ATTACK DEFENSE

by PentesterAcademy

Name	Windows: Securing WinRM Service
URL	https://attackdefense.com/challengedetails?cid=2032
Type	Windows Exploitation: Services

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Note: By default if you are using Windows Server 2012 R2+ then, WinRM service is already up and running. You need to configure the service in order to access it remotely. In this manual we are demonstrating how to secure WinRM service for learning purposes.

Securing WinRM

In this manual we are going to discuss the following points.

1. Strong Password for Administrator account
2. Configuration of HTTPS listener for WinRM and Disable HTTP listener
3. Dedicated User for WinRM service with minimal privilege.
4. Client side TrustedHosts management
5. Configure host firewall to restrict WinRM access to allow communication only to specific devices.
6. Stop & Disable the WinRM service if not in use. By default it is up and running.

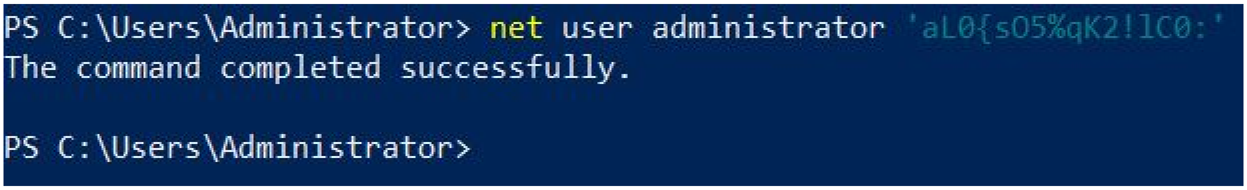
Strong Password for Administrator account

When there is a requirement to configure the WinRM service for remote access. then, it is very important to use strong passwords for all the remote manager users, especially the high privileged ones. This would reduce some amount of risks and when an attacker launches a brute force attack then it would generate a huge amount of noise and we can block/prevent from such attacks by monitoring the log events.

We can use **net** command to change the administrator password if it’s a weak one.

Step 1: Changing administrator password.

Command: net user administrator 'aL0{sO5%qK2!lC0:'



```
PS C:\Users\Administrator> net user administrator 'aL0{sO5%qK2!lC0:'
The command completed successfully.

PS C:\Users\Administrator>
```

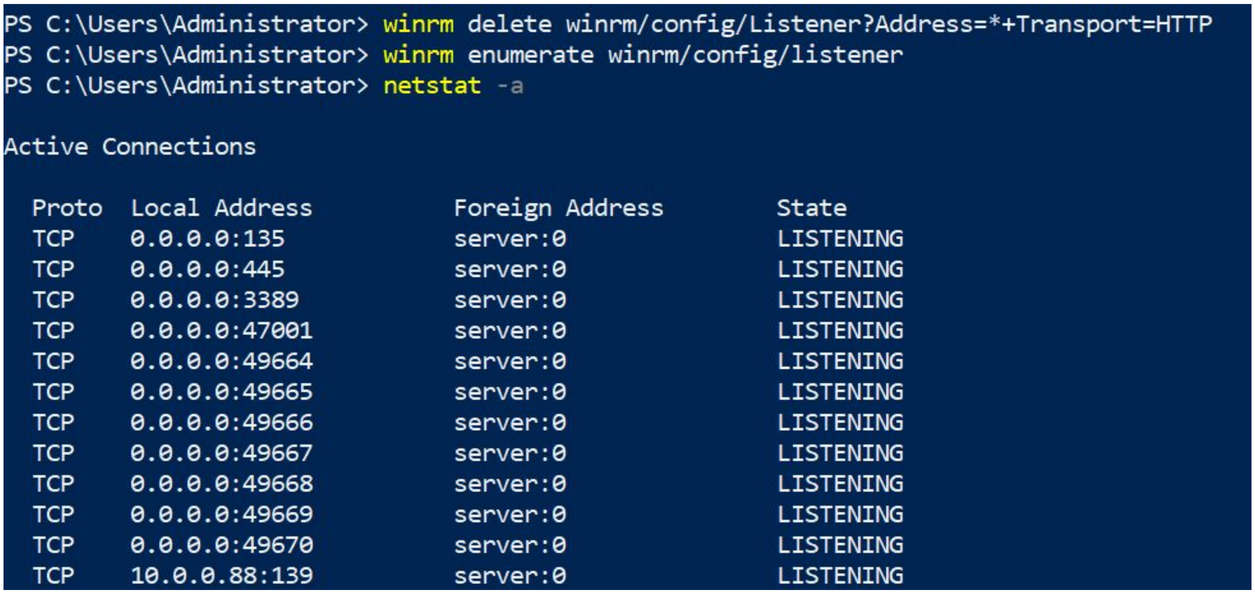
We have successfully changed the administrator password. The password contains Uppercase, Lowercase, Numbers, Symbols. This is just a random password.

