerShell króluje, jeśli chodzi o większość zadań wykorzystujących wiersz poleceń w systemie Windows Server, powinniśmy mieć możliwość realizacji tej samej operacji również za pomocą tego nowego interfejsu. Możesz oczywiście wprowadzić to samo polecenie add route w wierszu poleceń programu PowerShell i będzie ono poprawnie działać, jednak istnieje specjalne polecenie cmdlet, którego również możemy użyć. Wykorzystajmy więc instrukcję New-NetRoute, aby dodać kolejną podsieć do naszej tablicy routingu — tym razem dodamy 192.168.17.0. Oto polecenie, którego możemy użyć:

```
New-NetRoute -DestinationPrefix "192.168.17.0/24" -InterfaceIndex 5 -NextHop 10.10.10.1
```



```

Administrator: Windows PowerShell
PS C:\> New-NetRoute -DestinationPrefix "192.168.17.0/24" -InterfaceIndex 5 -NextHop 10.10.10.1

ifIndex DestinationPrefix      NextHop      RouteMetric
-----
5        192.168.17.0/24              10.10.10.1   256
5        192.168.17.0/24              10.10.10.1   256
PS C:\>
  
```

Widzimy, że składnia polecenia jest podobna do składni poprzedniego, ale nieco bardziej przyjazna. Zamiast wprowadzać całą maskę i adres podsieci, możesz użyć metody z ukośnikiem, aby określić podsieć i jej maskę w ramach tego samego identyfikatora. Ponadto nie znajdziemy tu opcji zwanej bramą, która zawsze jest nieco myląca, ale zamiast niej mamy parametr o nazwie NextHop (dosł. „następny przeskok”). Ma on dla mnie trochę więcej sensu.

Poprzednio korzystaliśmy z polecenia route print, aby wyświetlić całą tablicę routingu. Do wyświetlenia tej tablicy w programie PowerShell służy polecenie cmdlet o nazwie Get-NetRoute:

```

Administrator: Windows PowerShell

PS C:\> Get-NetRoute

ifIndex DestinationPrefix NextHop
-----
6 255.255.255.255/32 0.0.0.0
5 255.255.255.255/32 0.0.0.0
1 255.255.255.255/32 0.0.0.0
6 224.0.0.0/4 0.0.0.0
5 224.0.0.0/4 0.0.0.0
1 224.0.0.0/4 0.0.0.0
5 192.168.17.0/24 10.10.10.1
1 127.255.255.255/32 0.0.0.0
1 127.0.0.1/32 0.0.0.0
1 127.0.0.0/8 0.0.0.0
5 10.10.10.255/32 0.0.0.0
5 10.10.10.13/32 0.0.0.0
5 10.10.10.0/24 0.0.0.0
6 0.0.0.0/0 1.1.1.1
6 ff00::/8 ::
5 ff00::/8 ::
1 ff00::/8 ::
6 fe80::1c58:5bf4:8b46:3559/128 ::
5 fe80::402:a7ae:81ac:e95b/128 ::
6 fe80::/64 ::
5 fe80::/64 ::
1 ::1/128 ::

```

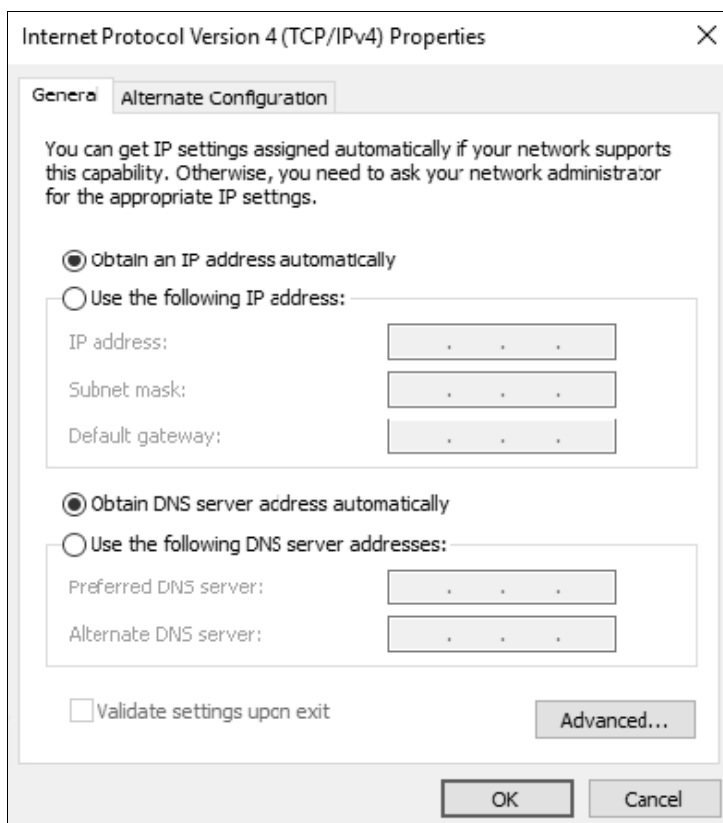
Grupowanie kart sieciowych

Przejdziemy teraz do innego zagadnienia związanego z siecią, które staje się coraz bardziej popularne w przypadku sprzętu serwerowego, a mianowicie *grupowania kart sieciowych* (ang. *NIC Teaming*). Grupowanie kart sieciowych polega zasadniczo na powiązaniu ze sobą dwóch lub więcej fizycznych interfejsów sieciowych w taki sposób, jakby były pojedynczym interfejsem sieciowym w systemie Windows. Umożliwia to podłączenie dwóch fizycznych przewodów do dwóch różnych portów przełącznika przy użyciu tych samych ustawień. Jeśli jeden port karty sieciowej, port przełącznika lub przewód połączeniowy ulegnie awarii, serwer będzie kontynuować swoją pracę i łączyć się z siecią, ponieważ grupowanie pozwala karcie sieciowej, która nadal działa, na obsługę ruchu sieciowego.

