Lab 03 - Just Enough Administration (JEA)

Objective

Use the techniques you learned in the first lab to track down activity in a PowerShell remoting or JEA session. You will add in the WinRM log to see connection events.

Background

This lab is not going to teach JEA directly, but instead how to log and investigate JEA activity. For more information see the official JEA audit documentation.

Overview

First you will configure a JEA endpoint and roles on the `ts1.training.com` server. Then you will use the Windows 10 `client01.training.com` client to remotely administer the server. Finally you will track the activity using Windows logs and PowerShell transcripts.

For remoting sessions you will add the WinRM log to your forensic toolbox, looking for event `193`. Also, JEA allows you to specify a separate transcription directory dedicated to only remoting sessions.

In the online lab web interface you must click the `View VMs` button under the **Virtual Clients** heading. You will use RDP to the **RDP/SSH IP** of the `ts1.training.com` and `client-01.training.com` machines. You will need both RDP sessions open throughout the lab.

For your convenience the PowerShell commands have been saved in script files under `C:\Labs` on the two Windows lab hosts. You can open these in the ISE to run commands without copy/paste from the lab guide.

Exercise 3.1 - Auditing JEA

- Set up the server PowerShell policies
- Configure JEA on the server
- Connect from the client
- Investigate the activity

3.1.1 - Set up the server PowerShell policies

You will begin by ensuring that PowerShell auditing measures are implemented on the Windows Server 2016 box. Then you will delegate JEA access, roles, and capabilities.

1. RDP into the `ts1.training.com` server using the **RDP/SSH IP** from the lab web page. Use the `Training\Administrator` credential from the lab setup guide.

2. For this lab we want to make sure the PowerShell policies are enabled. We will do this with a pre-configured GPO. Open an elevated PowerShell console. Run the following command and reuse the elevated session for the other commands as well:

```
Set-GPLink -Name 'PowerShell Security' -Target 'DC=training,DC=com' -
LinkEnabled Yes
```

3. Now refresh GPO in the same elevated PowerShell console:

   ```
   gpupdate /force /wait:0
   ```

4. Verify that the PowerShell policies are enabled in the registry:

   ```
   Get-ChildItem HKLM:\Software\Policies\Microsoft\Windows\PowerShell\ -
   Recurse
   ```

3.1.2 - Configure JEA on the server

Next you will delegate JEA access, roles, and capabilities. You will perform these steps in the RDP session to `ts1.training.com` by selectively executing portions of a script staged on the server.

1. Navigate to the `C:\Labs\` directory. Open the `JEA_Server.ps1` file in the PowerShell ISE.
2. Review and select all of the lines in the first region of code. Press `F8` to run the selection or use the `Run Selection` icon on the ISE toolbar.
3. One-at-a-time select and run the lines within the WinRM region of code. Study the output of each command.
4. Execute the lines in the next region of code to create the JEA module.
5. You would use the commands in the next code region to learn about PowerShell remoting and to identify the modules and commands to be whitelisted in the JEA session. Skip this section of code for now.
6. Select and run the next region of code to create the JEA endpoint with role capabilities defined. Study the lines in the code to understand the capabilities granted in each role.
7. Optionally run each line individually in the `Test JEA` code region to see which commands each test user will have available.

The JEA configuration is now ready for restricted remoting sessions.

> **NOTE** - To further restrict JEA access consider using the Windows Firewall. You can edit the default `Windows Remote Management (HTTP-In)` rules that restrict access on port 5985 to limit access based on the remote IP addresses coming into the server.

3.1.3 - Connect from the client

1. RDP into the `client-01.training.com` machine using the **RDP/SSH IP** from the lab web page. Use the `Training\Student01` credential from the lab setup guide.
2. Locate the JEA client script in the `C:\Labs\` directory, and open it in the PowerShell ISE.
3. Select lines 7-11 in the script and run them to create credentials for each of the three test users.