Configuration of HTTPS listener and Disable HTTP listener

By default WinRM service is configured on HTTP listener. One can easily sniff the HTTP traffic and get the plaintext credentials. However, we can create a self sign certificate and configure WinRM HTTPS listener which is more secure than HTTP listener and doesn't send plaintext credentials in the network.

Step 1: Removing HTTP listener

Command: winrm delete winrm/config/Listener?Address=*+Transport=HTTP
winrm enumerate winrm/config/listener
netstat -a



```
PS C:\Users\Administrator> winrm delete winrm/config/Listener?Address=*+Transport=HTTP
PS C:\Users\Administrator> winrm enumerate winrm/config/listener
PS C:\Users\Administrator> netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            server:0               LISTENING
  TCP    0.0.0.0:445            server:0               LISTENING
  TCP    0.0.0.0:3389           server:0               LISTENING
  TCP    0.0.0.0:47001          server:0               LISTENING
  TCP    0.0.0.0:49664          server:0               LISTENING
  TCP    0.0.0.0:49665          server:0               LISTENING
  TCP    0.0.0.0:49666          server:0               LISTENING
  TCP    0.0.0.0:49667          server:0               LISTENING
  TCP    0.0.0.0:49668          server:0               LISTENING
  TCP    0.0.0.0:49669          server:0               LISTENING
  TCP    0.0.0.0:49670          server:0               LISTENING
  TCP    10.0.0.88:139          server:0               LISTENING
```

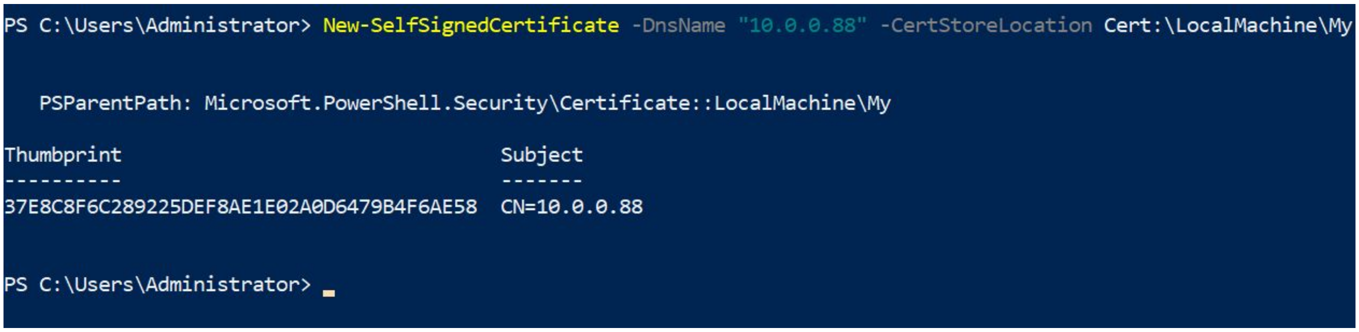
We can observe that we have successfully deleted the HTTP listener and port 5985 is not listening.

Note: Check the target machine IP address. In my case it is "**10.0.0.88**" and remember to replace the below IP with your valid target machine IP address.

Step 2: Generating self sign certificate.

Note: We are generating certificates for local setup so we can use the server IP address for DnsName.