Samo grupowanie kart sieciowych nie jest niczym nowym, ponieważ istnieje w systemie operacyjnym Windows Server już od ponad 10 lat. Jednak wczesne wersje tego rozwiązania były problematyczne, a na podstawie własnego doświadczenia uważam, że Windows Server 2016 to najwcześniejszy system operacyjny, który większość pracowników IT uważa za wystarczająco stabilny, by można było w nim korzystać z grupowania kart sieciowych w środowisku produkcyjnym. W związku z tym to rozwiązanie jest wciąż stosunkowo świeże.

Aby rozpocząć grupowanie kart sieciowych, musisz się upewnić, że istnieje ich więcej w serwerze. W swoim komputerze mam obecnie cztery porty kart sieciowych. Chciałbym utworzyć dwa zespoły: moja pierwsza i druga karta sieciowa zostaną zgrupowane, aby stać się *zespołem sieci wewnętrznej*, a karta trzecia i czwarta staną się *zespołem sieci DMZ*. W ten sposób po obu stronach swojej sieci zapewnię na tym serwerze redundancję kart sieciowych.

Pierwszym działaniem, które należy wykonać, jest usunięcie wszelkich ustawień związanych z adresami IP, które mogą istnieć w kartach sieciowych. Gdy połączysz wiele kart sieciowych w jeden zespół, skonfigurujesz dla niego parametry adresów IP. Nie będziesz już zajmował się właściwościami poszczególnych kart w celu przypisania im adresów IP. Otwórz więc właściwości każdej karty sieciowej i upewnij się, że nie zawierają informacji o statycznym adresie IP, tak jak na poniższym zrzucie ekranu:

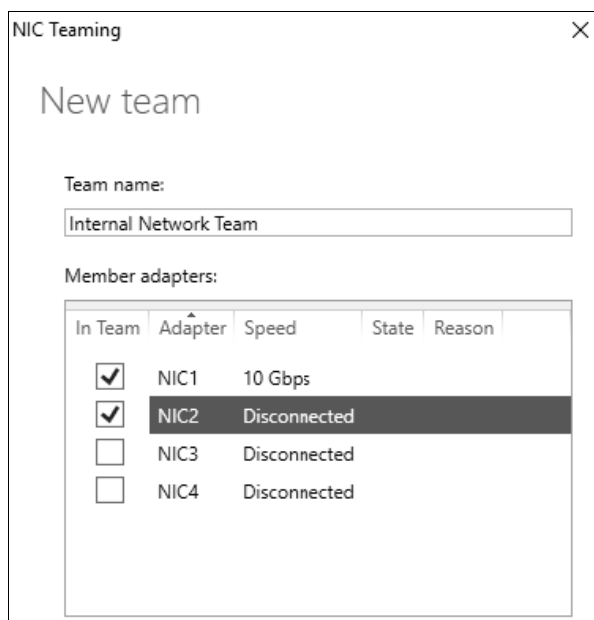


Teraz otwórz Menedżera serwera i kliknij łącze *Local Server (Serwer lokalny)*. Przeglądając informacje o właściwościach serwera, zobaczysz opis każdej karty sieciowej, a także opcję o nazwie *NIC Teaming (Zespół kart interfejsu sieciowego)*, która ma obecnie wartość *Disabled (Wyłączone)*:

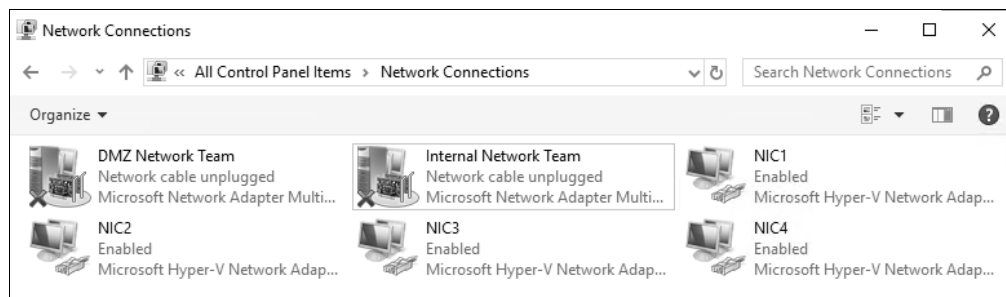
Windows Defender Firewall	Public: On
Remote management	Enabled
Remote Desktop	Disabled
NIC Teaming	Disabled
NIC1	IPv4 address assigned by DHCP, IPv6 enabled
NIC2	Not connected
NIC3	Not connected
NIC4	Not connected

Kliknij opcję *Disabled*, a następnie poszukaj sekcji zatytułowanej *Teams (Zespoły)*. Kliknij przycisk *Tasks (Zadania)* i wybierz opcję *New Team (Nowy zespół)*.

Nadaj nowemu zespołowi odpowiednią nazwę i wybierz karty sieciowe, które mają być częścią tego zespołu. Możesz wykonać te same kroki tyle razy, ile potrzebujesz, aby utworzyć dodatkowe zespoły z pozostałymi kartami sieciowymi:



Gdy zakończysz działania, Twoje zespoły zostaną wyświetlone w Menedżerze serwera. Jeśli w Panelu sterowania otworzysz okno *Network Connections (Połączenia sieciowe)*, zobaczysz, że oprócz czterech fizycznych kart sieciowych będziesz miał dwa nowe elementy, które odpowiadają konfiguracjom naszych nowych zespołów. Mogę kliknąć prawym przyciskiem myszy każdy z tych zespołów i skonfigurować parametry związane z adresem IP, podobnie jak zrobiłbym to dla pojedynczej karty sieciowej. Właściwości związane z adresem IP będą obowiązywać dla wszystkich kart sieciowych wchodzących w skład zespołu:



Programowalna sieć komputerowa

Nie można zaprzeczyć, że przetwarzanie w chmurze charakteryzuje się wysokim poziomem uniwersalności i elastyczności i że większość technicznej kadry kierowniczej testuje obecnie możliwości wykorzystania technologii chmurowych. Na przeszkodzie w ich szerszym użyciu stoi to, że trzeba zapewnić odpowiedni poziom zaufania. Usługi w chmurze udostępniają ogromną moc obliczeniową, a oprócz tego są osiągalne natychmiast po naciśnięciu jednego przycisku. Aby firmy mogły przechowywać swoje dane w tych systemach, poziom zaufania organizacji do dostawcy chmury musi być bardzo wysoki. Używając chmury, nie posiadasz przecież żadnej infrastruktury sprzętowej ani sieciowej, w której są przechowywane Twoje dane, więc Twoja kontrola nad tymi zasobami jest w najlepszym razie ograniczona. Widząc tę przeszkodę, firma Microsoft dołożyła wielu starań, by wprowadzić najnowsze technologie chmurowe do lokalnego centrum danych. Wysoki poziom uniwersalności serwerów w naszych centrach danych oznacza wirtualizację. Wirtualizacja serwerów trwa już od wielu lat, a jej możliwości są ciągle ulepszone. Teraz gdy mamy możliwość tak łatwego uruchamiania nowych serwerów przy użyciu technologii wirtualizacji, wygląda na to, że następną przeszkodą do pokonania będzie zdolność do łatwego przenoszenia tych wirtualnych serwerów w dowolnym czasie do dowolnego miejsca.

Czy masz serwer, który chciałbyś przenieść do innego centrum danych w Twoim kraju? A może zastanawiasz się nad przeniesieniem całego centrum danych do nowej kolokacji w innym mieście? Być może niedawno nabyłeś nową firmę i musisz umieścić jej infrastrukturę w swojej sieci, ale okazuje się, że pewne konfiguracje nakładają się na siebie. Czy kupiłeś już miejsce u dostawcy usług w chmurze, a teraz próbujesz uporządkować bałagan związany z planowaniem migracji wszystkich serwerów do tej chmury? Oto pytania wymagające odpowiedzi, którą jest programowalna sieć komputerowa.

Programowalna sieć komputerowa (SDN) jest szerokim, ogólnym terminem, który oznacza wiele technologii współpracujących ze sobą w celu uzyskania wymaganej funkcjonalności. Celem tego rozwiązania jest rozszerzenie granic sieci, bez względu na to, gdzie się znajdujesz i kiedy żądasz jakichś usług. Rzućmy okiem na niektóre składniki dostępne w systemie Windows Server 2019, które działając wspólnie, tworzą wirtualne środowisko sieciowe — pierwszy krok w procesie realizacji sieci zdefiniowanej programowo.

Wirtualizacja sieci Hyper-V

Najważniejszy komponent, który pozwala na analizę sieci i przemieszczanie jej do warstwy wirtualizacji, znajduje się w systemie Hyper-V. Ma to sens, ponieważ jest to ten sam element, który służy do wirtualizacji serwerów. Dzięki wirtualizacji sieci Hyper-V separujemy sieci wirtualne i fizyczne. Podczas definiowania nowych sieci wirtualnych nie trzeba już uwzględniać ograniczeń schematu IP w sieci fizycznej, ponieważ mogą one działać poprawnie nawet wówczas, gdy konfiguracje dwóch różnych sieci byłyby w tradycyjnych warunkach niezgodne ze sobą.

Ta koncepcja jest trochę trudna do zrozumienia, jeśli po raz pierwszy o niej słyszysz, więc za chwilę omówimy kilka rzeczywistych sytuacji, w których można by skorzystać z tego rodzaju separacji.

Chmury prywatne

Prywatne chmury pojawiają się w centrach danych na całym świecie, ponieważ ich tworzenie ma sens. Każdy administrator, który chciałby skorzystać z zalet chmury w swoim środowisku, jednocześnie unikając jej wad, może zastosować to rozwiązanie. Utworzenie chmury prywatnej daje możliwość dynamicznego powiększania i zmniejszania zasobów obliczeniowych oraz obsługiwanie wielu dzierżawców lub działów w ramach tej samej infrastruktury. Interfejsy zarządzające mogą być obsługiwane bezpośrednio w tych działach, dzięki czemu sam dzierżawca realizuje drobne prace konfiguracyjne, a administrator nie musi tracić czasu i zasobów na wykonywanie na poziomie dostawcy infrastruktury niewielkich, ale złożonych zadań.

Chmury prywatne umożliwiają korzystanie ze wszystkich funkcji chmury publicznej, bez obawy o poziom prywatności danych przechowywanych u zewnętrznego dostawcy usługi, nad którymi nie można mieć wówczas realnej kontroli.

Gdy w lokalnej infrastrukturze należy wdrożyć prywatną chmurę, szczególnie taką, w której chcesz zapewnić obsługę wielu dzierżawców, warto (a nawet trzeba) skorzystać z zalet wirtualizacji sieci. Załóżmy, że udostępniasz zasoby komputerowe dwóm oddziałom firmy, a każdy z nich ma własne potrzeby hostingu niektórych serwerów WWW. Nie wygląda to na wielkie wyzwanie, ale problemem jest to, że te oddziały mają zespoły administracyjne, które chcą korzystać ze schematów IP w zakresie 10.0.0.0. Oba muszą mieć możliwość korzystania z tych samych adresów IP w tej samej sieci szkieletowej, którą udostępniasz, a cały ruch musi zostać całkowicie rozdzielony i odseparowany. Wymagania te byłyby niemożliwe do spełnienia w tradycyjnej sieci fizycznej, ale wirtualizacja sieci pozwala w prosty sposób każdemu z działów przydzielić podsieci IP i dowolne schematy adresowania. Dzięki temu można uruchamiać serwery w dowolnych podsieciach i adresacjach IP, a cały ruch jest w pełni hermetyzowany, dzięki czemu pozostaje odseparowany i całkowicie „nieświadomy” pozostałego ruchu w tej samej fizycznej sieci szkieletowej, która działa pod warstwą wirtualizacji. Ten scenariusz dobrze się również sprawdza w przypadku przejęć korporacyjnych. Dwie firmy, które łączą siły na poziomie IT, często mają konflikty dotyczące domen i podsieci. Dzięki wirtualizacji sieci można zachować istniejącą konfigurację infrastruktury oraz serwerów, jednocześnie dodając je do tej samej sieci fizycznej dzięki zastosowaniu wirtualizacji sieci Hyper-V.

