29 Zabezpieczenie dostępu do urządzeń zdalnych (w tym mobilnych)

A. Zabezpieczenie dostępu do urządzeń zdalnych

Przygotowanie

Enable-PSRemoting -Force -SkipNetworkProfileCheck

New-LocalUser -Name "Admin02" -Password (ConvertTo-SecureString zaq1@WSX -AsPlainText -Force)

Set-LocalUser -Name "Admin02" -PasswordNeverExpires $true

Zapoznaj się z opisem działania i wykonaj skrypt oraz wykonaj notatkę, zaprezentuj i omów wykonane czynności.

Przetestujemy dwie metody poprawy bezpieczeństwa komunikacji zdalnej PowerShell.

Pierwsza to tymczasowe zwiększenie uprawnień administratora lokalnego.

Druga polega na automatycznej zmianie hasła w skrypcie.

1. Tymczasowe zwiększenie przywilejów
2. Automatyzacja resetowania hasła
3. Łączenie wszystkiego w skrypt PowerShell

Jednym z aspektów PowerShell, jest uruchamianie zdalnych zadań na wielu komputerach jednocześnie. Korzystając z polecenia Invoke-Command, administratorzy mogą łatwo i szybko wykonywać zadania na zdalnych komputerach, uruchamiając go z jednego komputera na wielu naraz.

W moim przykładowym scenariuszu chcę uruchomić skrypt, który używa polecenia Invoke-Command do ponownego uruchomienia usługi Windows Update na komputerach zdalnych. Oto część skryptu, która to robi:

```
Invoke-Command -ComputerName Test-1,Test-2 -Credential $Credential -ScriptBlock {
    Restart-Service wuauserv
}
```

do wysłania spisu działających i niedziałających usług do pliku C:\uslugi.txt

```
Invoke-Command -Session $s -ScriptBlock { Get-Service | FT -GroupBy Status | Out-File C:\uslugi.txt}
```

Dodajmy teraz trochę bezpieczeństwa do tego skryptu.

1. Tymczasowe zwiększenie uprawnień

Ponowne uruchamianie usług wymaga dostępu administratora lokalnego, więc konto, którego użyjemy do zdalnej obsługi tych komputera/ów, musi to mieć. Wiele razy administratorzy systemu po prostu utrzymują konto w lokalnej grupie administratorów przez cały czas i używają go w razie potrzeby.

To zła praktyka. Jeśli to konto zostanie naruszone, ma teraz dostęp do dowolnego komputera, który ma to konto w lokalnej grupie administratorów. Z tego powodu, należy użyć programu PowerShell do zwiększenia uprawnień wymaganych dla tego konta po uruchomieniu skryptu, a następnie usunąć je po zakończeniu działania skryptu.

Jednym ze sposobów użycia tej metody jest posiadanie grupy zabezpieczeń, w tym przypadku Administratorzy. Dodamy tę grupę do grupy administratorów lokalnych na komputerach, na których chcemy zainstalować oprogramowanie, dając w ten sposób każdemu kontu w tej grupie dostęp administratora lokalnego do komputerów zdalnych. Użyje konta (Admin02).

Po wykonaniu polecenia Invoke-Command usuniemy Admin02 z grupy zabezpieczeń Administratorzy, co spowoduje usunięcie uprawnień administratora lokalnego dla tego konta. Dzięki tej metodzie używane konto ma tylko uprawnienia administratora lokalnego podczas wykonywania skryptu. Oznacza to, że jeśli konto zostanie naruszone, osoba atakująca ma znacznie mniejszą szansę na wykorzystanie go do wyrządzenia szkody.

Aby wprowadzić te zmiany, w skrypcie używamy cmdletów Add-LocalGroupMember i Remove-LocalGroupMember.

```
Add-LocalGroupMember -Group "Administratorzy" -Member Admin02

$members = "Admin02"

Remove-LocalGroupMember -Group "Administratorzy" -Member $members –Confirm:$False
```

Należy pamiętać, że konto wykonujące skrypt będzie wymagało uprawnień do dodawania/usuwania użytkowników w grupach zabezpieczeń, resetowania hasła.