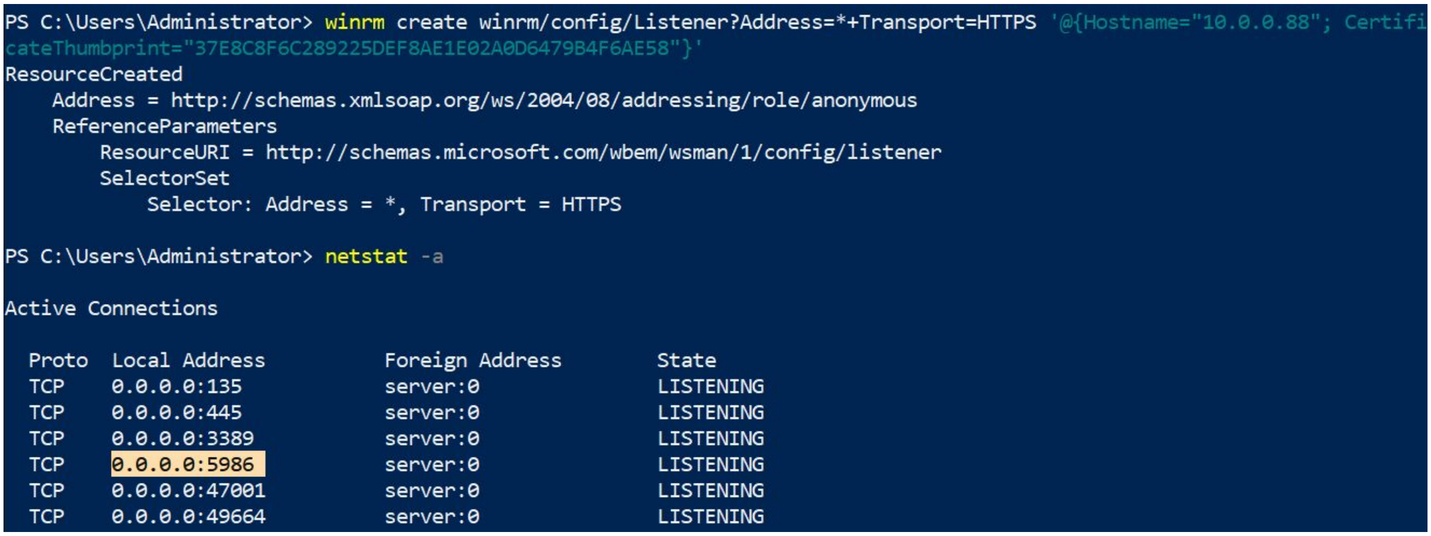
Command: New-SelfSignedCertificate -DnsName "**10.0.0.88**" -CertStoreLocation Cert:\LocalMachine\My



```
PS C:\Users\Administrator> New-SelfSignedCertificate -DnsName "10.0.0.88" -CertStoreLocation Cert:\LocalMachine\My


   PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                                Subject
----------                                -------
37E8C8F6C289225DEF8AE1E02A0D6479B4F6AE58  CN=10.0.0.88


PS C:\Users\Administrator> _
```

Step 3: Creating a HTTPS winrm listener using “Certificate Thumbprint” and Hostname.