Innym, prostszym przykładem jest sytuacja, w której po prostu chcesz przenieść serwer w sieci korporacyjnej. Być może masz starszy serwer biznesowy, do którego wciąż wielu pracowników potrzebuje dostępu, ponieważ musi używać aplikacji branżowej, która działa przez cały czas. Problem z przenoszeniem serwera polega na tym, że aplikacja branżowa na komputerach klientów zawiera statyczny adres IPv4 skonfigurowany w celu komunikacji z serwerem. Gdy użytkownik otwiera aplikację, powoduje *nawiązanie łączności z serwerem* o adresie 10.10.10.10. Tradycyjnie mógłby się tu pojawić duży problem, gdyż przeniesienie tego serwera z obecnego centrum danych do nowej lokalizacji oznaczałoby zmianę adresu IP, a to z kolei uniemożliwiłoby wszystkim dostęp do niego. W przypadku sieci wirtualnych nie stanowi to żadnego kłopotu. Dzięki możliwości kontrolowania ruchu sieciowego i podsieci IP w warstwie wirtualizacji serwer ten można przenieść z Warszawy do Krakowa i zachować jego wszystkie ustawienia dotyczące adresu IP, ponieważ działająca pod spodem sieć fizyczna nie ma żadnego znaczenia. Zanim dane zostaną wysłane przez sieć fizyczną, nastąpi ich hermetyzacja, dzięki czemu adres IP starszego serwera może pozostać niezmienny, a on sam może być bez żadnych problemów przenoszony do dowolnego miejsca w Twoim środowisku.

Chmury hybrydowe

Chociaż zwiększenie uniwersalności sieci korporacyjnych jest już ogromną korzyścią, możliwości udostępniane przez wirtualizację sieci rosną wykładniczo, gdy decydujesz się skorzystać z prawdziwych zasobów chmurowych. Kiedy podejmiesz decyzję o przeniesieniu części zasobów, która powinna być zarządzana przez dostawcę usług w chmurze publicznej, prawdopodobnie uruchomisz środowisko chmury hybrydowej. Oznacza to, że zdefiniujesz niektóre usługi w chmurze, ale zachowasz także niektóre serwery i usługi w lokalnej infrastrukturze. Przewiduję, że w przypadku większości firm scenariusz chmury hybrydowej będzie realizowany bez końca, ponieważ stuprocentowe przejście do chmury publicznej jest po prostu niemożliwe, biorąc pod uwagę sposoby, w jakie wiele przedsiębiorstw prowadzi działalność. Gdybyś chciał skonfigurować chmurę hybrydową, ponownie należałoby przeanalizować wszelkie problemy związane z przepływem zasobów między Twoimi sieciami fizycznymi i chmurowymi. Gdy chcesz przenieść lokalny serwer do chmury, musisz tak dostosować jego parametry, aby konfiguracja sieci była zgodna z infrastrukturą chmury. Czy więc nie musisz ponownie konfigurować karty sieciowej na swoim serwerze, aby pasowała do podsieci działającej w mojej sieci w chmurze? Nie, jeśli masz uruchomioną infrastrukturę wirtualizacji sieci. Po raz kolejny użycie sieci definiowanej programowo oszczędza czas dzięki temu, że daje nam możliwość zachowania informacji o adresach IP na serwerach, które są przenoszone, ponieważ po prostu będą one używały tej samej adresacji w chmurze. Jak już wspominałem, ze względu na hermetyzację ruchu fizyczna sieć zapewniana przez chmurę nie musi być zgodna z naszą siecią wirtualną, a to daje nam możliwość płynnego przenoszenia serwerów z infrastruktury lokalnej do chmury (i odwrotnie) bez konieczności spełnienia specjalnych wymagań dla sieci.

Jak działa programowalna sieć komputerowa?

Na razie brzmi to jak magia. Jak to wszystko faktycznie działa i jakie elementy muszą ze sobą współpracować, aby wirtualizacja sieci stała się rzeczywistością w organizacji? Coś tak złożonego z pewnością ma wiele składników i nie można tego włączyć po prostu przez naciśnięcie przycisku. W sieci, która została zwirtualizowana, funkcjonują różne technologie i komponenty. Zapre-

zentuję je za chwilę, dzięki czemu lepiej zrozumiesz działanie odpowiednich elementów i terminologię, z którą będziesz mieć do czynienia po rozpoczęciu pracy z sieciami programowalnymi.

System Center Virtual Machine Manager

Zestaw narzędzi Microsoft System Center (a w szczególności komponent **Virtual Machine Manager** — VMM) jest kluczowym elementem układanki tworzącej model sieci zdefiniowanej programowo. Zdolność do pobierania adresów IP i przenoszenia ich do innych lokalizacji wymaga pewnej koordynacji urządzeń sieciowych, a VMM jest w stanie w tym pomóc. Ten komponent jest centralnym punktem zarządzania podczas definiowania i konfigurowania sieci wirtualnych. Sam System Center to ogromne zagadnienie z wieloma opcjami, dlatego nie zostanie ono omówione w tej książce. Podaję jednak odpowiednie łącze, które może służyć jako punkt wyjścia do zapoznania się z komponentem VMM: [https://docs.microsoft.com/en-us/previous-versions/system-center/system-center-2012-R2/gg610610\(v=sc.12\)](https://docs.microsoft.com/en-us/previous-versions/system-center/system-center-2012-R2/gg610610(v=sc.12)).

