Detecting Offensive PowerShell Attack Tools

PowerShell exists in the *System.Management.Automation.dll* dynamic linked library file (DLL) and can host different runspaces which are effectively PowerShell instances. Since PowerShell code can be executed without running PowerShell.exe, blocking this executable is not an ideal solution to block attacks.

PowerShell supports various language modes that restrict what PowerShell can do. Constrained language mode limits the capability of PowerShell to base functionality removing advanced feature support such as .Net & Windows API calls and COM access. The lack of this advanced functionality stops most PowerShell attack tools since they rely on these methods.

Interesting Activity:

- Downloads via .Net (New-Object Net.WebClient).DownloadString)
- Invoke-Expression (& derivatives: “iex”).
- BITS activity
- Scheduled Task creation/deletion.
- PowerShell Remoting

The best method to detect PowerShell attack code is to look for key indicators – code snippets required for the code to run correctly.

Many PowerShell attack tools can be detected by monitoring PowerShell Operational log for the following indicators. These are specific to PowerSploit tools, but many other PowerShell attack tools use the same methods.

Invoke-Mimikatz:

- “System.Reflection.AssemblyName”
- “System.Reflection.Emit.AssemblyBuilderAccess “
- “System.Runtime.InteropServices.MarshalAsAttribute”
- “TOKEN_PRIVILEGES”
- “SE_PRIVILEGE_ENABLED“

Invoke-TokenManipulation:

- “TOKEN_IMPERSONATE”
- “TOKEN_DUPLICATE”
- “TOKEN_ADJUST_PRIVILEGES”
- Invoke-CredentialInjection:
- “TOKEN_PRIVILEGES”
- “GetDelegateForFunctionPointer”

Invoke-DLLInjection:

- “System.Reflection.AssemblyName“
- “System.Reflection.Emit.AssemblyBuilderAccess“

Invoke-Shellcode:

- “System.Reflection.AssemblyName“
- “System.Reflection.Emit.AssemblyBuilderAccess”
- “System.MulticastDelegate”
- “System.Reflection.CallingConventions”

Get-GPPPassword:

- “System.Security.Cryptography.AesCryptoServiceProvider”
- “0x4e,0x99,0x06,0xe8,0xfc,0xb6,0x6c,0xc9,0xfa,0xf4”
- “Groups.User.Properties.cpassword”
- “ScheduledTasks.Task.Properties.cpassword”

Out-MiniDump:

- “System.Management.Automation.WindowsErrorReporting”
- “MiniDumpWriteDump”