

The image features a word cloud where the words are arranged in a circular shape. The most prominent words are "ATTACK" in large red font and "DEFENSE" in large dark blue font, positioned centrally. Other visible words include "LABS", "PENTESTER ACADEMY", "RED TEAM", "TOOL BOX", "TRAINING", "COURSES", "ACCESS POINT", "HACKER", "PATV", "WORLD-CLASS TRAINERS", "ATTACKDEFENSE LABS", "TEAM LABS", "PENTESTER ACADEMY", "TOOL BOX", "TRAINING", "COURSES", "ACCESS POINT", "HACKER", "PATV", "WORLD-CLASS TRAINERS", "ATTACKDEFENSE LABS", "TEAM LABS", "PENTESTER ACADEMY", "TOOL BOX", "TRAINING", "COURSES", "ACCESS POINT", "HACKER", "PATV", "WORLD-CLASS TRAINERS". The background is white, and the overall design is clean and modern.

Name	WinRM: Configure via WinRM.CMD
URL	https://attackdefense.com/challengedetails?cid=2034
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 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 enable WinRM service and making necessary changes for learning purposes.

Configuration of WinRM

Step 1: Checking the status of the winrm service on the target machine. Run powershell.exe to check for WinRM service status, if it's running or not.

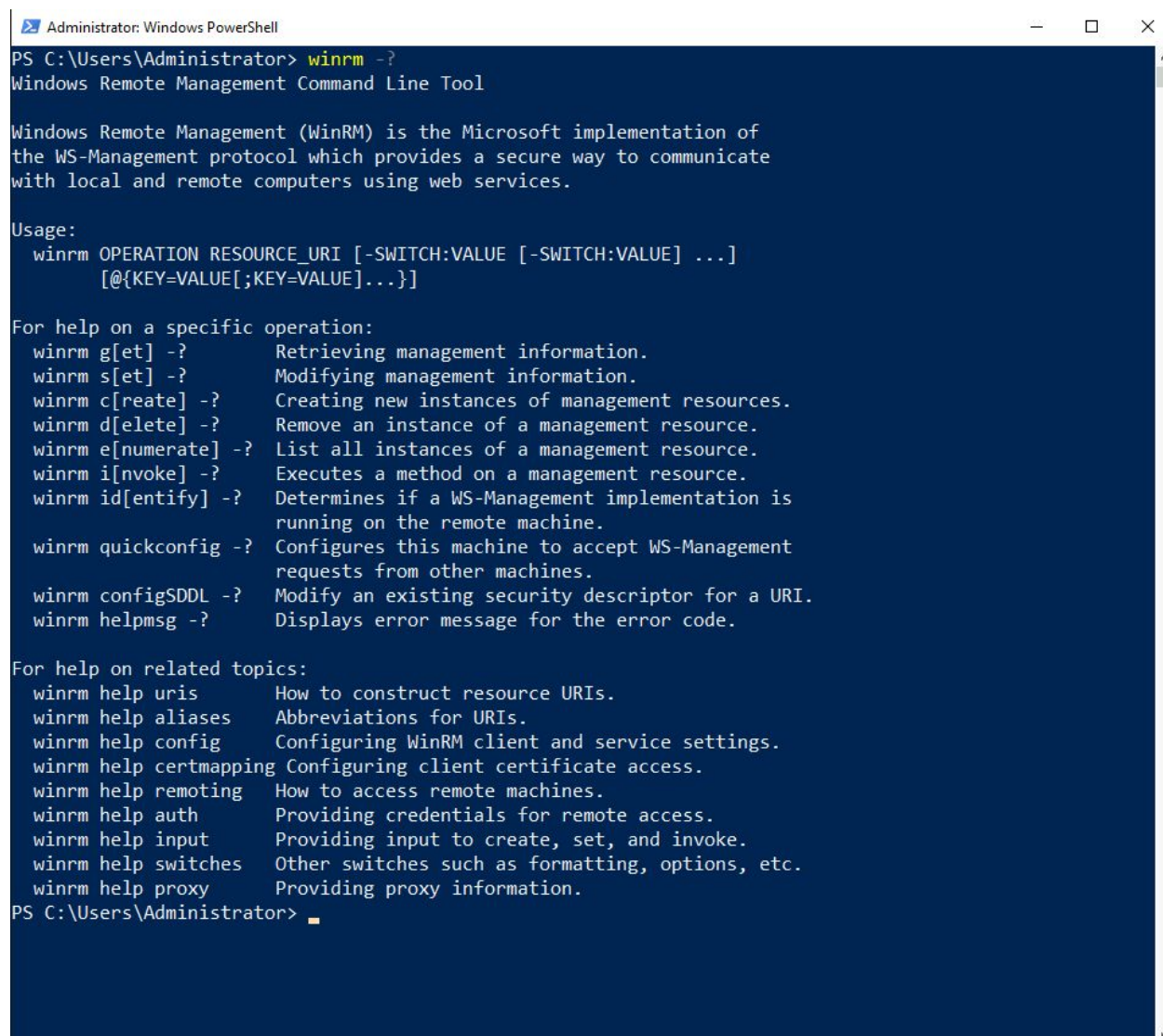
Command: Get-Service winrm

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-Service winrm

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

The WinRM service is not running. We will use **winrm.cmd** to enable the WinRM service, we can invoke the command by typing **winrm** only. First, check the help of the winrm.cmd.

Command: winrm -?

A screenshot of a Windows PowerShell window titled 'Administrator: Windows PowerShell'. The command 'winrm -?' has been entered, and the output is displayed. The output includes a description of WinRM, its usage syntax, and a list of available operations and help topics.

```
PS C:\Users\Administrator> winrm -?