4. Scroll down in the script and run the first `Enter-PSSession` command to test the access and capabilities of the first test user. Once in the remoting session, test the various commands listed above in the script by typing into the blue PowerShell console pane below. Notice which commands work and which ones do not. Type `Exit` to leave the JEA remoting session. Repeat this for each of the three `Enter-PSSession` commands to test capabilities of each user.

3.1.4 - Investigate the activity

1. Switch back over to the RDP session for `ts1.training.com1`.
2. Initially viewing the logs will be easiest with Windows Event Viewer. Then switch to PowerShell to search for specific keywords in the event logs. Pay close attention to the `Username` and `RunAs User` in the log headers. The `LabJEA-Server.ps1` script file has helpful command examples to get started. Look for evidence of remote command activity in the following locations:
    - WinRM
        - What can you find in event ID `193` in the log `Microsoft-Windows-WinRM/Operational`?
    - Pipeline Execution Details
        - What can you find in event ID `800` in the log `Windows PowerShell`?
    - Module Logging
        - What can you find in event ID `4103` in the log `Microsoft-Windows-PowerShell/Operational`?
    - Script Block Logging
        - What can you find in event ID `4104` in the log `Microsoft-Windows-PowerShell/Operational`?
    - Transcription
        - What can you find in transcript files in `C:\PSTranscriptsJEA`?
        - What can you find in transcript files in `C:\PSTranscripts`?
        - Browse and open individual transcript files to see all session activity.
3. JEA in this lab uses a virtual account. How does this look different in the logs and transcripts you identified?
4. Explore these log sources on your own to see what you can find of interest to your enterprise forensic investigations.

End of line.