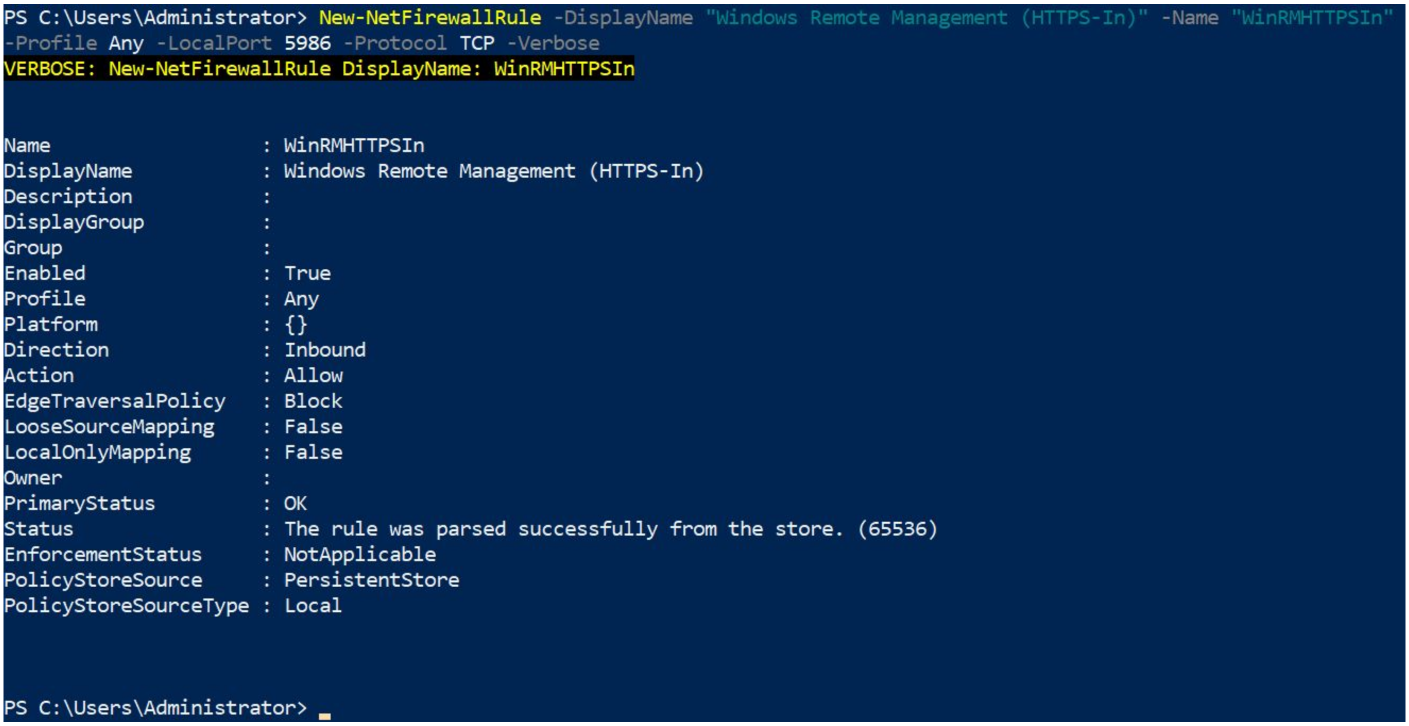
Command: winrm create winrm/config/Listener?Address=*+Transport=HTTPS '@{Hostname="**10.0.0.88**"; CertificateThumbprint="37E8C8F6C289225DEF8AE1E02A0D6479B4F6AE58"}'
netstat -a



```
PS C:\Users\Administrator> winrm create winrm/config/Listener?Address=*+Transport=HTTPS '@{Hostname="10.0.0.88"; Certifi
cateThumbprint="37E8C8F6C289225DEF8AE1E02A0D6479B4F6AE58"}'
ResourceCreated
    Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    ReferenceParameters
        ResourceURI = http://schemas.microsoft.com/wbem/wsman/1/config/listener
        SelectorSet
            Selector: Address = *, Transport = HTTPS

PS C:\Users\Administrator> netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            server:0               LISTENING
  TCP    0.0.0.0:445            server:0               LISTENING
  TCP    0.0.0.0:3389           server:0               LISTENING
  TCP    0.0.0.0:5986           server:0               LISTENING
  TCP    0.0.0.0:47001          server:0               LISTENING
  TCP    0.0.0.0:49664          server:0               LISTENING
```

Step 3: We need to create a firewall rule to allow 5986 incoming traffic.

Command: New-NetFirewallRule -DisplayName "Windows Remote Management (HTTPS-In)" -Name "WinRMHTTPSIn" -Profile Any -LocalPort 5986 -Protocol TCP -Verbose



```
PS C:\Users\Administrator> New-NetFirewallRule -DisplayName "Windows Remote Management (HTTPS-In)" -Name "WinRMHTTPSIn"
-Profile Any -LocalPort 5986 -Protocol TCP -Verbose
VERBOSE: New-NetFirewallRule DisplayName: WinRMHTTPSIn


Name                  : WinRMHTTPSIn
DisplayName           : Windows Remote Management (HTTPS-In)
Description           :
DisplayGroup          :
Group                 :
Enabled               : True
Profile               : Any
Platform              : {}
Direction             : Inbound
Action                : Allow
EdgeTraversalPolicy   : Block
LooseSourceMapping    : False
LocalOnlyMapping      : False
Owner                 :
PrimaryStatus         : OK
Status                : The rule was parsed successfully from the store. (65536)
EnforcementStatus     : NotApplicable
PolicyStoreSource     : PersistentStore
PolicyStoreSourceType : Local



PS C:\Users\Administrator>
```

The HTTPS listener is up and running.

Dedicated User for WinRM service with minimal privilege.

It is obvious that we can’t give an administrator user account access to all the clients. Also, we should not provide normal users with administrator privileges. We should select their role and we should create a group for that specific role and then the users are only able to access those services. i.e Event Log Readers, Performance Monitor, Backup Operators, etc..

We will be creating a normal user i.e **winrmdemo** and adding the user to **Remote Management Users**, **Event Log Readers**.