Windows Remote Management Command Line Tool

Windows Remote Management (WinRM) is the Microsoft implementation of
the WS-Management protocol which provides a secure way to communicate
with local and remote computers using web services.

Usage:
  winrm OPERATION RESOURCE_URI [-SWITCH:VALUE [-SWITCH:VALUE] ...]
  [{KEY=VALUE[;KEY=VALUE]...}]

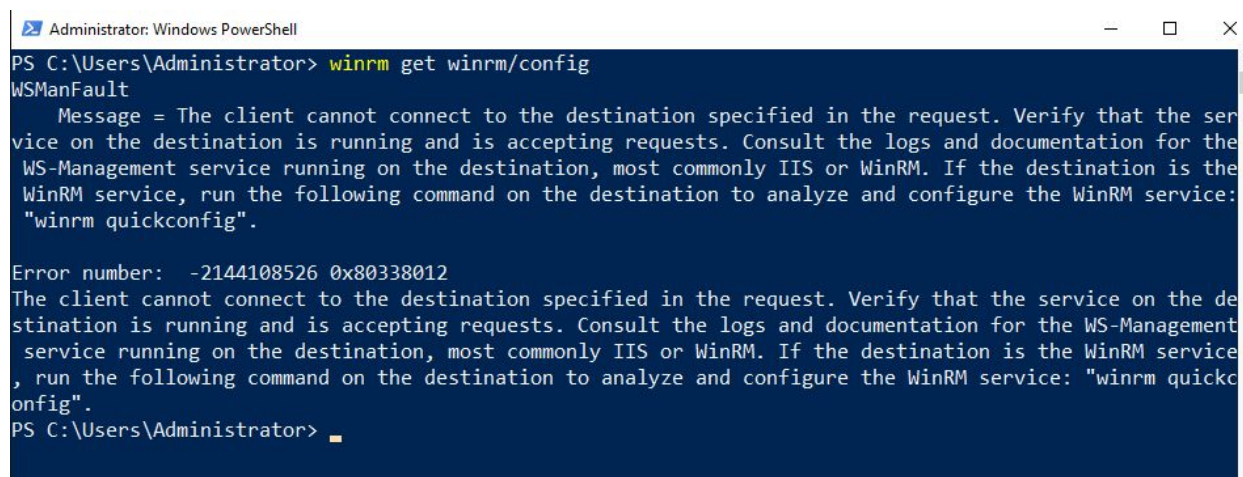
For help on a specific operation:
  winrm g[et] -?           Retrieving management information.
  winrm s[et] -?           Modifying management information.
  winrm c[reate] -?        Creating new instances of management resources.
  winrm d[ele]te -?        Remove an instance of a management resource.
  winrm e[numerate] -?     List all instances of a management resource.
  winrm i[nvoke] -?        Executes a method on a management resource.
  winrm id[entify] -?      Determines if a WS-Management implementation is
                           running on the remote machine.
  winrm quickconfig -?     Configures this machine to accept WS-Management
                           requests from other machines.
  winrm configSDDL -?      Modify an existing security descriptor for a URI.
  winrm helpmsg -?        Displays error message for the error code.

For help on related topics:
  winrm help uris          How to construct resource URIs.
  winrm help aliases       Abbreviations for URIs.
  winrm help config        Configuring WinRM client and service settings.
  winrm help certmapping   Configuring client certificate access.
  winrm help remoting      How to access remote machines.
  winrm help auth          Providing credentials for remote access.
  winrm help input         Providing input to create, set, and invoke.
  winrm help switches      Other switches such as formatting, options, etc.
  winrm help proxy         Providing proxy information.
PS C:\Users\Administrator>
```