LabJEA-Server.ps1

```
break

#region ==== Set up AD accounts
================================================

$Domain = $env:USERDOMAIN

$pw = ConvertTo-SecureString -String 'P@ssw0rd' -AsPlainText -Force
```

```
$cred_alice   = New-Object -TypeName PSCredential -ArgumentList
"$Domain\alice",$pw
$cred_bob     = New-Object -TypeName PSCredential -ArgumentList
"$Domain\bob",$pw
$cred_charlie = New-Object -TypeName PSCredential -ArgumentList
"$Domain\charlie",$pw

$alice   = New-ADUser -Name Alice   -AccountPassword $pw -Enabled $true -
PassThru
$bob     = New-ADUser -Name Bob     -AccountPassword $pw -Enabled $true -
PassThru
$charlie = New-ADUser -Name Charlie -AccountPassword $pw -Enabled $true -
PassThru

New-ADGroup -Name GGStorage -GroupCategory Security -GroupScope Global
New-ADGroup -Name GGNetwork -GroupCategory Security -GroupScope Global

Add-ADGroupMember -Identity GGStorage -Members $alice,$charlie
Add-ADGroupMember -Identity GGNetwork -Members $bob,$charlie

#endregion
=====================================================================


#region ==== WinRM WSMan Remoting
=============================================

Get-PSSessionConfiguration
# Notice permissions on Microsoft.PowerShell, the default endpoint...
# This is what you connect to with Enter-PSSession or Invoke-Command.

# Remoting configuration
dir WSMan:\localhost
dir WSMan:\localhost\Service
dir WSMan:\localhost\Listener\Listener_1084132640

#endregion
=====================================================================


#region ==== Setup JEA Module
================================================

# Create a folder for the module
$modulePath = Join-Path $env:ProgramFiles
"WindowsPowerShell\Modules\SummitJEA"

# Create an empty script module and module manifest.
New-Item -ItemType File -Path (Join-Path $modulePath
"SummitJEAFunctions.psm1") -Force
New-ModuleManifest -Path (Join-Path $modulePath "SummitJEA.psd1") -
RootModule "SummitJEAFunctions.psm1"

# Create the RoleCapabilities folder and copy in the PSRC file
$rcFolder = Join-Path $modulePath "RoleCapabilities"
```

```
New-Item -ItemType Directory $rcFolder
Set-Location $rcFolder

# Observe the folder structure for JEA
cls; Get-ChildItem $modulePath -Recurse

#endregion
=============================================================================


#region ==== Scope capabilities
================================================

# Constrained Language Mode
Get-Help New-PSRoleCapabilityFile
Get-Help about_Language_Modes

# Identify the modules to import and the command types
Get-Command -Name 'Sort-Object','Format-Table','Format-List' | Format-Table
-AutoSize
Get-Command -Name 'Get-SmbShare','Get-ChildItem' | Format-Table -AutoSize
Get-Command -Name 'Get-Disk','Get-Volume','Get-Partition' | Format-Table -
AutoSize
Get-Command -Name 'Get-NetAdapter','Test-NetConnection' | Format-Table -
AutoSize
Get-Command -Name ping,ipconfig,whoami | Format-Table -AutoSize

#endregion
=============================================================================


#region ==== Set up JEA with role capabilities
================================

$rc_disk = @{
    Description             = 'View Disks and Shares'
    ModulesToImport         = 'Storage','SmbShare' # Already imported by
default: 'Microsoft.PowerShell.Management'
    VisibleAliases          = 'cd', 'dir','ft','fl'
    VisibleCmdlets          = 'Get-*Item','Set-Location','Sort-
Object','Format-Table','Format-List'
    VisibleFunctions        =
'TabExpansion2','prompt','SmbShare\Get*','Storage\Get*'
    VisibleProviders        = 'FileSystem'
    VisibleExternalCommands = 'C:\Windows\System32\whoami.exe'
}
New-PSRoleCapabilityFile -Path .\ViewDisksAndShares.psrc @rc_disk

$rc_network = @{
    Description             = 'View Network'
    ModulesToImport         = 'NetAdapter', 'NetTCPIP'
    VisibleAliases          = 'ft','fl'
    VisibleCmdlets          = 'Sort-Object','Format-Table','Format-List'
    VisibleFunctions        =
'TabExpansion2','NetAdapter\Get*','NetTCPIP\Get*','Test-NetConnection'
```

```
    VisibleExternalCommands =
'C:\Windows\System32\whoami.exe','C:\Windows\System32\ping.exe','C:\Windows
\System32\ipconfig.exe'
}
New-PSRoleCapabilityFile -Path .\ViewNetwork.psrc @rc_network

$pssc = @{
    SessionType         = 'RestrictedRemoteServer'
    LanguageMode        = 'NoLanguage'
    ExecutionPolicy     = 'Restricted'
    RunAsVirtualAccount = $true
    TranscriptDirectory = 'C:\PSTranscriptsJEA\'
    RoleDefinitions     = @{
        "$Domain\GGStorage" = @{ RoleCapabilities = 'ViewDisksAndShares' }
        "$Domain\GGNetwork" = @{ RoleCapabilities = 'ViewNetwork' }
    }
}
New-PSSessionConfigurationFile -Path .\JEAConfig.pssc @pssc

Test-PSSessionConfigurationFile -Path .\JEAConfig.pssc

Register-PSSessionConfiguration -Path .\JEAConfig.pssc -Name SummitJEA

cd C:\Labs

#endregion
=================================================================


#region ==== Test JEA
=============================================================

$Domain = $env:USERDOMAIN

Get-PSSessionCapability -ConfigurationName 'SummitJEA' -Username
"$Domain\alice"
Get-PSSessionCapability -ConfigurationName 'SummitJEA' -Username
"$Domain\bob"
Get-PSSessionCapability -ConfigurationName 'SummitJEA' -Username
"$Domain\charlie"

### RDP TO CLIENT-01 AND RUN THROUGH THE LabJEA-Client.ps1 TEST SCRIPT.

#endregion
=================================================================


#region ==== Fingerprints
=====================================================

# Use these commands as a starting point. Adjust the parameters and
pipelines
# to find specific events of interest.

Get-WinEvent -LogName 'Microsoft-Windows-WinRM/Operational' -MaxEvents 10 -
```

```
FilterXPath '*[System[(EventID=193)]]' | Format-Table -Wrap
Get-WinEvent -LogName 'Microsoft-Windows-WinRM/Operational' -MaxEvents 10 -
FilterXPath '*[System[(EventID=193)]]' | Format-List *

Get-WinEvent -LogName 'Windows PowerShell' -MaxEvents 20 -FilterXPath '*
[System[(EventID=800)]]' | Format-Table -Wrap

Get-WinEvent -LogName 'Microsoft-Windows-PowerShell/Operational' -MaxEvents
20 -FilterXPath '*[System[(EventID=4103 or EventID=4104)]]' | Format-Table
-Wrap

# How can you differentiate between the local test sessions with Get-
PSSessionCapability
# and the actual remoting sessions? Browse the transcript files and open
then directly.

dir C:\PSTranscriptsJEA -Recurse

ise (dir C:\PSTranscriptsJEA | Sort-Object LastWriteTime -Descending)
[0].FullName

dir C:\PSTranscriptsJEA\ -Recurse | Sort-Object LastWriteTime | Where-
Object Length -gt 0 | Select-String "stop-service"

dir C:\PSTranscripts -Recurse

dir C:\PSTranscripts\ -Recurse | Sort-Object LastWriteTime | Where-Object
Length -gt 0 | Select-String "stop-service"

#endregion
======================================================================


#region ==== Reset demo
=====================================================

$ErrorActionPreference = 'SilentlyContinue'
'alice','bob','charlie' | ForEach-Object {Get-ADUser -Identity $_ | Remove-
ADUser -Confirm:$false}
'GGStorage','GGNetwork' | ForEach-Object {Get-ADGroup -Identity $_ |
Remove-ADGroup -Confirm:$false}
Get-PSSessionConfiguration |
    Where-Object {$_.Name -notin 'microsoft.powershell',
                                 'microsoft.powershell.workflow',
                                 'microsoft.powershell32',

'microsoft.windows.servermanagerworkflows'} |
    ForEach-Object {Unregister-PSSessionConfiguration $_.Name}
#Remove-Item -Path WSMan:\localhost\Plugin\SummitJEA -Recurse -Force -
Confirm:$false
cd \
Remove-Item C:\PSTranscriptsJEA\ -Force -Recurse
Remove-Item (Join-Path $env:ProgramFiles
"WindowsPowerShell\Modules\SummitJEA") -Force -Recurse
$ErrorActionPreference = 'Continue'
```

```
#endregion
=============================================================================
```