Remote Management Users:

Members of this group can access WMI resources over management protocols (such as WS-Management via the Windows Remote Management service). This applies only to WMI namespaces that grant access to the user.

Event Log Readers:

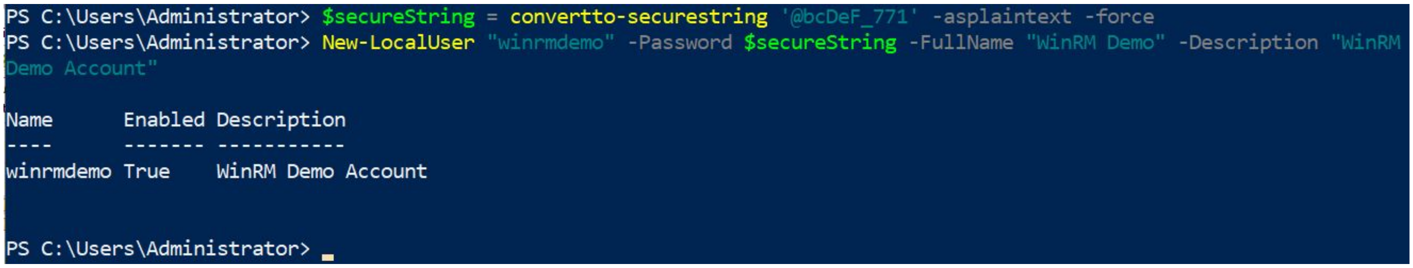
Members of this group can read event logs from local machine

Also, please note by default any users which we create on the windows machine using GUI are a member of the "**users**" group. This group restricts users from making any accidental or intentional system-wide changes but these users can run most of the applications.

Step 1: Creating a "**winrmdemo"** user and adding this user to remote management users and event log readers groups.

Command:

$secureString = convertto-securestring '**@bcDeF_771**' -asplaintext -force
New-LocalUser "**winrmdemo**" -Password $secureString -FullName "WinRM Demo" -Description "WinRM Demo Account"

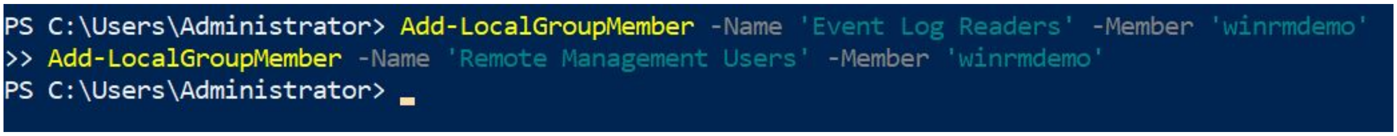


```
PS C:\Users\Administrator> $secureString = convertto-securestring '@bcDeF_771' -asplaintext -force
PS C:\Users\Administrator> New-LocalUser "winrmdemo" -Password $secureString -FullName "WinRM Demo" -Description "WinRM
Demo Account"

Name      Enabled Description
----      ------- -----------
winrmdemo True    WinRM Demo Account


PS C:\Users\Administrator> _
```

We have successfully, created an user i.e "winrmdemo" .

Step 2: Adding user to local group member.

Commands:
Add-LocalGroupMember -Name '**Event Log Readers**' -Member '**winrmdemo**'
Add-LocalGroupMember -Name '**Remote Management Users**' -Member '**winrmdemo**'



```
PS C:\Users\Administrator> Add-LocalGroupMember -Name 'Event Log Readers' -Member 'winrmdemo'
>> Add-LocalGroupMember -Name 'Remote Management Users' -Member 'winrmdemo'
PS C:\Users\Administrator> _
```

We have added the user in the Remote Management Users and Event Log Readers groups.