We can notice, there are a lot of operations which we can perform using winrm command, such as configuration of winrm, extracting winrm service/client information, allowing clients to connect to the specific host etc..

Step 2: Try to fetch the **winrm/config**. This is the file where we would get all the information about current WinRM service.

Command: winrm get winrm/config

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command prompt shows the command "winrm get winrm/config" being executed. The output is a "WSManFault" error message. The message text is: "Message = The client cannot connect to the destination specified in the request. Verify that the service on the destination is running and is accepting requests. Consult the logs and documentation for the WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service, run the following command on the destination to analyze and configure the WinRM service: 'winrm quickconfig'." Below the message, the error number is displayed: "Error number: -2144108526 0x80338012". The prompt then shows the command "winrm quickconfig" being entered, but it is not yet executed.

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> winrm get winrm/config
WSManFault
    Message = The client cannot connect to the destination specified in the request. Verify that the service on the destination is running and is accepting requests. Consult the logs and documentation for the WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service, run the following command on the destination to analyze and configure the WinRM service: "winrm quickconfig".

Error number: -2144108526 0x80338012
The client cannot connect to the destination specified in the request. Verify that the service on the destination is running and is accepting requests. Consult the logs and documentation for the WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service, run the following command on the destination to analyze and configure the WinRM service: "winrm quickconfig".
PS C:\Users\Administrator> winrm quickconfig
```

We have received an error message as mentioned below.

“WSManFault

Message = The client cannot connect to the destination specified in the request. Verify that the service on the destination is running and is accepting requests. Consult the logs and documentation for the WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service, run the following command on the destination to analyze and configure the WinRM service: "winrm quickconfig".

Error number: -2144108526 0x80338012

The client cannot connect to the destination specified in the request. Verify that the service on the destination is running and is accepting requests. Consult the logs and documentation for the WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service, run the following command on the destination to analyze and configure the WinRM service: "winrm quickconfig".“

Step 3: The WinRM service is not running and hence we are not able to fetch the configuration file. In the error message it is suggesting a command to configure the winrm service. We will run the “**winrm quickconfig**” command to start the service and configure it.

Command: winrm quickconfig


```
Administrator: Windows PowerShell
PS C:\Users\Administrator> winrm quickconfig
WinRM is not set up to receive requests on this machine.
The following changes must be made:

Start the WinRM service.
Set the WinRM service type to auto start.