Rola Network controller

Rola *Network controller* (*Kontroler sieci*) została początkowo wprowadzona w systemie Windows Server 2016 i jak sama nazwa wskazuje, służy ona do kontroli zasobów sieciowych w organizacji. W większości przypadków ta rola będzie współpracować z VMM, aby zarządzanie konfiguracjami sieci było jak najbardziej scentralizowane i bezproblemowe. Kontroler sieci to samodzielna rola, którą można zainstalować w systemie Server 2016 lub 2019 i do której następnie można uzyskać bezpośredni dostęp bez konieczności użycia VMM. Nie przewiduję jednak wielu wdrożeń, które by stosowały taką konfigurację. Korzystanie z roli *Network controller* jest możliwe przez użycie odpowiednich interfejsów API PowerShell, ale jeszcze lepszy rezultat zapewni dodanie interfejsu graficznego, z którego konfigurujesz nowe sieci, monitorujesz istniejące urządzenia lub rozwiązujesz problemy w modelu wirtualnym. Interfejs graficzny, którego można użyć, to System Center VMM.

Kontroler sieci może służyć do konfigurowania wielu różnych aspektów sieci wirtualnych i fizycznych. Dzięki niemu można skonfigurować podsieci i adresy IP, parametry sieci VLAN na przełącznikach Hyper-V, a nawet użyć go do zdefiniowania kart sieciowych na maszynach wirtualnych. Kontroler sieci pozwala także na tworzenie reguł związanych z **listą kontroli dostępu (ACL)** w przełączniku Hyper-V i zarządzanie nimi, dzięki czemu na tym poziomie można zdefiniować własne rozwiązanie zapory bez konieczności konfigurowania lokalnych zapór na samych maszynach wirtualnych lub posiadania specjalizowanego sprzętu. Kontroler sieci może nawet służyć do konfigurowania równoważenia obciążenia i zapewnienia dostępu VPN za pośrednictwem serwerów RRAS.

Protokół Generic Routing Encapsulation

Generic Routing Encapsulation (GRE) jest po prostu protokołem tunelowania, jednakże niezbędnym, aby wirtualizacja sieci przebiegła pomyślnie. Gdy wcześniej omawialiśmy przenoszenie podsieci IP oraz definiowanie sieci wirtualnych ponad sieciami fizycznymi bez konieczności zapewniania kompatybilnych konfiguracji IP powinienem był dodać, że cała ta funkcjonalność jest zapewniana przez użycie protokołu GRE. Gdy Twoja sieć fizyczna ma adres 192.168.0.x, ale w centrum danych chciałbyś zarządzać niektórymi maszynami wirtualnymi

z innej podsieci, możesz bez problemu utworzyć sieć wirtualną 10.10.10.x, jednakże aby wszystko działało, pakiety danych muszą być w stanie poruszać się po fizycznej sieci 192.168. W tym momencie zaczyna odgrywać rolę hermetyzacja routingu. Wszystkie pakiety z sieci 10.10.10.x są hermetyzowane przed wysłaniem ich przez fizyczną sieć 192.168.0.x.

Istnieją dwa różne, specyficzne protokoły hermetyzacji routingu, które są obsługiwane w środowisku wirtualizacji sieci Microsoft Hyper-V. We wcześniejszych wersjach środowiska Windows Server mogliśmy się skoncentrować wyłącznie na standardzie **Network Virtualization Generic Routing Encapsulation (NVGRE)**, ponieważ był to jedyny protokół wykorzystywany przez system Windows do wirtualizacji sieci. Od dłuższego czasu istnieje jednak inny protokół, zwany **Virtual Extensible Local Area Network (VXLAN)**, dlatego wiele przełączników sieciowych (szczególnie firmy Cisco) częściej obsługuje VXLAN niż NVGRE. Tak więc w przypadku nowych platform wirtualizacji sieci, poczynając od systemu Windows Server 2016, jesteśmy w stanie obsługiwać protokoły NVGRE lub VXLAN w zależności od tego, co najlepiej odpowiada potrzebom przedsiębiorstwa.

Nie musisz rozumieć, w jaki sposób działają protokoły GRE, aby sprawić, by wykonywały swoją pracę, ponieważ zostaną one skonfigurowane za pomocą narzędzi istniejących w stosie wirtualizacji sieci Hyper-V. Ważne jest jednak, aby biorąc pod uwagę koncepcję wirtualnego środowiska sieciowego, zrozumieć, że za sprawą protokołu GRE wszystkie elementy współpracują ze sobą i poprawnie działają.

Usługa Microsoft Azure Virtual Network

Gdy w środowisku korporacyjnym uruchomisz już wirtualizację sieci Hyper-V i poczujesz się komfortowo dzięki oddzieleniu sieci fizycznych od wirtualnych, najprawdopodobniej będziesz także chciał zbadać możliwości interakcji z sieciami dostawców usług w chmurze. Korzystając z Microsoft Azure jako dostawcy usług w chmurze, możesz zbudować hybrydowe środowisko chmurowe, które łączy lokalne sieci fizyczne ze zdalnymi sieciami wirtualnymi umieszczonymi na platformie Azure. Sieć wirtualna platformy Azure to składnik, który umożliwia wprowadzanie własnych adresów IP i podsieci do chmury. Oto miejsce, w którym możesz uzyskać więcej informacji (a nawet dostać możliwość bezpłatnego przetestowania wirtualnej sieci platformy Azure): <https://azure.microsoft.com/en-us/services/virtual-network/>.

