

5nine Virtual Firewall 2.0 (v-Firewall, Beta)

Ver. 2.0 Getting Started Guide

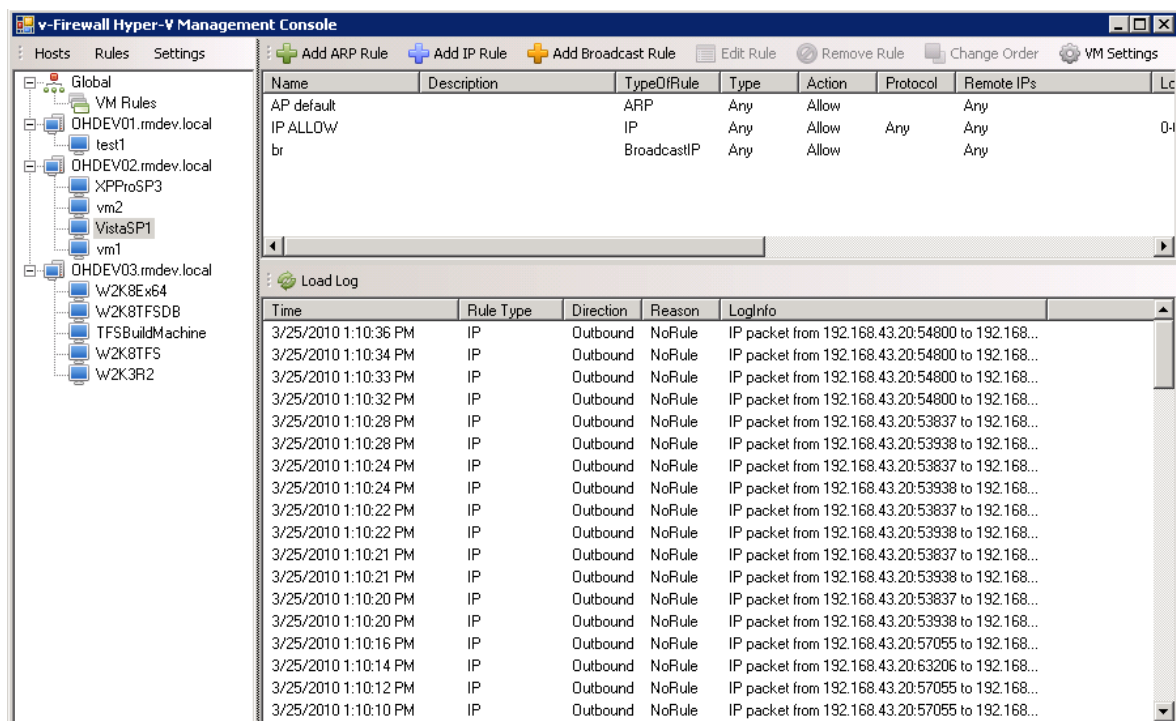
Summary

5nine Virtual Firewall (vFW2) is a Virtual Infrastructure monitoring tool with an ability to define network traffic rules for Hyper-V Virtual Machines and harden your Virtual Infrastructure from Security perspective. Both programmatically – using PowerShell API and via Management Console. Virtual Firewall allows reviewing network traffic logs for each of the monitored Virtual machines and generates related reports. Special Security Heartbeat service checks if firewall rules are enforced, and powers Virtual machine down, of network filter is not communicated.

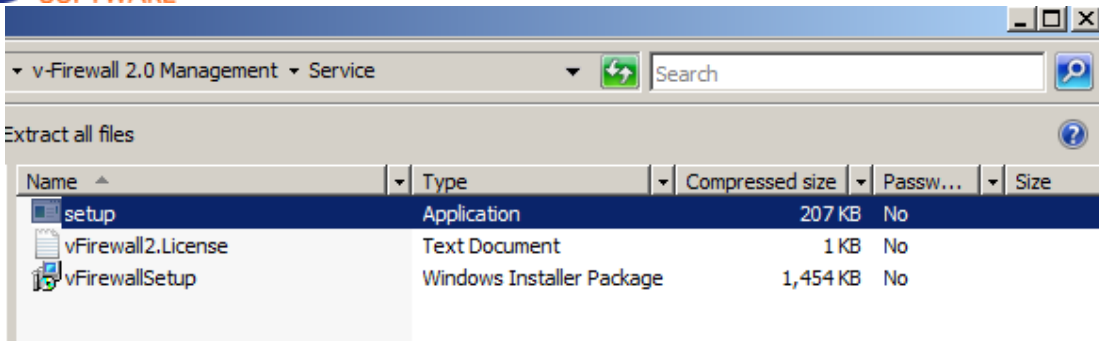
Version 2.0 of 5nine Virtual Firewall monitors and controls the traffic between Hyper-V Virtual machines and between Virtual machines and external network.

Features and Benefits:

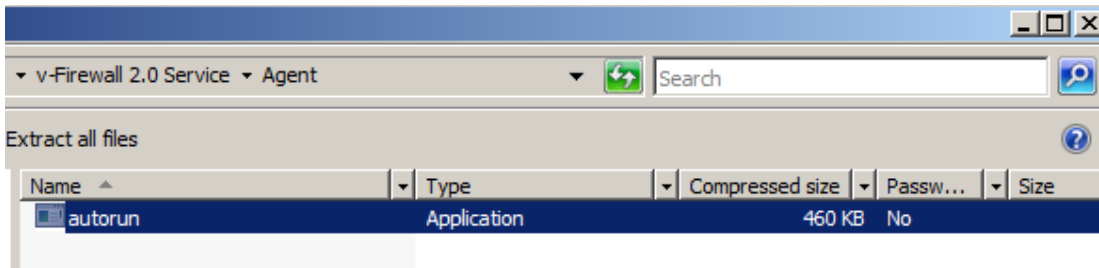
Simple installation. 5nine Virtual Firewall has 2 components that need to be installed. One – is intuitive Management interface (DLL) that supports **PowerShell API (described below)** to set and change traffic rules. Another – is ‘Security heartbeat’ service that needs to be installed on VM similar to Hyper-V Integration service. Management API also has a simple to use GUI application that allows setting the traffic rules between the virtual machines and external network. Management interface can be installed either on a server or Virtual machine, and allows System Administrator to access rules, logs and reports:



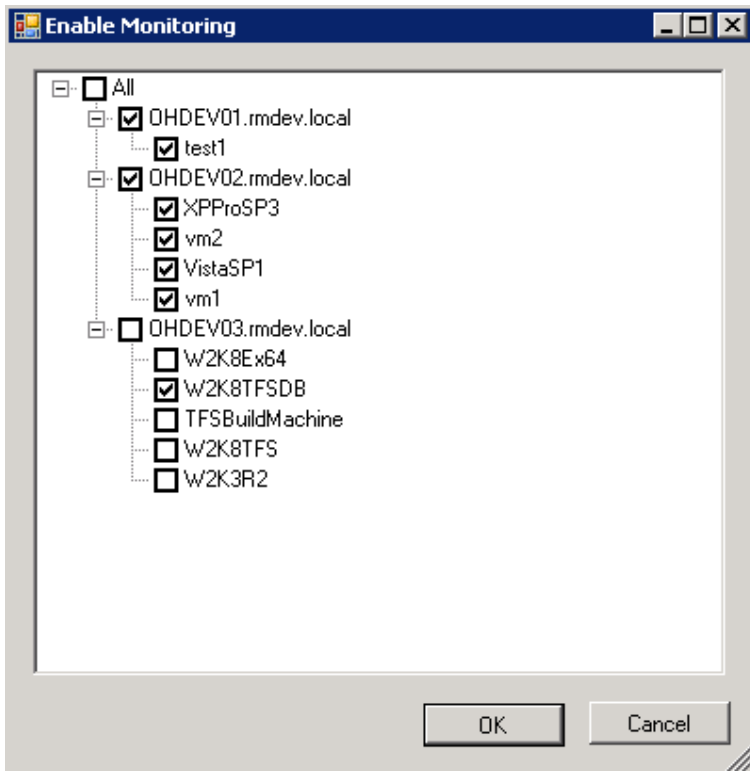
To setup Management interface (DLL and Management GUI application) – admin needs to run setup.exe application from the downloaded **v-Firewall 2.0 Management** archive on the server or VM that matches v-Firewall 2.0 ‘System Requirements’, and use appropriate license when prompted:



To setup v-Firewall Security Heartbeat service on the Virtual machines – administrator needs to launch autorun.exe on the VM being provisioned – either directly or through SC VMM template:

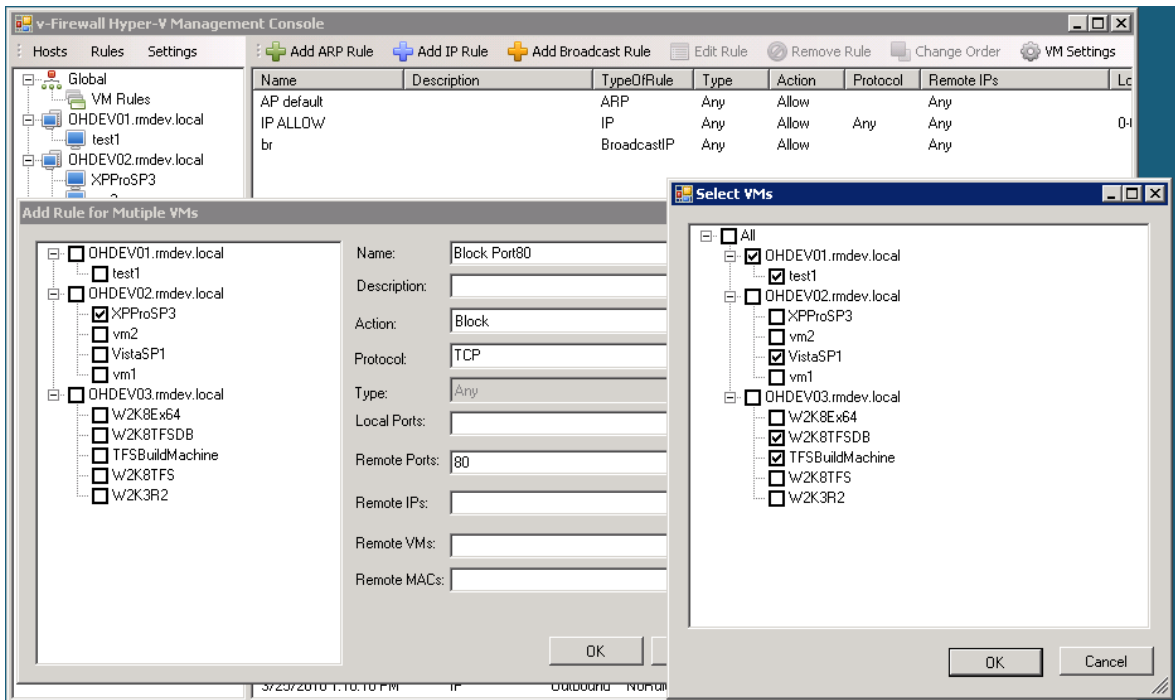
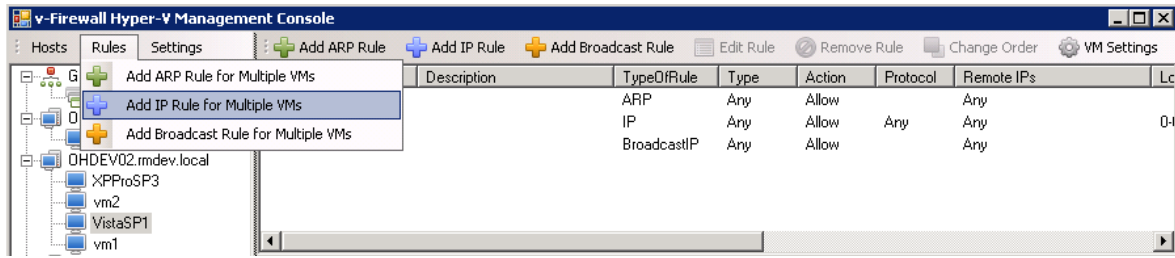


During and after setup of the Management interface and GUI application – administrator needs to specify which hosts and VMs will be controlled and monitored by v-Firewall 2.0:

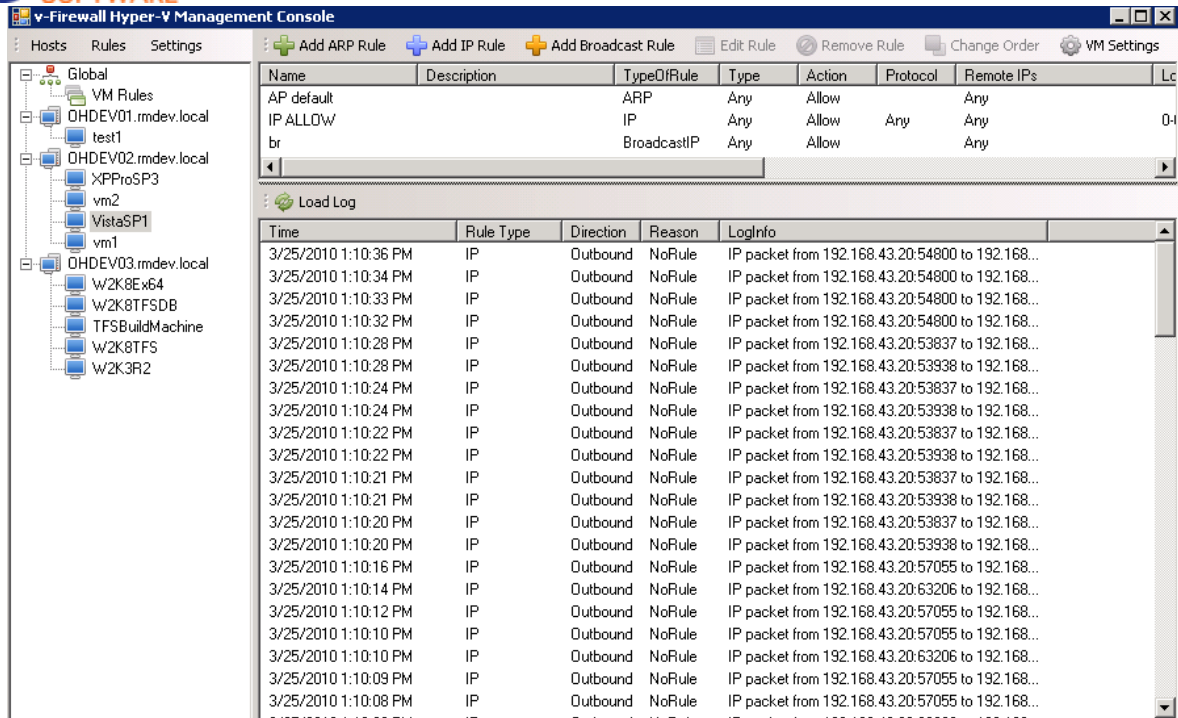


as well as which VMs will be controlled by 'Security Heartbeat Service'. The latter service will stop (or pause) the Virtual machine if the network filter is not communicated, and therefore current rules system may be compromised.

📁 **Firewall Rules** - Using intuitive PowerShell API (described below) or a simple GUI an administrator can define various firewall rules to allow or restrict various types of network traffic coming from the external network to Hyper-V Virtual machines, and between Virtual Machines.