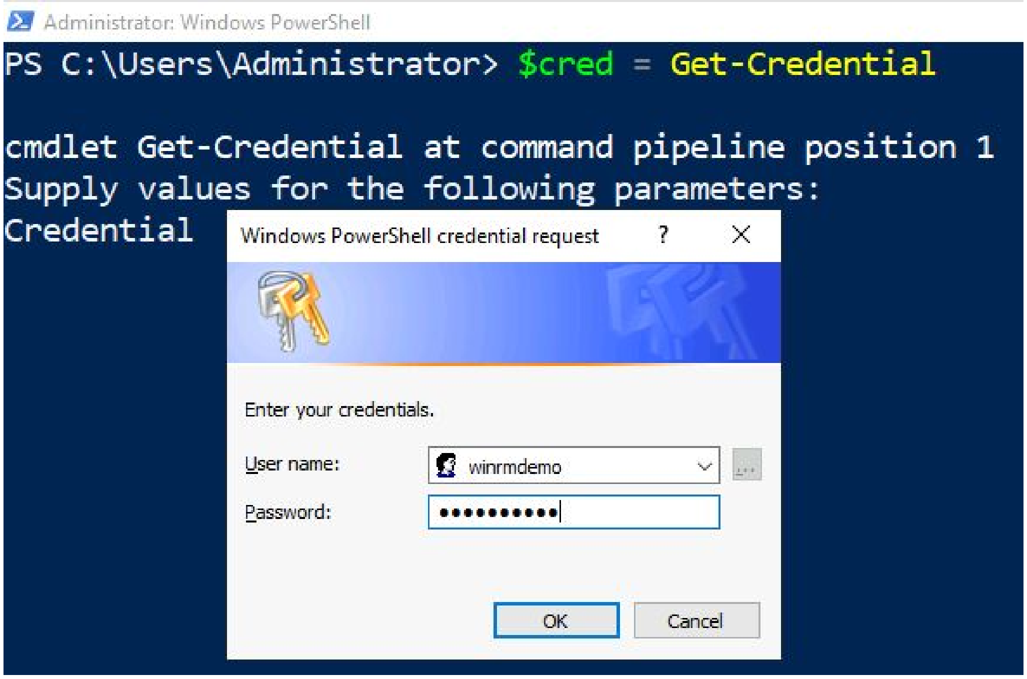
We have successfully configured the WinRM service on the remote server.

Step 3: Switch to the attacker machine and connect to the winrm service using winrmdemo user.

Commands:

#Store credentials to $cred

$cred = Get-Credential



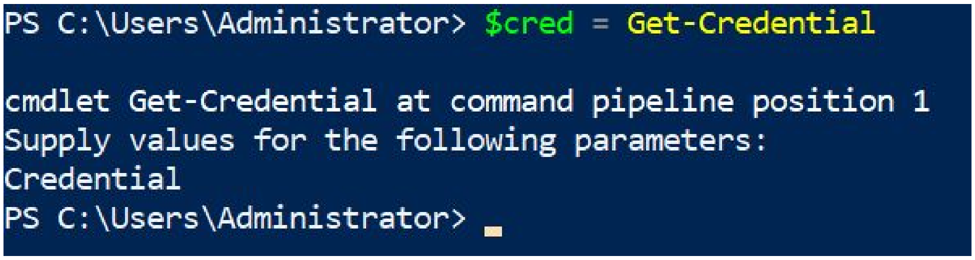
```
PS C:\Users\Administrator> $cred = Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
```

Windows PowerShell credential request

Enter your credentials.

User name: winrmdemo

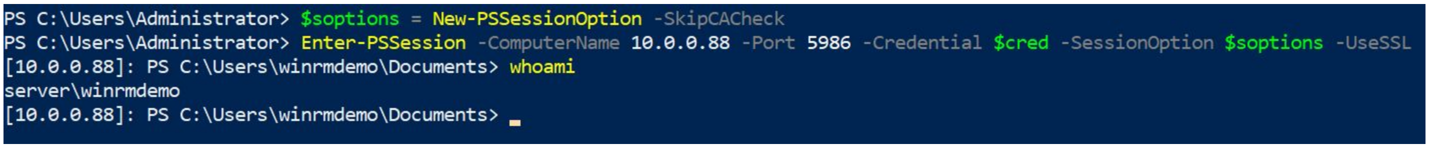
Password: ••••••••••



```
PS C:\Users\Administrator> $cred = Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
PS C:\Users\Administrator> _
```

# Connecting to the machine

$soptions = New-PSSessionOption -SkipCACheck
Enter-PSSession -ComputerName 10.0.0.88 -Port 5986 -Credential $cred -SessionOption $soptions -UseSSL



```
PS C:\Users\Administrator> $soptions = New-PSSessionOption -SkipCACheck
PS C:\Users\Administrator> Enter-PSSession -ComputerName 10.0.0.88 -Port 5986 -Credential $cred -SessionOption $soptions -UseSSL
[10.0.0.88]: PS C:\Users\winrmdemo\Documents> whoami
server\winrmdemo
[10.0.0.88]: PS C:\Users\winrmdemo\Documents> _
```

We are connected to the remote server using ‘winrmdemo’ user over HTTPS listener.