Make these changes [y/n]? 
```

It is prompting for a permission to start the WinRM service and set it to automatic startup mode. We will set "y" and press enter.

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> winrm quickconfig
WinRM is not set up to receive requests on this machine.
The following changes must be made:

Start the WinRM service.
Set the WinRM service type to auto start.

Make these changes [y/n]? y

WinRM has been updated to receive requests.

WinRM service type changed successfully.
WinRM service started.
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:
```

Once, we type "y" and hit enter. The WinRM service is now up and running. Next, it is prompting for permission to add a Firewall rule. Type "y" and press enter.

```
Enable the WinRM firewall exception.
Configure LocalAccountTokenFilterPolicy to grant administrative rights remotely to local users.

Make these changes [y/n]? 
```

```
Enable the WinRM firewall exception.  
Configure LocalAccountTokenFilterPolicy to grant administrative rights remotely to local users.  
Make these changes [y/n]? y  
WinRM has been updated for remote management.  
WinRM firewall exception enabled.  
Configured LocalAccountTokenFilterPolicy to grant administrative rights remotely to local users.  
PS C:\Users\Administrator>
```

We have successfully configured the firewall rules for the WinRM service.

Step 4: Try to fetch the **winrm/config**. This is the file where we would get all the information about current WinRM service.

Command: winrm get winrm/config

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> winrm get winrm/config
Config
  MaxEnvelopeSizekb = 500
  MaxTimeoutms = 60000
  MaxBatchItems = 32000
  MaxProviderRequests = 4294967295
  Client
    NetworkDelays = 5000
    URLPrefix = wsman
    AllowUnencrypted = false
    Auth
      Basic = true
      Digest = true
      Kerberos = true
      Negotiate = true
      Certificate = true
      CredSSP = false
    DefaultPorts
      HTTP = 5985
      HTTPS = 5986
    TrustedHosts
  Service
    RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
    MaxConcurrentOperations = 4294967295
    MaxConcurrentOperationsPerUser = 1500
    EnumerationTimeoutms = 240000
    MaxConnections = 300
    MaxPacketRetrievalTimeSeconds = 120
    AllowUnencrypted = false
    Auth
      Basic = false
      Kerberos = true
      Negotiate = true
      Certificate = false
      CredSSP = false
      CbtHardeningLevel = Relaxed
    DefaultPorts
      HTTP = 5985
      HTTPS = 5986
    IPv4Filter = *
    IPv6Filter = *
    EnableCompatibilityHttpListener = false
    EnableCompatibilityHttpsListener = false
```

We have received winrm service config information.

Step 5: Fetch only service configuration settings.

Command: winrm get winrm/config/service

Administrator: Windows PowerShell

```
PS C:\Users\Administrator> winrm get winrm/config/service
```

Service

```
RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
MaxConcurrentOperations = 4294967295
MaxConcurrentOperationsPerUser = 1500
EnumerationTimeoutms = 240000
MaxConnections = 300
MaxPacketRetrievalTimeSeconds = 120
AllowUnencrypted = false
Auth
    Basic = false
    Kerberos = true
    Negotiate = true
    Certificate = false
    CredSSP = false
    CbtHardeningLevel = Relaxed
DefaultPorts
    HTTP = 5985
    HTTPS = 5986
IPv4Filter = *
IPv6Filter = *
EnableCompatibilityHttpListener = false
EnableCompatibilityHttpsListener = false
CertificateThumbprint
AllowRemoteAccess = true
```

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

We can observe that by default winrm service is not allowing unencrypted traffic and winrm service uses 5985 port for HTTP traffic and 5986 for HTTPS traffic. Also, by default it allows the remote access and we can notice other basic information i.e number of maximum connections allowed, default ports, supported authentication type etc.

By default the winrm listener runs on HTTP, we can verify it by fetching the listener config details.

Command: winrm e winrm/config/listener