📁 **Network Statistics and Logs** - Network activity data is collected by 5nine Virtual Firewall into a database or flat files (optionally); 'Load Log' pane needs to be clicked to load the current Firewall logs:



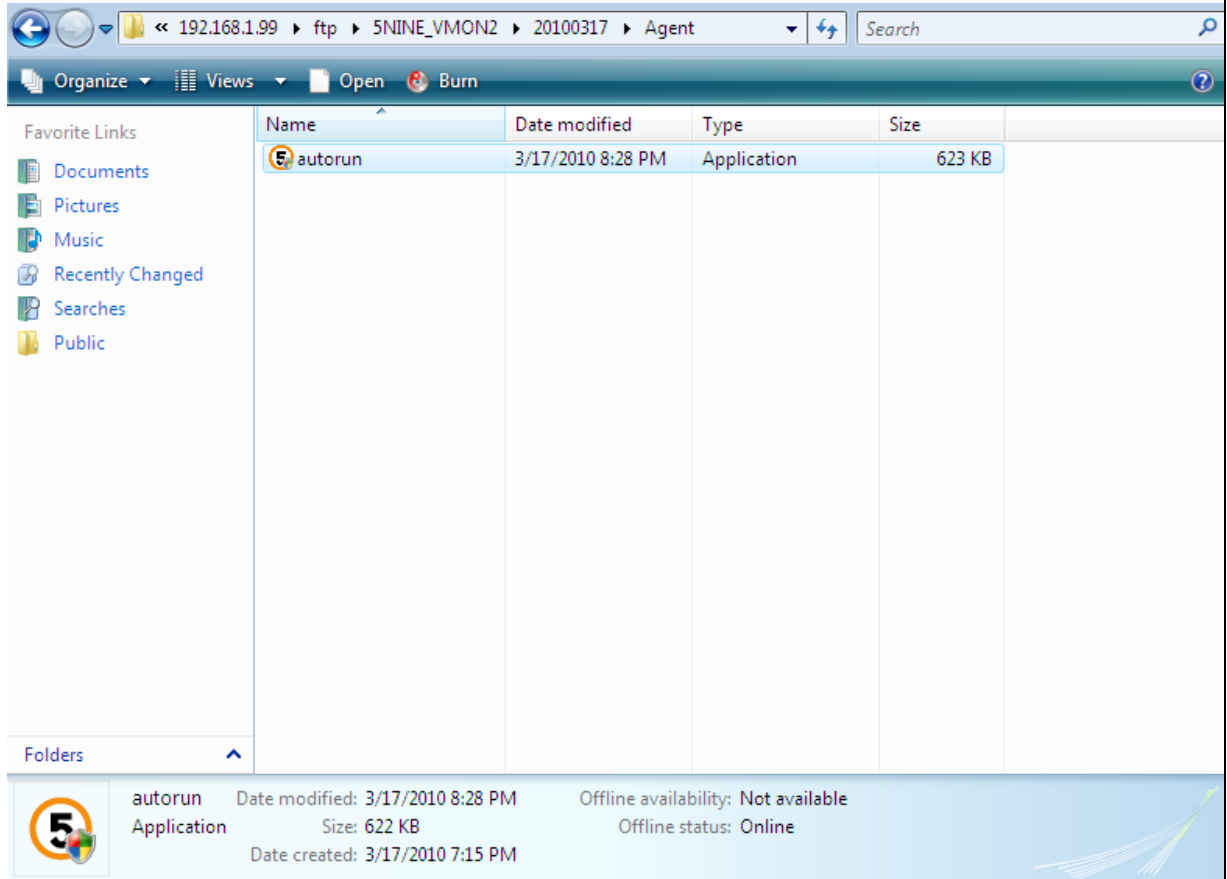
System Requirements.

- Windows 2008 server with Hyper-V;
- XP Pro SP3, Vista SP1 (Business, Enterprise or Ultimate editions), Win 2003 R2 SP2, Win 2008 server or later virtual machine(s), x64 or x86 for Management API and GUI application; v- Firewall Web Console Virtual machine needs to be on the same Hyper-V host where the service and the driver get installed;
- .NET 3.5 Sp1 or higher on the Server or VM that hosts Management API and/or GUI application;
- SQL 2008 Express edition on Management server/VM (in case DB logging is required).Web Console Virtual machine. It gets installed by v-Firewall setup.
- MS PowerShell IIS


Installation

I. vFW2 Agent Service installation.

1. Run autorun.exe from vFW2 agent setup directory



2. Confirm installation

 Do you want to setup v-Firewall 2.0 Agent?

Yes

No

3. Confirm reboot on setup prompt

You must restart your system for the configuration changes made to 5nine v-Firewall 2.0 Agent to take effect. Click Yes to restart now or No if you plan to manually restart later.

Yes

No

Installing vFW2 agent on new VMs

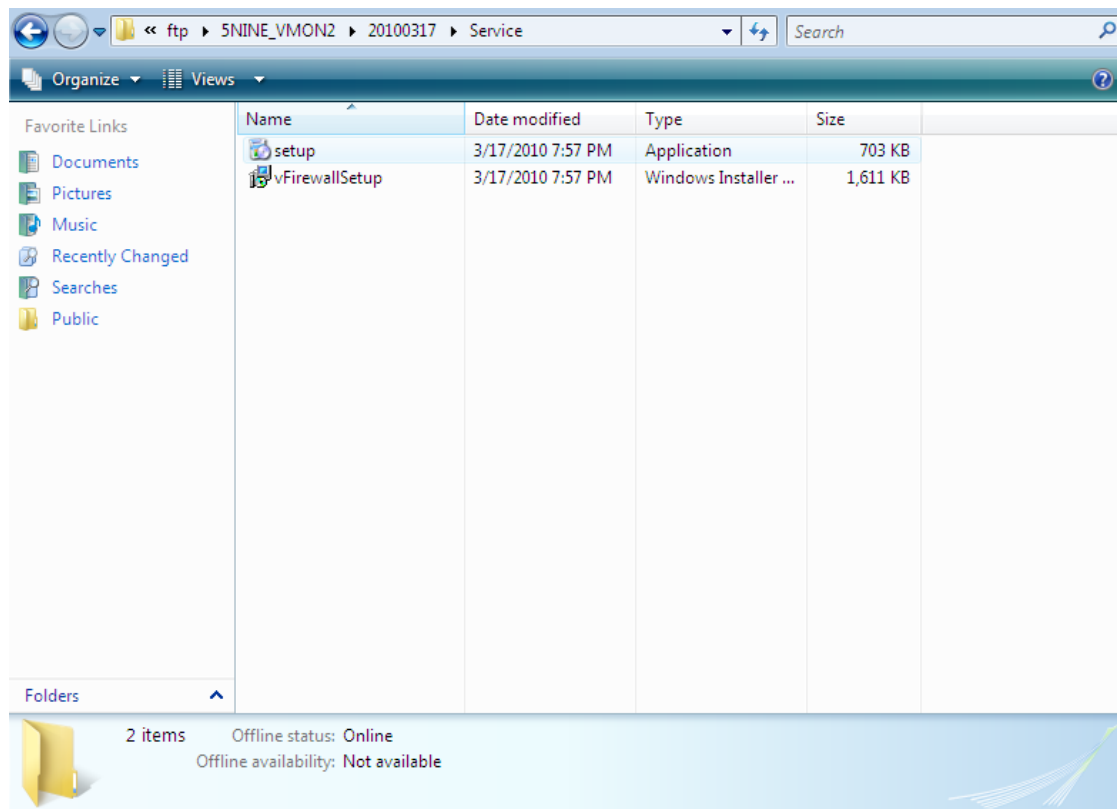
To deploy vFW2 agent on each new VM you can create VMM template from VM with vFW2 already installed.