LabJEA-Client.ps1

```
break

#region ==== Test JEA
=============================================================

### RUN THESE LINES FROM CLIENT-01

$Domain = $env:USERDOMAIN
$pw = ConvertTo-SecureString -String 'P@ssw0rd' -AsPlainText -Force
$cred_alice   = New-Object -TypeName PSCredential -ArgumentList
"$Domain\alice",$pw
$cred_bob     = New-Object -TypeName PSCredential -ArgumentList
"$Domain\bob",$pw
$cred_charlie = New-Object -TypeName PSCredential -ArgumentList
"$Domain\charlie",$pw

# Within each of the following three remoting sessions try the following
commands.
# Test TAB expansion as you go along.
Get-Command
format C:
Stop-Service NTDS -Force
Get-NetAdapter
Get-NetAdapterStatistics
Get-NetTCPConnection
Get-SmbShare
dir C:\
Get-Disk
Get-Volume
Get-Partition
# try some commands of your own
Exit

# Disks
Enter-PSSession -ComputerName ts1 -ConfigurationName SummitJEA -Credential
$cred_alice

# Shares
Enter-PSSession -ComputerName ts1 -ConfigurationName SummitJEA -Credential
$cred_bob

# Disks & Shares
Enter-PSSession -ComputerName ts1 -ConfigurationName SummitJEA -Credential
$cred_charlie
```

```
#endregion
================================================================
```