2. Automatyczne resetowanie hasła

Innym ważnym aspektem uruchamiania skryptów PowerShell, które wykonują zdalne za Admin02, jest automatyzacja poświadczeń używanych w Invoke-Command dla konta, którego będziesz używać do uwierzytelniania i wykonywania na zdalnych komputerach.

Jest na to kilka sposobów, ale lepiej zresetować hasło do konta, na którym będę uwierzytelniać się na zdalnych maszynach przed wywołaniem Invoke-Command, a następnie zresetować je ponownie po zakończeniu Invoke-Command. Korzystając z tej metody, nawet jeśli ktoś ukradnie hasło, skrypt zmieni je zaraz po zakończeniu jego działania, czyniąc je przestarzałym.

```
$s = New-PSSession -ComputerName "$env:ComputerName" -Credential(Get-Credential -Credential "$env:ComputerName\Admin02")

Invoke-Command -Session $s -ScriptBlock { net user Admin02 /random | Out-Null }
```

Out-Null zastosuję, aby po przejęciu sesji nie możliwe było odnalezienie przypadkowego hasła.

3. Łączenie wszystkiego w skrypt PowerShell

Teraz, gdy przedstawiłem już pewien wgląd w bezpieczeństwo tego skryptu zdalnego, przyjrzyjmy się kodowi i wynikom. Kod umieściłem w skrypcie PowerShell.

```
Add-LocalGroupMember -Group "Administratorzy" -Member Admin02 -Verbose

$cred = Get-Credential -Credential "$env:ComputerName\Admin02"

Invoke-Command -ComputerName "$env:ComputerName" -Credential $cred -ScriptBlock {

    Restart-Service wuauserv -Verbose

}

$s = New-PSSession -ComputerName "$env:ComputerName" -Credential(Get-Credential -Credential "$env:ComputerName\Admin02")

Invoke-Command -Session $s -ScriptBlock { Get-Service | FT -GroupBy Status | Out-File C:\uslugi.txt } -Verbose

Invoke-Command -Session $s -ScriptBlock { net user Admin02 /random | Out-Null }

$members = "Admin02"
```

```
Remove-LocalGroupMember -Group "Administratorzy" -Member $members –Confirm:$False -Verbose

Get-PSSession | Remove-PSSession -Verbose
```

```
VERBOSE: Performing the operation "Add member stacja\Admin02" on target "Administratorzy".
VERBOSE: Performing the operation "Restart-Service" on target "Windows Update (wuauserv)".
VERBOSE: Performing the operation "Remove member stacja\Admin02" on target "Administratorzy".
VERBOSE: Performing the operation "Remove" on target "STACJA".
```

Czynności, które wykonał skrypt:

Dodał konto Admin02 do grupy bezpieczeństwa Administratorzy (lokalna grupa administratorów na zdalnym komputerze).

Potwierdzam hasło, jednocześnie umieszczając je w obiekcie poświadczeń.

Używamy Invoke-Command z tym poświadczeniem, aby połączyć się z komputerami zdalnymi i ponownie uruchomić usługę Windows Update.

Używamy Invoke-Command z tym poświadczeniem, aby połączyć się z komputerami zdalnymi i wysłać spis działających i nie działających usług do pliku C:\uslugi.txt

Resetujemy hasło Admin02 generując przypadkowe.

Usuwamy konto Admin02 z grupy Administratorzy, usuwając w ten sposób uprawnienia administratora na zdalnych komputerach.

Usuwamy sesje.

Ideą tej metody jest stworzenie ruchomego celu dla tego konta, co zmniejsza prawdopodobieństwo naruszenia bezpieczeństwa.

B. Zabezpieczenie dostępu do urządzeń mobilnych

Wykonaj notatkę.