II.) vFW2 management API and management console installation

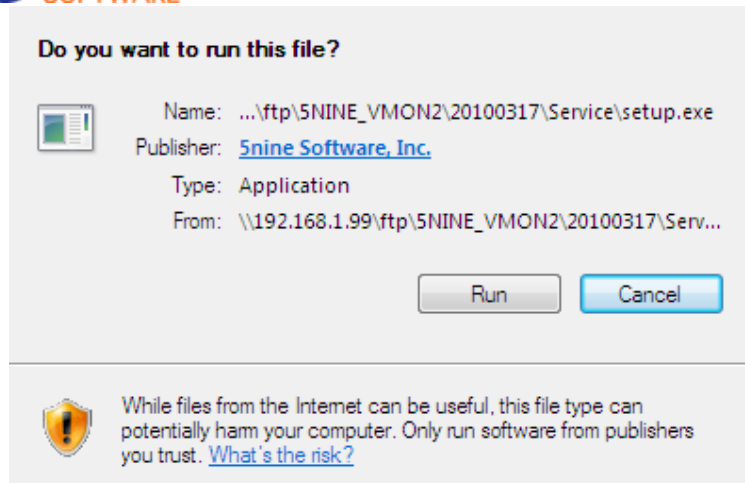
Requirements for Management server or VM (Xp SP3, Vista, Windows7, Server 2008/R2 or later, x 86 or x 64):

- .NET v3.5 SP1
- MS PowerShell

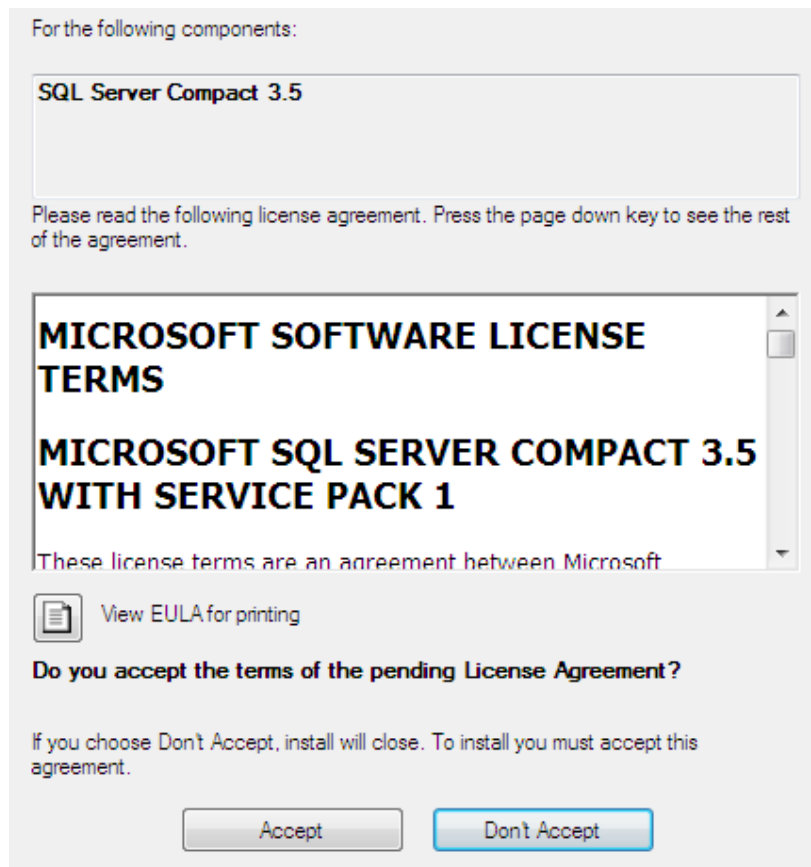
1. Run setup.exe from vFW2 service setup directory



2. Confirm installation



3. Confirm installation



4. Accept EULA and click 'Next'

Welcome to the 5nine v-Firewall 2.0 Setup Wizard



The installer will guide you through the steps required to install 5nine v-Firewall 2.0 on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Cancel

< Back

Next >

5. Select installation directory

Select Installation Folder



The installer will install 5nine v-Firewall 2.0 to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:

Browse...

Disk Cost...

Install 5nine v-Firewall 2.0 for yourself, or for anyone who uses this computer:

 Everyone Just me

Cancel

< Back

Next >

6. Confirm installation

Confirm Installation



The installer is ready to install 5nine v-Firewall 2.0 on your computer.

Click "Next" to start the installation.

7. Set vFW2 service account

Set account for vFirewall service

Set Service account

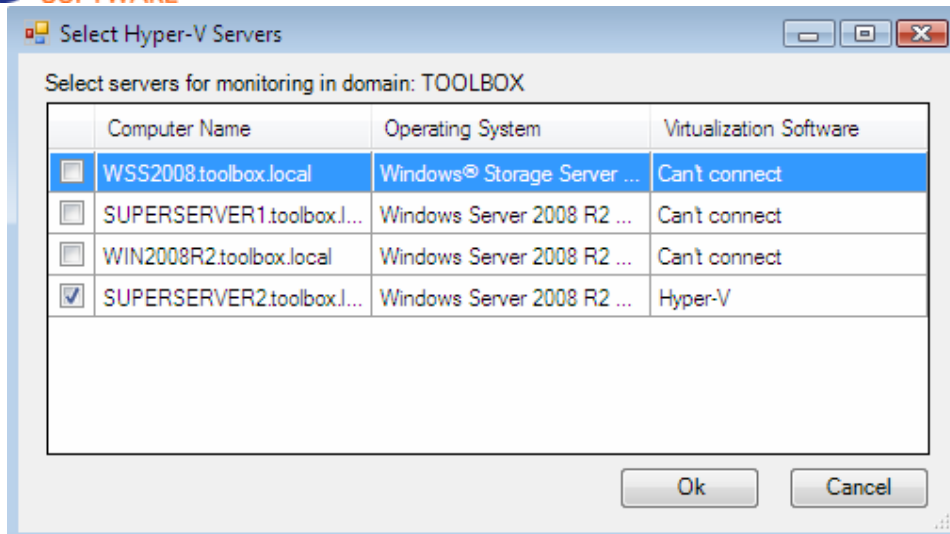
Account must have domain administrator rights for domain:
TOOLBOX

Username:

Password:

Confirm Password:

8. Select Hyper-V servers for monitoring



9. Complete installation

Installation Complete



5nine v-Firewall 2.0 has been successfully installed.

Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

Cancel

< Back

Close

vFW2 Configuration file and PowerShell API

- v-Firewall vFW2 service configuration file

%Program Files%\5nine\5nine v-Firewall 2.0\5Nine.vFW.vFWService.exe.cfg

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<configuration>
```

```
<configSections>
```

```
<section name="MonitoredHosts"
```

```
type="FiveNine.vFW.vFWServiceHelpers.MonitoredHostsConfigurationSection,
```

©2009 5nine Software, Inc. The information contained herein is subject to change without notice. 5nine shall not be liable for technical or editorial errors or omissions contained herein.

