# Configuring WINRM Guide

## Version 1.0

# TABLE OF CONTENTS

# 1 Using WinRM in Domain environment

This is the best option as you can leverage Kerberos for authentication. This allows you to pass through your Active Directory credentials.

## 1.1 Enable WinRM in Group Policy

Using Group Policy, applied to the default policy, or a GPO you create, navigate to and enable the following:

- Administrative Templates
  - ↳ Windows Components
    - ↳ Windows Remote Management (WinRM)
      - ↳ WinRM Service
- ENABLE - Allow remote server management through WinRM
  - ↳ Set the IPv4 range of your environment or allow all ranges using the wildcard *
- ENABLE - Allow unencrypted traffic

You may decide to go down the HTTPS path to encrypt the traffic, but this will require more effort to secure the sessions. Get it working first, then tackle the HTTPS improvement so you know it works.

## 1.2 Windows Firewall

If you are using the Windows Firewall in your environment, then you will need to add a rule to allow inbound WinRM traffic over HTTP, or HTTPS.

- Windows Settings
  - ↳ Security Settings
    - ↳ Windows Firewall with Advanced Security
      - ↳ Inbound Rule
- Add the Predefined Rule – Windows Remote Management (HTTP-in)
- Use only Domain and Private
- Disable Public access

Here are some articles on using Group Policy to enable WinRM in your environment:

- http://www.mustbegeek.com/how-to-enable-winrm-via-group-policy/
- https://docs.microsoft.com/en-us/windows/desktop/winrm/installation-and-configuration-for-windows-remote-management
- https://www.techrepublic.com/article/how-to-enable-powershell-remoting-via-group-policy/

### 1.3 START THE WINRM SERVICE

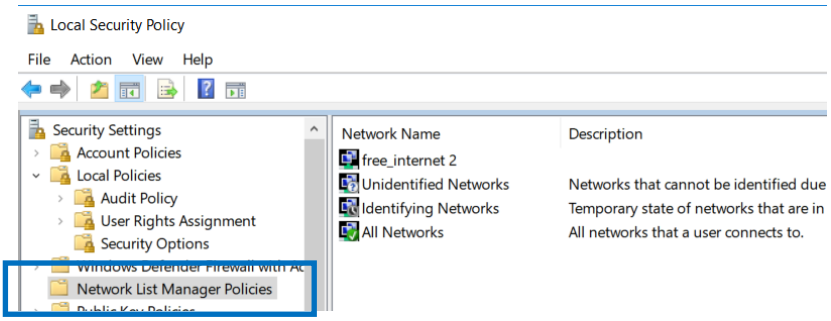
1. Navigate to Computer Configuration
    - ↳ Preferences
        - ↳ Control Panel Settings
            - ↳ Services
- Right-click Services and choose New > Service
- Choose Automatic (Delayed Start) as startup type
- Pick WinRM as the service name
- Set “Start service” as the action
- Click OK to save the change.

## 2 USING WINRM IN A NON-DOMAIN ENVIRONMENT

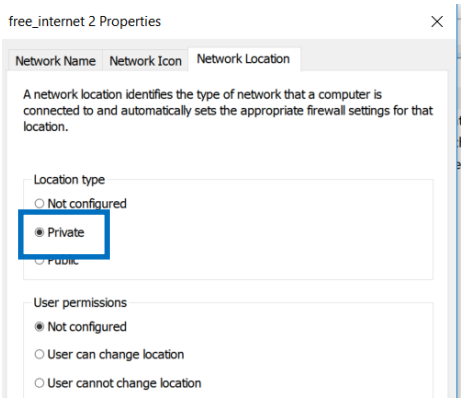
If you are not in a domain, or using a test or malware lab, then there are some configuration tips to help you get WinRM up and running.

On each system, the Host and Targets do the following to configure WinRM:

1. Open an Administrator CMD window type the following:
    - “Winrm quickconfig” or “winrm qc”
        - Answer “Yes”
        - If you get an error it is most likely due to your network being set to “Public”, change it to “Private” from the next two steps below.
2. Check and adjust the policy you have configured for your network card
    - Make sure your network profile is ***NOT Public*** so either Private or Domain, WinRM will not work over the Public setting. In PowerShell type the following:
        - Get-NetConnectionProfile (Not available for Win 7 running PS 5.1, use the GUI)
        - Get the “InterfaceIndex”
    - In PowerShell type the following:
        - Set-NetConnectionProfile -InterfaceIndex <index number> -NetworkCategory Private
        - Or do it thru the GUI, Open “Local Security Policy”
        - Select “Network List Manager Policies”




- Select your network connection
- Select “Network Location”
- Change the “ Location Type” to “Private”




- You may also adjust the network policy registry key manually
  - *HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles"*
  - Select your network card
  - Change the value to '1' for Private, '0' *zero* for Public
  - Restart Windows for this to take affect