Omówienie podstawowej mobilności i zabezpieczeń dla platformy Microsoft 365

Możesz zarządzać urządzeniami przenośnymi i zabezpieczać je, gdy są połączone z organizacją Microsoft 365, korzystając z podstawowej mobilności i zabezpieczeń. Urządzenia mobilne, takie jak smartfony i tablety, które są używane do uzyskiwania dostępu do służbowej poczty e-mail, kalendarza, kontaktów i dokumentów, odgrywają dużą rolę w zapewnianiu pracownikom wykonywania pracy w dowolnym czasie i miejscu. Dlatego tak ważne jest, aby pomóc chronić informacje organizacji, gdy

ludzie korzystają z urządzeń. Możesz użyć Basic Mobility and Security, aby ustawić zasady bezpieczeństwa urządzeń i reguły dostępu oraz wyczyścić urządzenia mobilne, jeśli zostaną zgubione lub skradzione.

Prywatność i bezpieczeństwo w Basic Mobility and Security

Basic Mobility and Security to usługa oparta na chmurze obsługiwana przez Microsoft Intune, która pomaga zarządzać i zabezpieczać urządzenia mobilne w organizacji. Po aktywacji Basic Mobility and Security możesz tworzyć zasady zarządzania urządzeniami mobilnymi. Te zasady można następnie wdrożyć na urządzeniach przenośnych zarejestrowanych przez licencjonowanych użytkowników platformy Microsoft 365 w organizacji.

Usługa Microsoft Intune wysyła do usługi Microsoft 365 informacje o stanie zgodności każdego zarządzanego urządzenia, a następnie można generować raporty, które pokazują, czy zarządzane urządzenia w organizacji są zgodne na podstawie ustawionych zasad.

W PowerShell możesz zarządzać urządzeniami przenośnymi i zabezpieczeniami w Microsoft 365, korzystając z różnych cmdletów, które pozwalają na tworzenie i wdrażanie zasad bezpieczeństwa, a także na monitorowanie stanu urządzeń.

Usługa Microsoft Intune to oparte na chmurze rozwiązanie do zarządzania punktami końcowymi, które pomaga organizacjom bezpiecznie zarządzać tożsamościami, aplikacjami i urządzeniami. Intune umożliwia zarządzanie dostępem użytkowników do zasobów organizacji oraz upraszcza zarządzanie aplikacjami i urządzeniami na wielu platformach, w tym na urządzeniach przenośnych, komputerach stacjonarnych i wirtualnych punktach końcowych.

Kluczowe funkcje i korzyści usługi Intune:

Zarządzanie użytkownikami i urządzeniami: Intune obsługuje urządzenia z systemami Android, iOS/iPadOS, macOS i Windows. Pozwala na bezpieczne uzyskiwanie dostępu do zasobów organizacji przy użyciu utworzonych zasad.

Uproszczone zarządzanie aplikacjami: Intune posiada wbudowane środowisko aplikacji, które umożliwia wdrażanie, aktualizację i usuwanie aplikacji.

Ochrona danych: Intune integruje się z Microsoft Entra dla kontroli tożsamości i dostępu oraz z Azure Information Protection dla ochrony danych.

Zgodność i raportowanie: Intune oferuje funkcje zgodności i raportowania, które wspierają model zabezpieczeń Zero Trust.

Intune jest szczególnie przydatne dla organizacji obsługujących pracowników hybrydowych i zdalnych, ponieważ pozwala na zarządzanie różnymi urządzeniami uzyskującymi dostęp do zasobów organizacji z dowolnego miejsca1.