```
5Nine.vFW.vFWServiceHelpers" />
</configSections>
<MonitoredHosts>
  <host name="host1" />
  <host name="host2" />
  .....
  <host name="hostN" />
</MonitoredHosts>
<appSettings>
  <add key="HeartBeatPeriod" value="5000" />
  <add key="AttemptsBeforePause" value="4" />
  <add key="LogFile" value="vFirewall2.log" />
  <add key="LogLevel" value="Information" />
</appSettings>
</configuration>
```

Get the list of VM machines

The sample of Power Shell script to get GUIDs of VM machines from the specified host

```
$VMs = get-wmiobject -computerName $hyper -namespace "root\virtualization" -query "SELECT * FROM
Msvm_ComputerSystem WHERE Caption Like '%virtual%'"
foreach ($VM in $VMs)
{
    write-host "===== "
    write-host "VM Name: " $VM.ElementName
    write-host "VM GUID: " $VM.Name
}
}
```

- API description

Add-ARP-Rule

```
Add-ARP-Rule -VMId <Guid> -Name <String> [-Description <String>] [
-Type <String>] -Action <RuleAction> [-IPAddresses <String>] [-VMs
<String>] [-MACAddresses <String>] [-Priority <Int32>] [-ApplyNow
] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningA
ction <ActionPreference>] [-ErrorVariable <String>] [-WarningVaria
ble <String>] [-OutVariable <String>] [-OutBuffer <Int32>]
```

Add-BroadcastIP-Rule

```
Add-BroadcastIP-Rule -VMId <Guid> -Name <String> [-Description <St
ring>] [-Type <String>] -Action <RuleAction> -Protocol <String> [-
LocalPorts <String>] [-RemotePorts <String>] [-IPAddresses <String
>] [-VMs <String>] [-MACAddresses <String>] [-Priority <Int32>] [-
ApplyNow] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-
WarningAction <ActionPreference>] [-ErrorVariable <String>] [-Warn
ingVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]
```

Add-IP-Rule