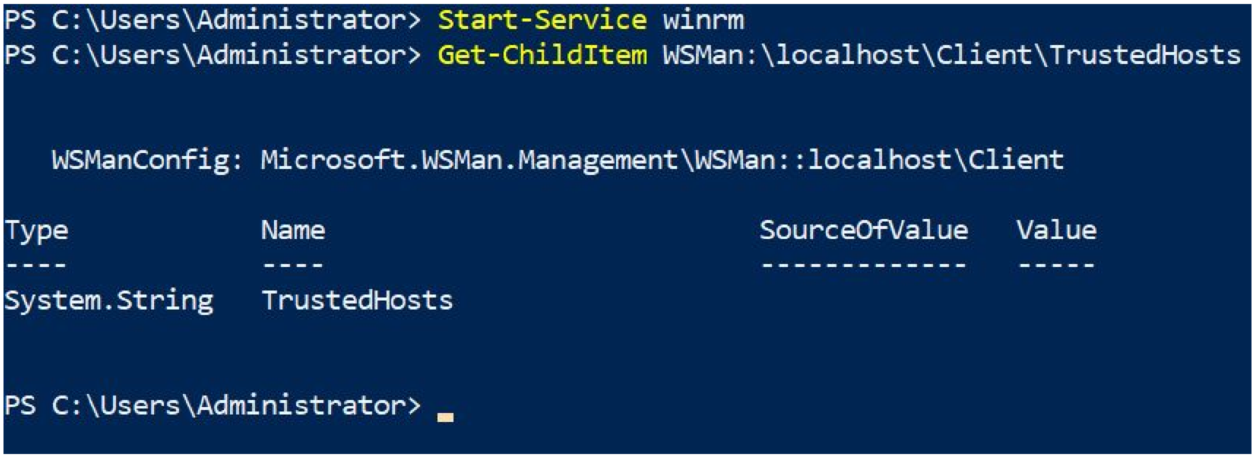
Client side TrustedHosts Management

We do not have to add the remote hosts to our “TrustedHosts” as we are connected to the remote server via HTTPS. It is only required when the remote server is running WinRM service on HTTP listener.

We can verify by running below command on the attacker machine. You could exit the PSSession or run another powershell terminal to execute the below command.

Command: Start-Service winrm

Get-ChildItem WSMan:\localhost\Client\TrustedHosts



```
PS C:\Users\Administrator> Start-Service winrm
PS C:\Users\Administrator> Get-ChildItem WSMan:\localhost\Client\TrustedHosts


   WSManConfig: Microsoft.WSMan.Management\WSMan::localhost\Client

Type            Name                           SourceOfValue   Value
----            ----                           -------------   -----
System.String   TrustedHosts


PS C:\Users\Administrator> _
```

We can observe that we have not added any TrustedHost entries.

Configure host firewall to restrict WinRM access to allow communication only to specific machines.

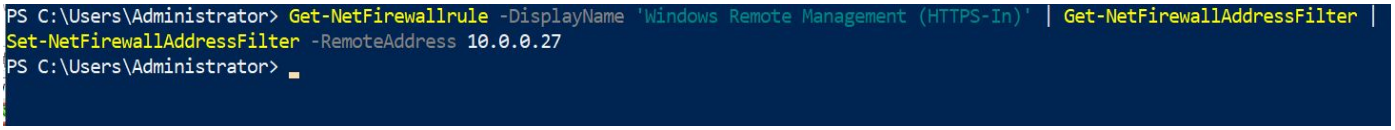
This is the best possible way to block unwanted devices to connect to the WinRM service. This will block any brute force attempt.

We need to modify the firewall rule and we need to add trusted machines IP addresses only for incoming connections to the WinRM service. This way only someone who has that specific IP address eg: “X.X.X.X” only can connect. We can also add subnet instead of single IP addresses.

Please make sure about your attacker machine IP address.