Brama Windows Server Gateway (SDN Gateway)

Podczas pracy z sieciami fizycznymi, wirtualnymi i takimi, które są przechowywane w środowiskach chmurowych, potrzebujesz jakiegoś komponentu umożliwiającego sieciom interakcję i komunikację między sobą. Tu właśnie wchodzi w grę brama **Windows Server Gateway** (zwana również **SDN Gateway**). Windows Server Gateway to nowszy termin. Poprzednio używano (i czasami nadal tak się robi) nazwy Hyper-V Network Virtualization Gateway, więc możesz ją spotkać w niektórych dokumentacjach. Cel bramy Windows Server Gateway jest dość prosty: ma łączyć sieci wirtualne i fizyczne. Sieci wirtualne mogą się znajdować w środowisku lokalnym lub w chmurze. W każdym z tych przypadków, aby połączyć się z siecią, trzeba zastosować bramę Windows Server Gateway. Podczas tworzenia pomostu między lokalną infrastrukturą a chmurą dostawcy usług w chmurze będzie po swojej stronie korzystał z bramy, do której można się podłączyć z sieci fizycznej za pośrednictwem tunelu VPN.

Windows Server Gateway jest, ogólnie rzecz biorąc, maszyną wirtualną, zintegrowaną z wirtualizacją sieci Hyper-V. Pojedynczej bramy można użyć do kierowania ruchem w przypadku różnych klientów, dzierżawców lub oddziałów. Chociaż ci różni klienci mają oddzielne sieci, które muszą być odseparowane od siebie, dostawca chmury (publicznej lub prywatnej) może nadal wykorzystywać pojedynczą bramę do zarządzania całym ruchem, ponieważ bramy pozwalają zapewnić całkowitą izolację między poszczególnymi strumieniami danych.

Brama Windows Server Gateway istniała już w systemie Server 2016, ale po jej wdrożeniu wykryto pewne problemy związane z wydajnością, które ograniczają przepustowość ruchu sieciowego. Te ograniczenia zostały znacznie zmniejszone w systemie Windows Server 2019, co oznacza, że jedna brama może teraz obsługiwać szerszy zakres ruchu i większą liczbę dzierżawców.

Szyfrowanie sieci wirtualnej

Zespoły zarządzania bezpieczeństwem nieustannie dbają o szyfrowanie danych. Niezależnie od tego, czy dane są przechowywane czy przesyłane, należy się upewnić, że są odpowiednio zabezpieczone i chronione przed nieuprawnioną modyfikacją. W czasach przed wersją Server 2019 szyfrowanie ruchu wewnątrz sieci podczas przesyłania danych było generalnie zadaniem samej aplikacji, a nie sieci. Jeśli Twoje oprogramowanie ma możliwość szyfrowania danych przepływających między klientem a serwerem lub między serwerem aplikacji a serwerem bazy danych, to świetnie! Jeśli jednak aplikacja nie ma wbudowanych funkcji szyfrowania, prawdopodobnie komunikacja między klientem a serwerem odbywa się w postaci jawnego tekstu. Nawet w przypadku aplikacji, które faktycznie szyfrują dane, algorytmy szyfrów są czasami łamane i narażane na szwank. W przyszłości wraz z odkryciem nowych luk w zabezpieczeniach można mieć tylko nadzieję, że sposób, w jaki aplikacja chroni ruch, może zostać zaktualizowany w celu obsługi nowszych i lepszych metod szyfrowania.

Na szczęście Windows Server 2019 udostępnia nową funkcjonalność w ramach sieci definiowanych programowo. Nazywa się ona **szyfrowaniem sieci wirtualnej** i realizuje dokładnie to, co sugeruje jej nazwa. Gdy dane są przesyłane między maszynami wirtualnymi a serwerami Hyper-V (w tej samej sieci), całe podsieci mogą zostać przeznaczone do zaszyfrowania, co oznacza, że ruch w nich jest automatycznie szyfrowany na poziomie sieci wirtualnej. Aby można było skorzystać z tego rozwiązania, serwery VM i aplikacje działające na nich nie muszą być w żaden sposób konfigurowane ani zmieniane, ponieważ to sama sieć automatycznie szyfruje cały odbywający się w niej ruch.

Za pomocą programowalnej sieci komputerowej systemu Server 2019 każda podsieć w sieci wirtualnej może zostać oznakowana w celu szyfrowania, włącznie ze zdefiniowaniem odpowiedniego certyfikatu. Jeśli kiedyś obecne standardy szyfrowania staną się nieaktualne lub niepewne, programowalna sieć komputerowa będzie mogła zostać odpowiednio zaktualizowana, aby spełniała nowe wymagania. Odpowiednie podsieci nadal będą szyfrowane przy użyciu nowych metod bez konieczności wprowadzania jakichkolwiek zmian w maszynach wirtualnych lub aplikacjach. Jeśli w swoim środowisku używasz programowalnej sieci komputerowej i sieci wirtualnych, włączenie ich szyfrowania nie jest żadnym problemem!

Łączenie sieci lokalnej z usługą Azure

Większość firm, które zarządzają swoimi serwerami poprzez usługę Microsoft Azure, nadal ma fizyczne, lokalne sieci. Jedno z podstawowych pytań, na które zawsze należy odpowiedzieć, brzmi: „W jaki sposób będziemy mogli połączyć swoje fizyczne centrum danych z centrum danych Azure?”. Zazwyczaj firmy wybierają w tym celu jedną z dwóch różnych metod. Możesz wdrożyć serwery bramy na brzegu sieci lokalnej i platformy Azure oraz połączyć je za pomocą sieci VPN typu lokacja-lokacja. Dzięki temu uzyskujemy stały tunel między dwiema sieciami. Alternatywnie Microsoft zapewnia usługę o nazwie Azure Express Route, która wykonuje to samo działanie — tworzy stały tunel między siecią fizyczną a siecią wirtualną platformy Azure. Każde z tych rozwiązań po skonfigurowaniu działa świetnie, ale małe organizacje, mające tylko kilka lokalnych serwerów, które muszą zostać podłączone do chmury Azure, mogą je uznać za zbyt skomplikowane.

