Using PowerShell for Security Testing (1)

PowerShell allows for modules to be loaded and their functions executed. The first that that we need to do is to identify if we have the ability to execute a PowerShell script. We can do this via the **Get-ExecutionPolicy** command.

```
PS C:\> Get-ExecutionPolicy
Restricted
```

The restricted execution policy is the policy used that prohibits the execution of a PowerShell script. We can set the context of the local user to allow PowerShell scripts to execute as follows:

```
PS C:\> Set-ExecutionPolicy -Scope CurrentUser Unrestricted
PS C:\> Get-ExecutionPolicy
Unrestricted
```

Should you be unable to set then execution policy for power shell then you can also make use of the command flag. This flag when used with invoking the execution of PowerShell allows for PowerShell commands to be executed.

```
PS C:\> PowerShell -Command {Get-ExecutionPolicy}
Restricted
```

We can use the **Import-Module** function to load a module into PowerShell so that its verbs and available. First let us list the modules available to us via the **Get-Module** command.

```
PS C:\> Get-Module - ListAvailable
```

We can then load all available modules into PowerShell via the following.

```
PS C:\> Get-Module – ListAvailable | Import-Module
```

We can load a specific module if we know the directory where it can be located via the use of the **-Name** flag.

```
PS C:\> Import-Module -Name C:\Users\ABlyth\Modules\MyMod
VERBOSE: Loading Module from C:\Users\ABlyth\Modules\MyMod.psm1
```

We can get information about the function/verbs that a module supports via the **Get-Command** command. This will display the version information.

```
PS C:\> Get-Command -Module PSDiagnostics

CommandType     Name                Version    Source
-----------     ----                -------    ------
Function        Disable-PSTrace     6.1.0.0    PSDiagnostics
Function        Enable-PSTrace      6.1.0.0    PSDiagnostics
```

We can also identify where modules are physically located.

```
PS> Get-Module -ListAvailable | Select-Object Path

Path
----
C:\Windows\System32\WindowsPowerShell\v1.0\Modules\ISE\ISE.psm1
```

Because modules may live in multiple locations we can import a module using a fully qualified path as follows:

```
PS> Get-Module -ListAvailable PowerShellGet | Select-Object Path

Path
----
C:\PowerShell\Modules\2.2.1\PowerShellGet.psd1
PS> Import-Module -Name ' C:\PowerShell\Modules\2.2.1\PowerShellGet.psd1'
```