```
Administrator: Windows PowerShell
PS C:\Users\Administrator> winrm e winrm/config/listener
Listener
  Address = *
  Transport = HTTP
  Port = 5985
  Hostname
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint
  ListeningOn = 10.0.0.189, 127.0.0.1, ::1, fe80::950c:1e2b:a156:673a%4
PS C:\Users\Administrator> _
```

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

Step 6: We also need to allow clients (administration machines) to connect to the specific or all remote servers by modifying the “**TrustedHosts**”. Switch to the **Attacker machine** and configure TrustedHosts to allow all remote servers.

Note: You cannot modify the TrustedHosts file if the WinRM service is not running.

Starting the winrm service

Command: Start-Service winrm

```
PS C:\Users\Administrator> Start-Service winrm
PS C:\Users\Administrator>
PS C:\Users\Administrator>
```

The winrm service is now running.

Step 7: Modifying the TrustedHosts file and allowing all remote hosts to connect.

Command: Set-item wsman:localhost\client\trustedhosts -value *

```

PS C:\Users\Administrator> Set-item wsman:localhost\client\trustedhosts -value *

WinRM Security Configuration.
This command modifies the TrustedHosts list for the WinRM client. The computers in the TrustedHosts list might not be au
thenticated. The client might send credential information to these computers. Are you sure
that you want to modify this list?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator>

```

We have modified trustedhosts and allowed any remote servers.

Note: It's always a good practice to mention the IP address or a computer name of the remote server. Also, you don't need to keep the WinRM service running on the target machine. We can turn it off after modifying the TrustedHosts.

Creating a Demo User

Step 1: We will create a demo user on the remote machine (target machine) and the same user credentials we will use to execute commands from the client.

Commands:

```

$secureString = convertto-securestring "password_123321" -asplaintext -force
New-LocalUser "winrmdemo" -Password $secureString -FullName "WinRM Demo" -Description
"WinRM Demo Account"
Add-LocalGroupMember -Name 'Administrators' -Member 'winrmdemo'

```

```

Administrator: Windows PowerShell
PS C:\Users\Administrator> $secureString = convertto-securestring "password_123321" -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> Add-LocalGroupMember -Group "Administrators" -Member "winrmdemo"
PS C:\Users\Administrator>

```

We have successfully, created an user i.e “winrmdemo” and added the user in the administrators group.

Execute Commands On Remote Server

Step 1: We will execute the command using “winrs.exe” binary on the target machine.

“WinRS.exe is a windows remote management tool, allowing you to manage and execute commands & programs remotely.”

Note: Please check the target machine IP address. By running the “**ipconfig**” command.

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : ap-southeast-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::c434:975f:9938:9f83%4
    IPv4 Address. . . . . : 10.0.0.50
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1
PS C:\Users\Administrator>
```

Note: Please remember to replace below command with your valid target machine IP address.

Command: winrs.exe -r:http://10.0.0.50:5985 -u:winrmdemo -p:password_123321 ipconfig

```
PS C:\Users\Administrator> winrs.exe -r:http://10.0.0.50:5985 -u:winrmdemo -p:password_123321 ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : ap-southeast-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::c434:975f:9938:9f83%4
    IPv4 Address. . . . . : 10.0.0.50
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1
```

We have successfully executed the command on the remote server.

Step 2: We can also get the powershell.exe shell of the remote server by running the winrs.exe. Get the powershell shell.

Command: winrs.exe -r:http://10.0.0.50:5985 -u:winrmdemo -p:password_123321 powershell.exe

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> winrs.exe -r:http://10.0.0.50:5985 -u:winrmdemo -p:password_123321 powershell.exe
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\winrmdemo>

PS C:\Users\winrmdemo> whoami
whoami
ec2amaz-d9nv5hs\winrmdemo
PS C:\Users\winrmdemo> █
```

We now have full access to the remote server.

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>)
- WinRS.exe (<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/winrs>)