Command: Get-NetFirewallrule -DisplayName 'Windows Remote Management (HTTPS-In)' | Get-NetFirewallAddressFilter | Set-NetFirewallAddressFilter -RemoteAddress 10.0.0.27

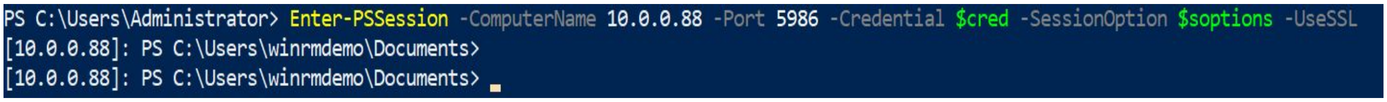
Note: The above command would overwrite all existing entries for this rule and add the remote address 10.0.0.27.



```
PS C:\Users\Administrator> Get-NetFirewallrule -DisplayName 'Windows Remote Management (HTTPS-In)' | Get-NetFirewallAddressFilter |
Set-NetFirewallAddressFilter -RemoteAddress 10.0.0.27
PS C:\Users\Administrator> _
```

Now, only the 10.0.0.27 host is allowed to connect to the WinRM service.

Switch to the attacker machine and connect to the server.



```
PS C:\Users\Administrator> Enter-PSSession -ComputerName 10.0.0.88 -Port 5986 -Credential $cred -SessionOption $soptions -UseSSL
[10.0.0.88]: PS C:\Users\winrmdemo\Documents>
[10.0.0.88]: PS C:\Users\winrmdemo\Documents>
```

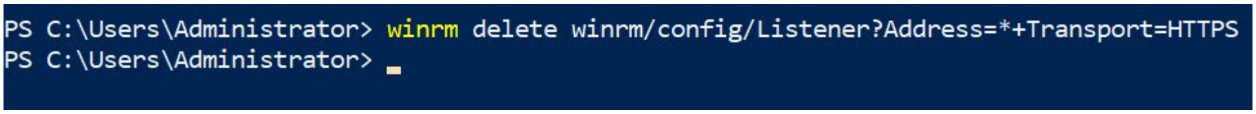
Stop & Disable the WinRM service if not in use

It is always a good practice to disable or remove the WinRM service if it is not in use. We can stop and disable the service using powershell as shown below and run this command on the target machine.

Step 1: Delete HTTP or HTTPS listener

Note: We have already deleted HTTP listener while configuring HTTPS listener, so we are only deleting HTTPS listener now.

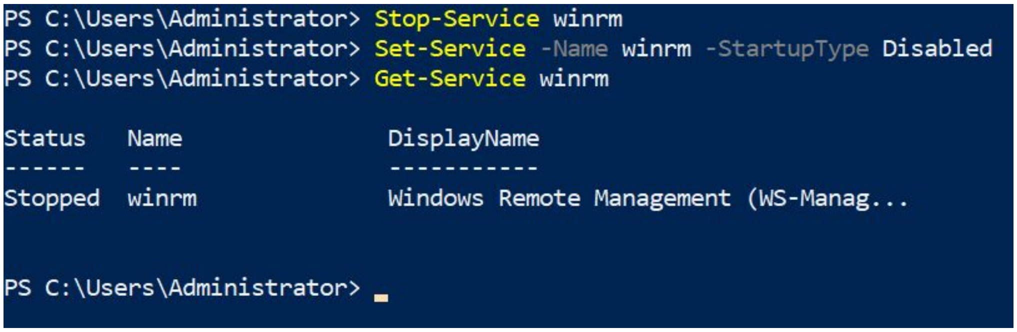
Command: winrm delete winrm/config/Listener?Address=*+Transport=HTTPS



```
PS C:\Users\Administrator> winrm delete winrm/config/Listener?Address=*+Transport=HTTPS
PS C:\Users\Administrator>
```

Step 2: Stop and disable the WinRM service.

Commands: Stop-Service winrm
Set-Service -Name winrm -StartupType Disabled
Get-Service winrm



```
PS C:\Users\Administrator> Stop-Service winrm
PS C:\Users\Administrator> Set-Service -Name winrm -StartupType Disabled
PS C:\Users\Administrator> Get-Service winrm

Status   Name               DisplayName
------   ----               -----------
Stopped  winrm              Windows Remote Management (WS-Manag...


PS C:\Users\Administrator> _
```

We have successfully stopped and disabled the WinRM service.

References:

- Installation and configuration for Windows Remote Management (https://docs.microsoft.com/en-us/windows/win32/winrm/installation-and-configuration-for-windows-remote-management)
- Enable-PSRemoting (https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7)
- WinRM Security (https://docs.microsoft.com/en-us/powershell/scripting/learn/remoting/winrmsecurity?view=powershell-7)