```
Add-IP-Rule -VMId <Guid> -Name <String> [-Description <String>] [-
Type <String>] -Action <RuleAction> -Protocol <String> [-LocalPort
s <String>] [-RemotePorts <String>] [-IPAddresses <String>] [-VMs
<String>] [-MACAddresses <String>] [-Priority <Int32>] [-ApplyNow]
[-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAc
tion <ActionPreference>] [-ErrorVariable <String>] [-WarningVariab
le <String>] [-OutVariable <String>] [-OutBuffer <Int32>]
```

Set-Heartbeat

Set-Heartbeat -VMId <Guid> -Enable 1|0 [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Set-VMMonitoring

Set-VMMonitoring -VMId <Guid> -Enable 1|0 [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Get-Heartbeat

Get-Heartbeat [-VMId <Guid>] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Get-LogRecords

Get-LogRecords -VMId <Guid> [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Get-Rules

Get-Rules [-Id <Guid[]>] [-VMId <Guid>] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Get-VMIPMAC

Get-VMIPMAC -VMId <Guid> [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Get-VMMonitoring

Get-VMMonitoring [-VMId <Guid>] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Remove-Rule

Remove-Rule -Id <Guid> [-ApplyNow] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Reset-Rules

Reset-Rules -VMId <Guid> [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Set-Rule

Set-Rule -Id <Guid> [-Name <String>] [-Description <String>] [-Type <String>] [-Action <RuleAction>] [-Protocol <String>] [-LocalPorts <String>] [-RemotePorts <String>] [-IPAddresses <String>] [-MACAddresses <String>] [-VMs <String>] [-Priority <Int32>] [-ApplyNow]

] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

Set-VMIPMAC

Set-VMIPMAC -VMId <Guid> [-IPAddresses <String>] [-MACAddresses <String>] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-WarningAction <ActionPreference>] [-ErrorVariable <String>] [-WarningVariable <String>] [-OutVariable <String>] [-OutBuffer <Int32>]

 **How to Set Firewall rules in vFW2**

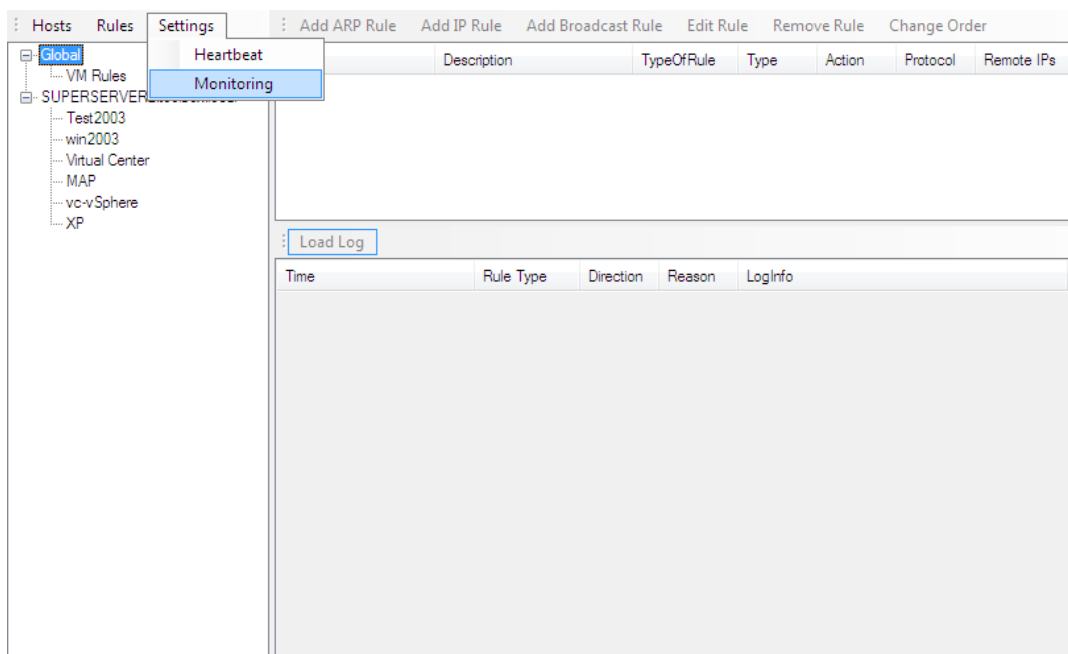