## 2.1 CONFIGURE WINRM DETAILS

On each system you want to run WinRM on you will need to make sure WinRM traffic over HTTP is open for incoming traffic and set the WinRM details. In an Administrator CMD window type the following commands:

- netsh advfirewall firewall add rule name="WinRM-HTTP" dir=in localport=5985 protocol=TCP action=allow
- winrm set winrm/config/service @{AllowUnencrypted="true"}
- winrm set winrm/config/service/auth @{Kerberos="false"}
- winrm set winrm/config/service/auth @{Negotiate="true"}
- winrm set winrm/config/service @{EnableCompatibilityHttpListener="true"}

Repeat this on each system that you want to run WinRM against.

## 2.2 ON THE HOST YOU WILL RUN ARTHIR ON

You will need to trust each system that you want to control from your host system, the one you launch ARTHIR from. Set the following items:

1. winrm s winrm/config/client @{TrustedHosts="<systemname_you_want_to_trust>"}
   a. winrm s winrm/config/client @{TrustedHosts="PC1, PC2, PC3, …"}
2. Create a hosts entry to resolve the IP address to the system name
   a. C:\Windows\systems32\drivers\etc\hosts

# 3 TESTING WINRM

Before using WinRM, it is best to test it, and/or have a test to ensure it is working in your environment, or when you run into issues with a system. The following are some simple steps you can use to test that WinRm is configured and working. All of these command should work if WinRM is configured correctly.

## 3.1 GET THE WINRM CONFIGURATION.

Run this on each system to check the configuration, if set by Group Policy, then on one system, or any system that is having issues.

- Winrm get winrm/config/service
- Review all the settings mentioned above

You should see something like the following:

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

## 3.2 Is the system listening for WinRM traffic?

- winrm e winrm/config/listener or
- winrm enumerate winrm/config/listener

You should see something like the following with your IP address(es)

```
D:\>winrm e winrm/config/listener
Listener
    Address = *
    Transport = HTTP
    Port = 5985
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 127.0.0.1, 169.254.10.172, 192.168.1.106,
```

## 3.3 Check if a remote system is responding, run from a PowerShell window

- Open an administrative PowerShell console
- Test-WSMan -ComputerName <***IP or host name***>

You should see the following:

```
wsmid           : http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd
ProtocolVersion : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
ProductVendor   : Microsoft Corporation
ProductVersion  : OS: 0.0.0 SP: 0.0 Stack: 3.0
```

## 3.4 Test if you can authenticate and create a manual remote PS-Session

- $Credential = Get-Credential <username>
  - ↳ You should get a Auth screen to enter your password securely
- Enter-PSSession ***<computername>*** -Authentication Negotiate -Credential $Credential

  Or

- Enter-PSSession ***<computername>*** -Authentication Negotiate -Credential ***<username>***

You should see a remote console prompt of the target system

- **[TARGET_SYSTEMNAME]: PS C:\Users\Bob\Documents**
- **Exit-PSSession** to quit

## 3.5 Additional Testing resources

- Here are some additional commands you can run to test WinRM
  - ↳ https://blogs.te_chnet.microsoft.com/otto/2007/02/09/a-few-good-vista-ws-man-winrm-commands/

# 4 POWERSHELL REMOTING

You can also use PowerShell Remoting to manually create a session on a system to push files, run LOG-MD-Pro, and retrieve files outside ARTHIR. This can be useful when you are having issues or want to perform one of tasks.

- Enter-PSSession M-PC -Authentication Negotiate -Credential MG
- $s = New-PSSession -ComputerName M-PC -Authentication Negotiate -Credential MG
- Invoke-Command -Session $s {Get-Process}
- Invoke-Command -Session $s {C:\'Program Files'\LMD\Log-MD-Pro.exe -hb -o 'C:\Program Files\LMD\Results'}
- Copy-Item -Path "C:\Program Files\LMD\Results\Reg_Compare.txt" -Destination "D:\ARTHIR_LOG-MD" -FromSession $s
- Exit-PSSession

- $s1 = New-PSsession -ComputerName <computername> -SessionOption (New-PSSessionOption -NoMachin
- eProfile) -ErrorAction Stop -Authentication Negotiate -Credential MG
- Enter-PSSession -Session $s1
- Invoke-Command -ScriptBlock {Get-Process} -Session $s1
- Invoke-Command -file file.ps1 -Session $s1
- Copy-Item -Path "C:\Users\bob\Downloads\maliciousdoc.docx" -Destination "D:\triage_files" -FromSession $s1I
- Invoke-Command -ScriptBlock { Get-Process} -Session $s1
- Get-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

# 5 REVISION HISTORY

| Date | Revision | Description |
|---|---|---|
| 03/15/2019 | Ver 1.0 | Initial release |
| | | |