Pierwszy przykład skryptu PowerShell, który tworzy nową zasadę zarządzania urządzeniami mobilnymi w Microsoft Intune:

```
# Połączenie z usługą Intune

Connect-MSGraph

# Tworzenie nowej zasady zarządzania urządzeniami mobilnymi

$deviceCompliancePolicy = New-DeviceCompliancePolicy -Name "Zasada Bezpieczeństwa Urządzeń" -Platform Android -Settings @{passwordRequired=$true; passwordMinimumLength=6}

# Wdrażanie zasady na urządzeniach

Assign-DeviceCompliancePolicy -PolicyId $deviceCompliancePolicy.id -IncludedGroups $groupId

# Sprawdzanie stanu zgodności urządzeń

Get-DeviceCompliancePolicyDeviceStatus -PolicyId $deviceCompliancePolicy.id
```

Ten skrypt najpierw nawiązuje połączenie z usługą Microsoft Intune za pomocą Connect-MSGraph. Następnie tworzy nową zasadę (New-DeviceCompliancePolicy), która wymaga hasła na urządzeniach z systemem Android i określa minimalną długość hasła. Po utworzeniu zasady, skrypt przypisuje ją do grupy urządzeń (Assign-DeviceCompliancePolicy) i na koniec sprawdza stan zgodności urządzeń z tą zasadą (Get-DeviceCompliancePolicyDeviceStatus).

Pamiętaj, że aby używać tych cmdletów, musisz mieć odpowiednie uprawnienia w Microsoft 365 oraz zainstalowany moduł PowerShell dla Microsoft Graph.

Drugi przykład skryptu PowerShell, który może być użyty do zarządzania urządzeniami przenośnymi w Microsoft 365:

```
# Połączenie z usługą Intune

Connect-MSGraph

# Pobieranie listy wszystkich zasad zgodności urządzeń
```

```
$deviceCompliancePolicies = Get-DeviceCompliancePolicy

# Wyświetlanie nazw i identyfikatorów zasad

foreach ($policy in $deviceCompliancePolicies) {

    Write-Host "Nazwa zasady: " $policy.displayName

    Write-Host "ID zasady: " $policy.id

}

# Usuwanie zasady zgodności urządzenia o określonym ID

Remove-DeviceCompliancePolicy -PolicyId "ID_zasady_do_usunięcia"
```

Ten skrypt najpierw łączy się z usługą Microsoft Intune (Connect-MSGraph). Następnie pobiera listę wszystkich zasad zgodności urządzeń (Get-DeviceCompliancePolicy) i wyświetla ich nazwy oraz identyfikatory. Na końcu skrypt usuwa zasadę zgodności urządzenia o określonym ID (Remove-DeviceCompliancePolicy).

Pamiętaj, że przed użyciem tego skryptu należy zastąpić "ID_zasady_do_usunięcia" rzeczywistym identyfikatorem zasady, którą chcesz usunąć. Ponadto, upewnij się, że masz odpowiednie uprawnienia do zarządzania zasadami w Microsoft Intune.

Trzeci przykład skryptu PowerShell, który może być używany do zarządzania urządzeniami przenośnymi w Microsoft 365:

```
# Połączenie z usługą Intune

Connect-MSGraph

# Pobieranie informacji o zarejestrowanych urządzeniach

$registeredDevices = Get-IntuneManagedDevice

# Wyświetlanie informacji o każdym urządzeniu

foreach ($device in $registeredDevices) {

    Write-Host "Nazwa urządzenia: " $device.deviceName

    Write-Host "Model urządzenia: " $device.model

    Write-Host "System operacyjny: " $device.operatingSystem
```

```
    Write-Host "Wersja systemu operacyjnego: " $device.osVersion
    Write-Host "Status zgodności: " $device.complianceState
}
# Wysyłanie żądania resetowania hasła na urządzeniu o określonym ID
Invoke-DeviceAction -ActionType "resetPasscode" -DeviceId "ID_urządzenia_do_resetu"
```

W tym skrypcie, po połączeniu z usługą Microsoft Intune (Connect-MSGraph), pobierane są informacje o zarejestrowanych urządzeniach (Get-IntuneManagedDevice). Następnie skrypt wyświetla szczegółowe informacje o każdym urządzeniu. Na końcu skrypt wysyła żądanie resetowania hasła na urządzeniu o określonym ID (Invoke-DeviceAction).

