Defending the Windows Domain: Supplement

Exercises

1. Chapter 8 showed that if the registry key HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer was set to DWORD 1, then an attacker with an unprivileged shell can escalate privileges to SYSTEM. Write a script (PowerShell or batch) that can check every system in the domain to see if this is the case.
2. It is possible to use the registry to block applications from running. On a Windows system, create the key HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer. Inside this key, create the DWORD value "DisallowRun" and set the data to 1. Inside the key HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun, create the value "1" of type String, and set the data to cmd.exe; create the value "2" of type String, and set the value to powershell.exe. Try to launch cmd.exe or powershell.exe; note that both are blocked. Finally, create the file laugh.bat with the content "cmd.exe" and run the batch file. Note that, though cmd.exe cannot be run from Explorer, it can be run from the batch file.
3. Another approach to blacklisting files is through group policy. Create a group policy object and navigate Group Policy → User Configuration → Administrative Templates → System → Don’t run specified Windows Applications. Compare this approach to the approach in the previous problem.
4. Compile the file PSTrace.mof from Listing 7-4 that makes use of a WMI filter and an ActiveScriptEventConsumer to log the use of PowerShell on a system. Configure Software Restriction Policies with a whitelist. Does it block the script from the WMI filter?
5. Create a WMI MOF file that triggers whenever a StartupCommand class instance is created. Block it or log it.
6. Create a WMI MOF file that triggers whenever a service is installed. Consider \_\_InstanceCreationEvent for Win32\_Service. Block it or log it.
7. Create a WMI MOF file that triggers whenever a registry value is created or changed. Consider RegistryKeyChangeEvent or RegistryValueChangeEvent.
8. Create WMI persistence on a Windows 10 system and search the log Applications and Services Logs → Microsoft → Windows → WMI-Activity → Operational for EventID 5861. Do the same on a Windows 8 system and compare.
9. Create a group policy object to automatically create the registry value UseLogonCredentials in the key HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest and set its DWORD value to zero
10. The command winmgmt.exe can be used to backup and restore the WMI repository. Test this.
11. Use Impacket's wmiexec.py to execute a command on a remote target. What appears in the logs?