Azure Network Adapter

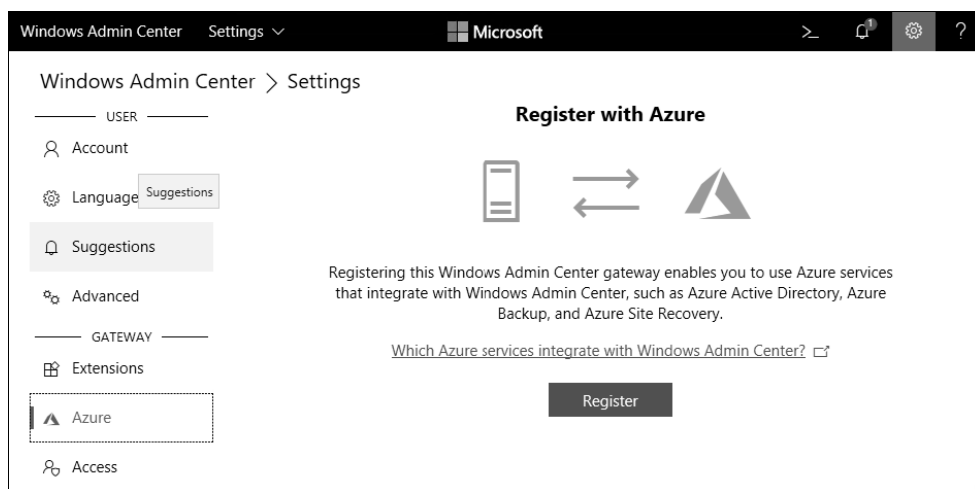
Jeśli masz lokalny serwer, który należy szybko połączyć ze środowiskiem platformy Azure (a nie zdefiniowałeś jeszcze stałego połączenia między lokalizacją fizyczną a tym środowiskiem), możesz w chmurze hybrydowej skorzystać z zupełnie nowej funkcji zwanej **Azure Network Adapter** (karta sieciowa platformy Azure). Aby użyć takiej karty sieciowej, musisz wykorzystać nową platformę Windows Admin Center, służącą do zarządzania serwerami.

Za pomocą aplikacji Windows Admin Center można szybko dodać kartę Azure Network Adapter do lokalnego serwera, który połączy się bezpośrednio z siecią platformy Azure za pomocą połączenia VPN typu punkt-lokacja. Super!

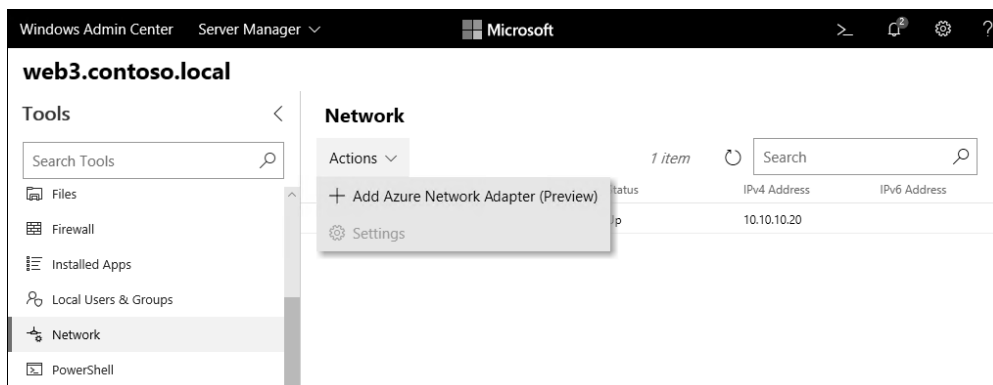
Co więcej, ta funkcjonalność została wdrożona również w starszych wersjach systemu Windows Server, dzięki czemu możesz dodawać karty sieciowe nie tylko w komputerach z systemem Server 2019, ale także tych, które mają zainstalowane wersje Server 2016 i Server 2012 R2.

Aby to było możliwe, należy spełnić kilka wymagań: musisz mieć aktywną subskrypcję platformy Azure i skonfigurować co najmniej jedną sieć Azure Virtual Network.

Następnie musisz zarejestrować swoją aplikację Windows Admin Center na platformie Azure. Aby to zrobić, możesz otworzyć *Windows Admin Center*, a następnie przejść do opcji *Settings* (*Ustawienia*). W dalszej kolejności klikasz łącze *Azure* w grupie *Gateway* (*Brama*) i przechodzisz przez proces rejestracji:



Gdy na platformie Azure zarejestrowałeś już swoją aplikację Windows Admin Center, za jej pomocą otwórz serwer, którym zarządzasz, a następnie przejdź do sekcji *Network* (Sieć). Pojawi się lista wszystkich kart sieciowych, które są dostępne w Twoim serwerze, a w górnej części okna będzie widnieć menu rozwijane *Actions* (Akcje). Wewnątrz tego menu wybierz opcję *Add Azure Network Adapter* (Dodaj kartę sieciową Azure).



Przekonasz się, że wszystkie dane wymagane przez platformę Azure do nawiązania połączenia zostały już automatycznie uzupełnione na podstawie parametrów Twojej sieci i subskrypcji. Jeśli nie masz jeszcze sieci Azure Virtual Network, ten kreator może nawet ją utworzyć. Będziesz miał również możliwość zdefiniowania własnego certyfikatu w celu uwierzytelnienia tworzonego połączenia. Takie działanie będzie dobrym pomysłem, jeśli planujesz używać długoterminowego połączenia z platformą Azure. W przeciwnym razie możesz pozwolić aplikacji Windows Admin Center i platformie Azure na wygenerowanie certyfikatu z podpisem własnym i po prostu kliknąć przycisk *Create* (Utwórz). Windows Admin Center uruchomi połączenie między serwerem lokalnym a wirtualną siecią platformy Azure. To tylko kilka kliknięć myszą! Oto wspaniała metoda prostego i szybkiego tworzenia połączeń *ad hoc* między serwerami a platformą Azure.

Jeśli później będziesz musiał odłączyć serwer od sieci platformy Azure, możesz otworzyć okno połączeń sieciowych, tak jak zrobiłeś podczas próby modyfikacji właściwości karty sieciowej, a przekonasz się, że aplikacja Windows Admin Center po prostu skonfigurowała połączenie VPN typu punkt-lokacja, które pojawia się obecnie na liście *Network Connections* (*Połączenia sieciowe*). Możesz kliknąć prawym przyciskiem myszy połączenie Azure VPN i rozłączyć je.