Sample scenario to allow RDP access to VM

Launch Power Shell and input the following commands:

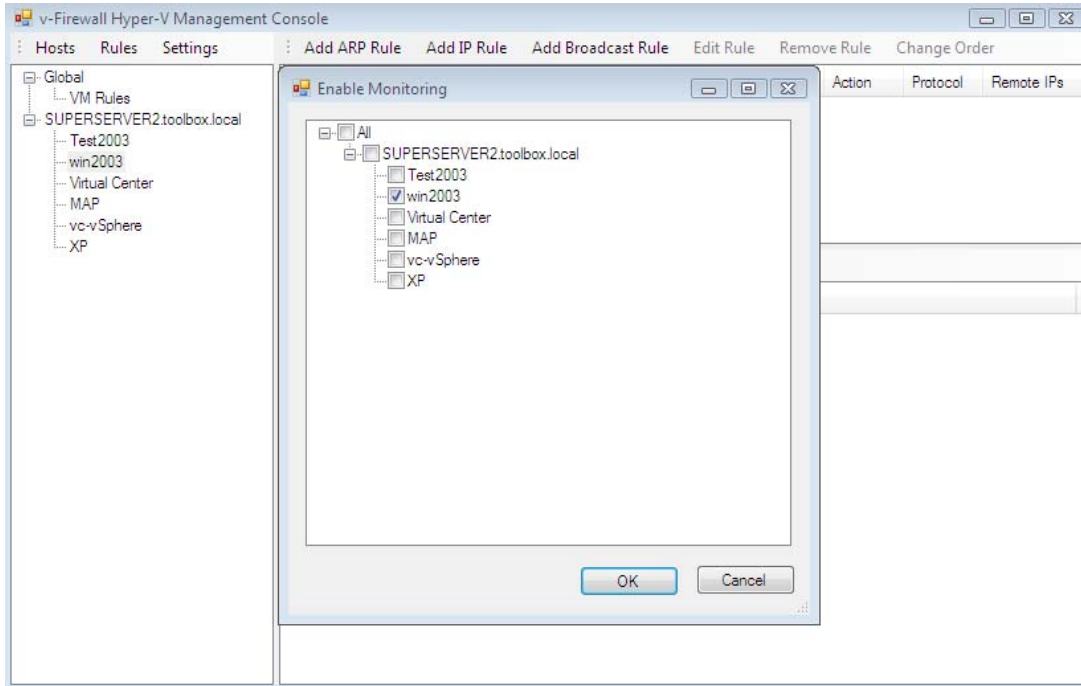
1. Add-PSSnapIn RulesAPI – *add vFW2 API snap-in to Power Shell*
2. Get VM GUIDs by applying sample PS script
3. Set-VMMonitoring -VMId <Guid> -Enable 1 - *set VM to vFW2 monitoring*
4. Add-ARP-Rule -VMId <Guid> -Name <String> -Action Allow - *add ARP allow rule to VM*
5. Add-IP-Rule -VMId <Guid> -Name "Allow RDP" -Action Allow -Protocol TCP -LocalPort 3389 – *add IP rule to allow incoming packets to 3389 port (RDP)*
6. Get-LogRecords -VMId <Guid> - *display blocked events (not to 3389 port)*

The same scenario with vFW2 management console

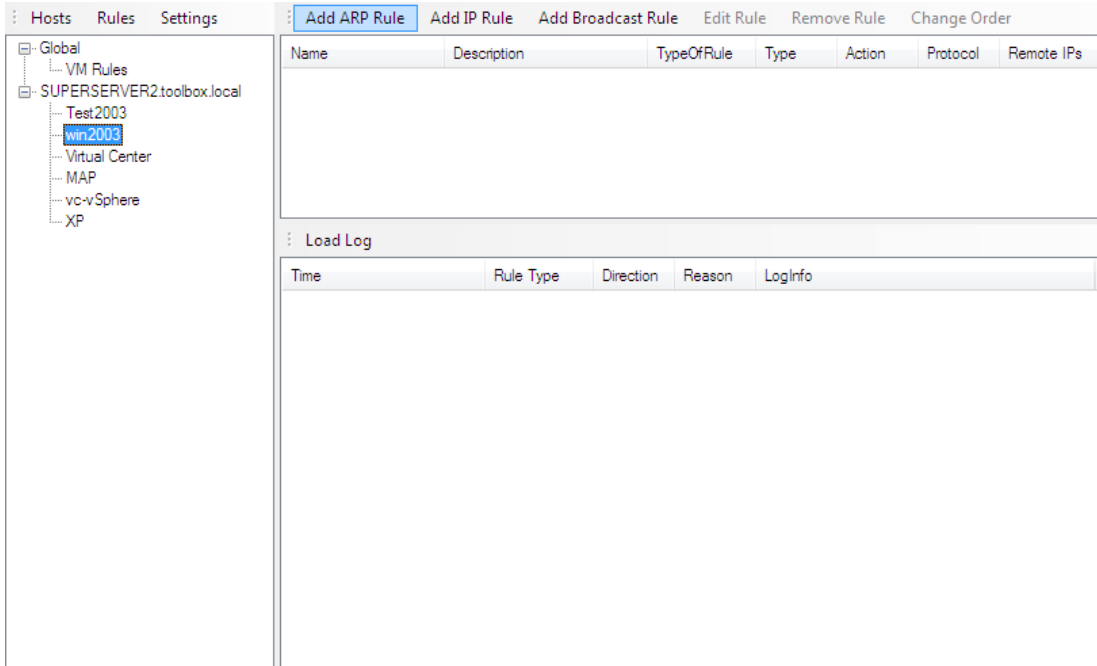
1. Set VM machines for monitoring



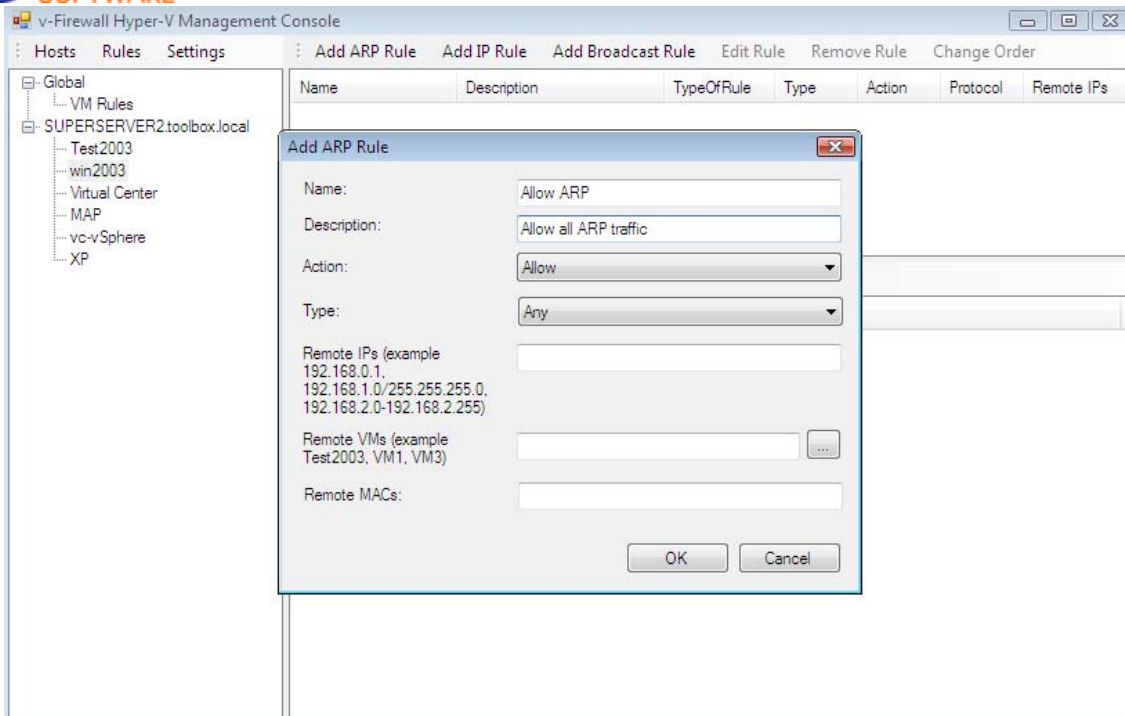
2. Set VM machines for monitoring



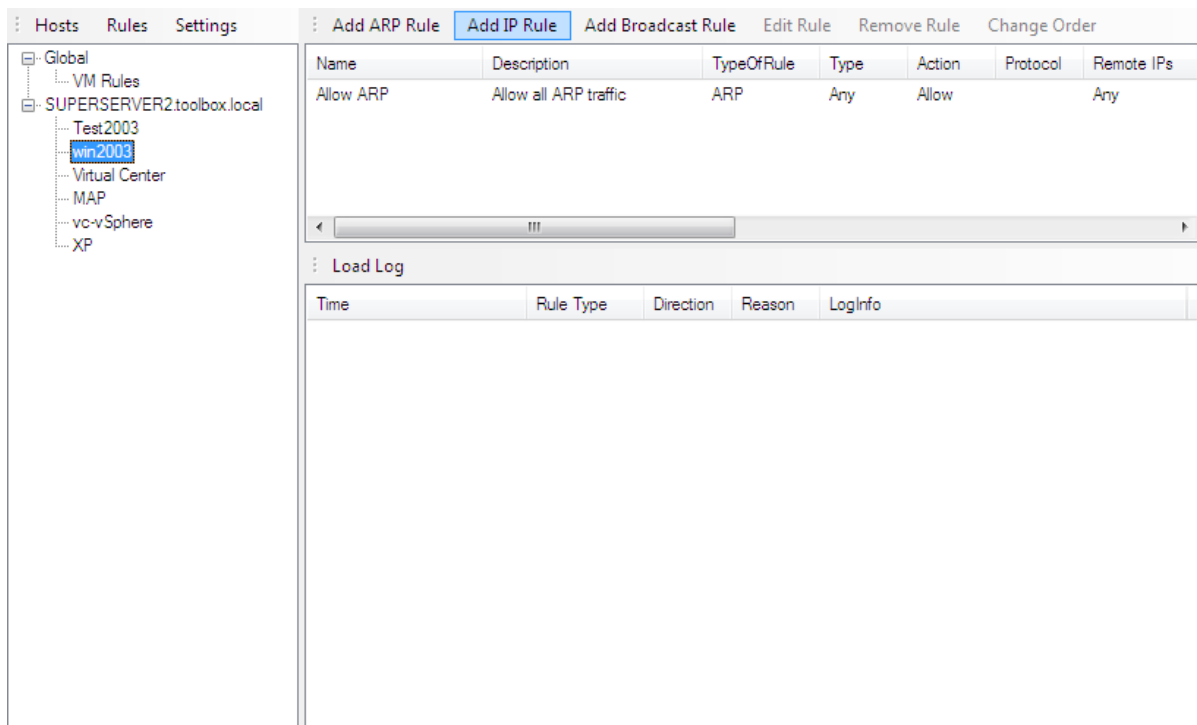
3. Set ARP rule to allow all ARP traffic



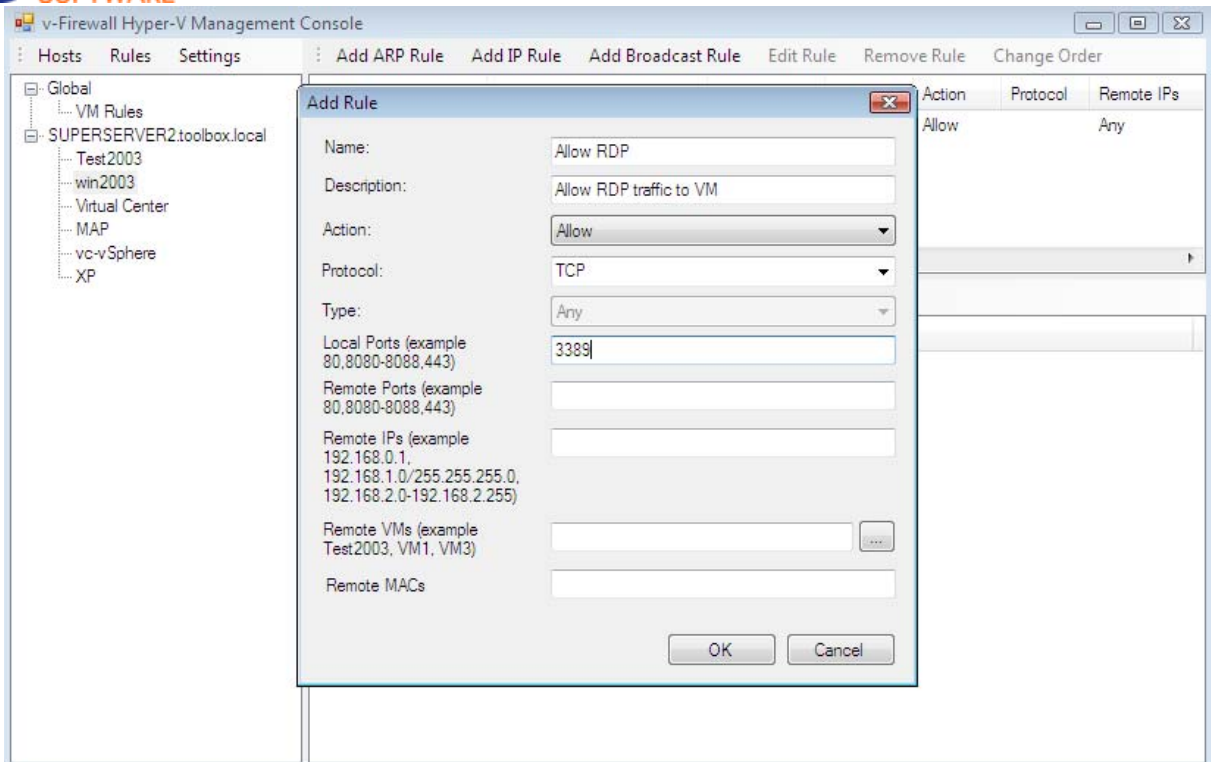
4. Set ARP rule to allow all ARP traffic



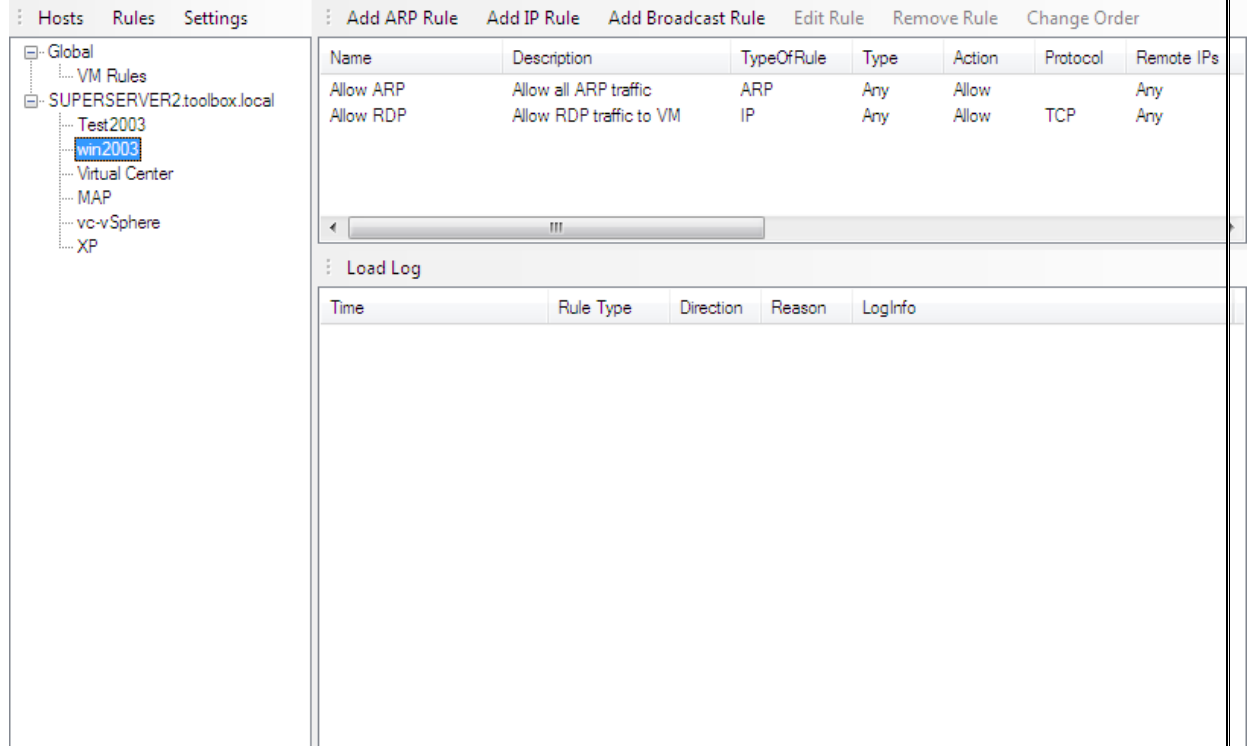
5. Set IP rule to allow inbound traffic to port 3389



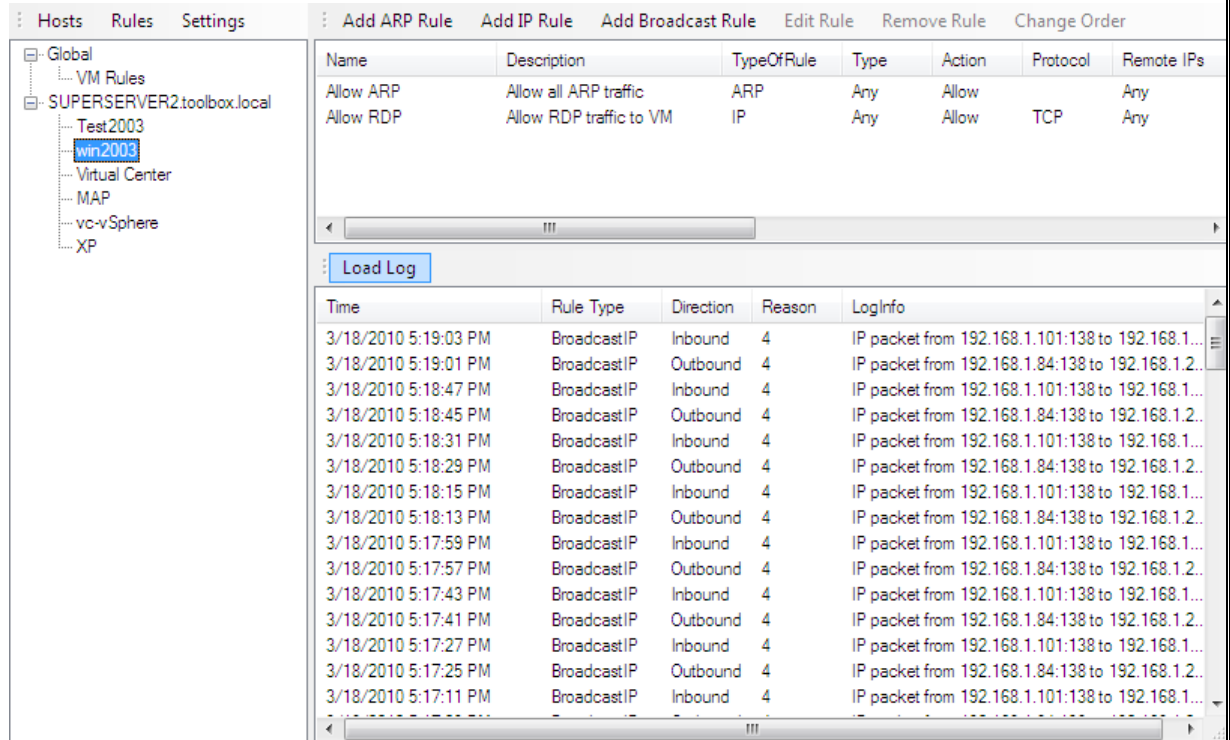
6. Set IP rule to allow inbound traffic to port 3389



7.



8. Load log records from VM



Sample scripts.

Basic sample script to allow 80 port on Win2003 VM:

1. \$VMs = get-wmiobject -computerName superserver2 -namespace "root\virtualization" -query

"SELECT * FROM Msvm_ComputerSystem WHERE Caption Like '%virtual%'"

foreach (\$VM in \$VMs)

{

 write-host "=====

 write-host "VM Name: " \$VM.ElementName

 write-host "VM GUID: " \$VM.Name

}

Press Enter two times . Get GUID for Win2003 - it is 7D2FDDAB-3B41-4FB1-99E0-CDD633453FCA

2. Set-VMMonitoring -VMId 7D2FDDAB-3B41-4FB1-99E0-CDD633453FCA -Enable 1

3. Add-ARP-Rule -VMId 7D2FDDAB-3B41-4FB1-99E0-CDD633453FCA -Name "Allow ARP" -Action Allow

4. Add-IP-Rule -VMId 7D2FDDAB-3B41-4FB1-99E0-CDD633453FCA -Name "Allow RDP" -Action Allow - Protocol TCP -LocalPort 80

5. Get-LogRecords -VMId 7D2FDDAB-3B41-4FB1-99E0-CDD633453FCA