Zanim użyjesz tego skryptu, upewnij się, że "ID_urządzenia_do_resetu" jest zastąpione rzeczywistym identyfikatorem urządzenia, na którym chcesz przeprowadzić akcję. Ponadto, sprawdź, czy masz odpowiednie uprawnienia do wykonania tej akcji w środowisku Microsoft Intune.

Microsoft Azure, znana również jako **Chmura Azure**, to platforma chmurowa działająca w modelu Platform as a Service (PaaS). Jest to jedno z najpopularniejszych rozwiązań chmurowych na świecie, które oferuje szeroki zakres usług i produktów dla przedsiębiorstw różnej wielkości.

Kluczowe funkcji i usług oferowanych przez Azure:

Wirtualizacja: Azure pozwala na tworzenie i zarządzanie maszynami wirtualnymi, serwerami, magazynami danych i środowiskami programistycznymi.

Elastyczność: Umożliwia elastyczne skalowanie zasobów w zależności od potrzeb, co jest szczególnie przydatne w przypadku zmiennego obciążenia.

Bezpieczeństwo: Azure zapewnia wysoki poziom bezpieczeństwa danych dzięki nowoczesnym centrom danych spełniającym najwyższe standardy.

Oszczędność kosztów: Dzięki modelowi płatności za rzeczywiste zużycie zasobów (pay-per-use), Azure pozwala na obniżenie kosztów infrastruktury IT.

Wsparcie dla wielu języków programowania: Azure obsługuje popularne języki programowania, co ułatwia tworzenie i wdrażanie aplikacji.

Azure jest używany przez około 95% firm z listy Fortune 500, co świadczy o jego dużej popularności i zaufaniu wśród największych potentatów branżowych. Platforma ta jest szczególnie przydatna dla firm, które chcą uniknąć kosztów związanych z posiadaniem i utrzymaniem własnej serwerowni, oferując jednocześnie możliwość szybkiego wdrożenia i zarządzania rozwiązaniami IT.

Czwarty przykład, zarządzanie urządzeniami w chmurze Azure można wykonać za pomocą Azure PowerShell, który jest potężnym narzędziem do automatyzacji zadań zarządzania zasobami Azure. Przykład skryptu PowerShell, który pokazuje, jak można zarządzać maszyną wirtualną (VM) w Azure:

```
# Logowanie do konta Azure
Connect-AzAccount

# Wybór subskrypcji, w której znajduje się maszyna wirtualna
Set-AzContext -SubscriptionId 'Twoje-ID-subskrypcji'

# Pobieranie informacji o maszynie wirtualnej
$vm = Get-AzVM -ResourceGroupName 'Twoja-nazwa-grupy-zasobów' -Name 'Nazwa-twojej-VM'

# Wyświetlanie stanu maszyny wirtualnej
Write-Host "Stan maszyny wirtualnej:" $vm.Statuses[1].DisplayStatus

# Uruchamianie maszyny wirtualnej
Start-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName

# Zatrzymywanie maszyny wirtualnej
Stop-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName -Force

# Usuwanie maszyny wirtualnej
Remove-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName -Force
```

W tym skrypcie, Connect-AzAccount służy do logowania do konta Azure. Set-AzContext pozwala wybrać subskrypcję, w której znajduje się maszyna wirtualna. Get-AzVM pobiera informacje o maszynie wirtualnej. Start-AzVM i Stop-AzVM służą odpowiednio do uruchamiania i zatrzymywania maszyny wirtualnej. Remove-AzVM pozwala na usunięcie maszyny wirtualnej.

Pamiętaj, aby zastąpić 'Twoje-ID-subskrypcji', 'Twoja-nazwa-grupy-zasobów' i 'Nazwa-twojej-VM' odpowiednimi wartościami dla Twojego środowiska Azure. Użyj tego skryptu z odpowiednią

ostrożnością, ponieważ akcje takie jak zatrzymywanie lub usuwanie maszyny wirtualnej są nieodwracalne.