Podsumowanie

Zadania administrowania serwerami i sieciami były w większości organizacji dość wyraźnie rozdzielone, ale z czasem te obszary zaczęły się pokrywać. Istnieje wiele konfiguracji i zadań związanych z siecią, które muszą być teraz wykonywane przez administratorów systemu Windows Server bez angażowania zespołu administratorów sieciowych, dlatego ważne jest, abyś dobrze rozumiał, jak działa Twoja infrastruktura. Znajomość narzędzi przedstawionych w tym rozdziale umożliwi konfigurowanie, monitorowanie i rozwiązywanie problemów w przypadku większości sieci zorientowanych na produkty firmy Microsoft.

Wprowadzenie do sieci definiowanej programowo może się okazać dość trudnym podrozdziałem, jeśli nigdy wcześniej nie spotkałeś się z takim rozwiązaniem. Mam jednak nadzieję, że zachęci Cię to do zainteresowania się tym tematem, dzięki czemu będziesz mógł sobie z nim poradzić w przyszłości. Przetwarzanie w chmurze będzie już cały czas dostępne niezależnie od tego, czy jesteś na nie gotowy. Sieci lokalne firmy Microsoft potrafią w różny sposób współpracować z usługą Microsoft Azure, dlatego wkrótce pracownicy działu IT będą musieli się zapoznać z tymi koncepcjami. Idea programowalnej sieci komputerowej zyska na popularności w nadchodzących latach. Obecnie może się ona wydawać trochę zniechęcająca, ale za kilka lat będziemy się dziwić, w jaki sposób wszystko kiedyś mogło funkcjonować bez sieci wirtualnych. Znacznie więcej informacji o takich sieciach znajduje się zarówno w witrynie Microsoft Docs, jak też w opublikowanych książkach na temat wirtualnej sieci Hyper-V oraz platformy System Center Virtual Machine Manager. Warto, abyś dogłębniej zapoznał się z tymi materiałami, jeśli jesteś zainteresowany przetestowaniem nowych rozwiązań. W następnym rozdziale zajmiemy się użyciem opcji zdalnego dostępu.

Pytania

1. Ile bitów ma adres IPv6?
2. Zapisz następujący adres IPv6 w formie skróconej:
2001:ABCD:0001:0002:0000:0000:0000:0001.
3. Jak nazywa się polecenie podobne do `tracert`, ale wyświetlające lokalną kartę sieciową, z której są wysyłane dane?
4. Na serwerze z wieloma kartami sieciowymi można wprowadzić adres bramy domyślnej dla każdej z tych kart — prawda czy fałsz?

5. Jak nazywa się polecenie cmdlet programu PowerShell, którego można używać do tworzenia nowych tras w systemie Windows Server?
6. Jakich wersji systemów operacyjnych Windows Server można używać z kartą Azure Network Adapter w celu połączenia serwerów bezpośrednio z wirtualnymi sieciami platformy Azure?

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Serwery Windows królują w przestrzeni centrów danych. Nawet po przeniesieniu zasobów do chmury Azure dane są nadal zarządzane przez Windows Server: właśnie ten system operacyjny stanowi podstawę platformy Azure! Windows Server 2019 może obsłużyć nawet najpoważniejsze zadania w środowisku chmury. Microsoft konsekwentnie zmniejsza rozmiar platformy obliczeniowej w serwerach i tworzy nowe sposoby komunikacji z nimi. Także technologia kontenerów bardzo się rozwinęła: Server 2019 przenosi aplikacje do kontenerów, dzięki czemu uruchamia się je w odizolowaniu od siebie i na masową skalę. Co więcej, nowe narzędzia dla administratorów sprawiają, że z systemem Windows Server 2019 pracuje się efektywnie i przyjemnie.

W tej książce znalazły się wszystkie informacje potrzebne do wdrożenia i wykorzystywania Windows Server 2019 LTSC. Omówiono zagadnienia związane z jego instalacją oraz z administrowaniem tym systemem. Sporo miejsca poświęcono scentralizowanemu zarządzaniu, monitorowaniu i konfiguracji serwerów. Opisano Menedżer serwera, język skryptowy PowerShell, a także nową aplikację Windows Admin Center. Szczegółowo zaprezentowano również kontenery i Nano Server, które są związane z kanałem półrocznym (SAC) platformy serwerowej. Ponadto przedstawiono różne technologie zdalnego dostępu, które można wykorzystać w tym systemie operacyjnym, a także wytyczne dotyczące wirtualizacji centrum danych za pomocą funkcji Hyper-V.

W książce między innymi:

- instalowanie systemu Windows Server 2019 i podstawy zarządzania nim
- obsługa sieci i infrastruktura, MMC i MSC
- bezpieczeństwo systemu Windows Server
- wydajna redundancja danych i obliczeń i Spaces Storage Direct
- narzędzia: Windows Server Containers, Hyper-V, Docker i Kubernetes

Jordan Krause — wielokrotnie otrzymywał tytuł MVP. Zawodowo zajmuje się sieciami Microsoft i technologiami zdalnego dostępu. Specjalizuje się w technologiach Microsoft DirectAccess i Always On VPN i ciągle poszerza swoją wiedzę. Posiada liczne certyfikaty Microsoftu: MCP, MCTS, MCSA i MCITP Enterprise Administrator. Regularnie publikuje artykuły na temat tych technologii. Mieszka w zachodniej części stanu Michigan w USA.

Helion helion.pl 0 801 339900 0 601 339900	<i>Sprawdź nasze szkolenia!</i> SZKOLENIA  AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL	KOD KORZYŚCI Sięgnij po więcej!   ISBN 978-83-283-6485-1  9 788328 364851
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 89,00 zł

Packt