The same scenario for RDP access is described in QSG document.

Sample common scenarios using Management console GUI**a.) Allowing FTP, DHCP**

1. allow ARP to/from gateway only (192.168.1.1)

Add ARP Rule

Name: ARP allow gateway

Description:

Action: Allow

Type: Any

Remote IPs (example 192.168.0.1, 192.168.1.0/255.255.255.0, 192.168.2.0-192.168.2.255) 192.168.1.1

Remote VMs (example Test2003, VM1, VM3)

Remote MACs:

OK Cancel

2. allow active FTP on VM

Edit IP Rule

Name: Allow FTP1

Description:

Action: Allow

Protocol: TCP

Type: Any

Local Ports (example 80,8080-8088,443) 20, 21

Remote Ports (example 80,8080-8088,443) 0-65535

Remote IPs (example 192.168.0.1, 192.168.1.0/255.255.255.0, 192.168.2.0-192.168.2.255)

Remote VMs (example Test2003, VM1, VM3)

Remote MACs Any

OK Cancel

3. broadcast rule to allow DHCP

Edit Broadcast Rule

Name:

Description:

Action:

Type:

Remote IPs (example 192.168.0.1, 192.168.1.0/255.255.255.0, 192.168.2.0-192.168.2.255)

Remote VMs (example Test2003, VM1, VM3) ...

Remote MACs:

b.) **Allowing remote access to VM**

Common scenario:

- VM has IIS on it, and possibly MS SQL server;
- RDP should be opened;

Add Rule

Name:

Description:

Action:

Protocol:

Type:

Local Ports (example 80,8080-8088,443)

Remote Ports (example 80,8080-8088,443)

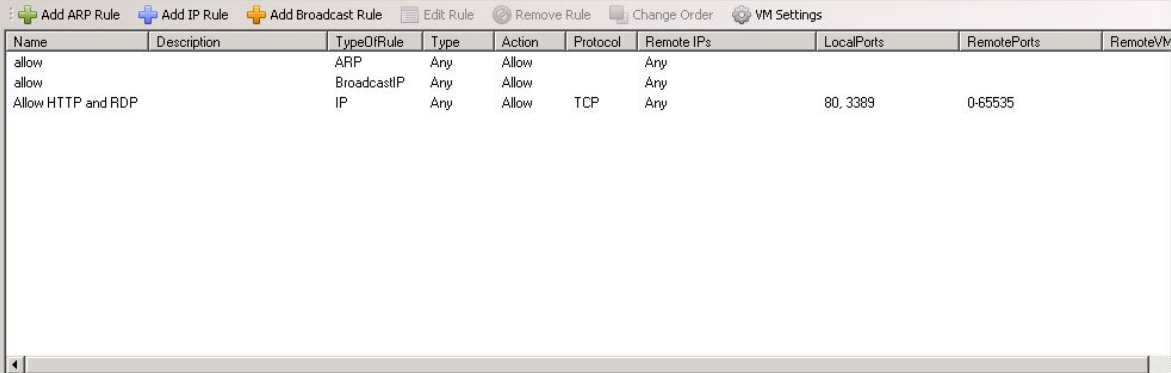
Remote IPs (example 192.168.0.1, 192.168.1.0/255.255.255.0, 192.168.2.0-192.168.2.255)

Remote VMs (example Test2003, VM1, VM3) ...

Remote MACs

- http:// traffic should be allowed:

ARP and IP broadcast rule could be set to allow all. VM -to-VM TCP traffic will be blocked then.



Name	Description	TypeOfRule	Type	Action	Protocol	Remote IPs	LocalPorts	RemotePorts	RemoteVM
allow		ARP	Any	Allow		Any			
allow		BroadcastIP	Any	Allow		Any			
Allow HTTP and RDP		IP	Any	Allow	TCP	Any	80, 3389	0-65535	

***/Notes:**

- *If DHCP is used – you need an additional IP broadcast rule to allow all IP broadcasts.*
- *Windows FW on the management machine and on the Hyper-V host should allow WMI traffic.*