CYBER RANGERS

ARMORING YOUR BUSINESS

POWERSHELL BASIC SCRIPTING
MODULE 7: SCRIPTING PREREQUISITES

Jan Marek | Cyber Rangers

CEH | CHFI | CompTIA Pentest+ | CEI | MVP | MCT | MCC | MCSA | MCSE

Ethical Hacker, Forensic Investigator & Security Engineer

jan@cyber-rangers.com | www.cyber-rangers.com

AGENDA

- Execution Policy
- Code Signing
- PowerShell Security
- Working with variables, arrays, collections and hash tables
- Working with input data
- Conditions and loops (if, switch, while, do/while, do/until)

EXECUTION POLICY

- Controls which script is allowed to run (signed)

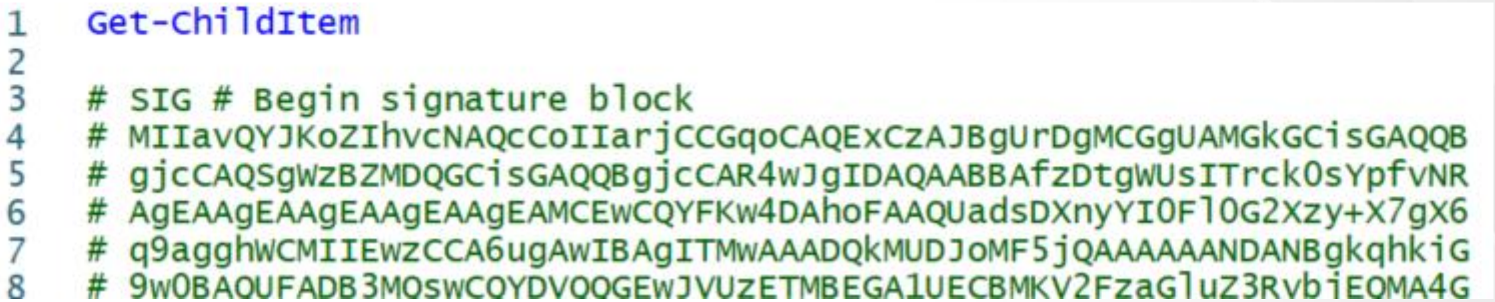
```
Get-ExecutionPolicy
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
powershell.exe -ExecutionPolicy Unrestricted
```

CODE SIGNING

- Sign the code to trust it in your environment and to control the changes

```
Set-AuthenticodeSignature -Certificate $cert -FilePath 'C:\CyberTool.ps1’
```

- Signature added to the script content



```
Get-ChildItem

# SIG # Begin signature block
# MIIavQYJKoZIhvcNAQcCoIIarjCCGqoCAQExCzAJBgUrDgMCGgUAMGkGCisGAQQB
# gjcCAQSgWzBZMDQGCisGAQQBgjcCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
# AgEAAgEAAgEAAgEAAgEAMCEwCQYFKw4DAhoFAAQUadsDXnyYI0Fl0G2Xzy+X7gX6
# q9agghWCMIIEwzCCA6ugAwIBAgITMwAAADQkMUDJoMF5jQAAAAANDANBgkqhkiG
# 9w0BAQUFADB3MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4G
```

HOW TO CREATE SELF-SIGNED CERTIFICATE

```
New-SelfSignedCertificate -CertStoreLocation cert:\CurrentUser\My `
-KeyAlgorithm RSA `
-Subject "CN=CodeSign" `\
-KeyLength 2048 `
-Provider "Microsoft Enhanced RSA and AES Cryptographic Provider" `
-KeyExportPolicy Exportable `
-KeyUsage DigitalSignature `
-Type CodeSigningCert
```

POWERSHELL SECURITY

- PowerShell 3+ features
  - Logging
  - Constrained Language Mode
- Anti-malware Scan Interface

VARIABLES AND ARRAYS

```
[string]$variable1 = 'text'
[int]$variable2 = 123
[bool]$variable3 = 1
$variable4 = Get-Date
$variable5 = Get-Item C:\Windows
$variable6 = @()
$variable7 = New-Object -TypeName "System.Collections.ArrayList"
```

HASHTABLES

- Basic Hashtable
- Ordered
- Enumerating Hashtables
- Splatting

INPUT DATA

- `Read-Host`
- `Read-Host -AsSecureString`

- `Out-GridView -PassThru`

CONDITIONS

```
if ($var -eq 1) {
    Write-Host 'Var is 1'
}
```

```
switch ($var) {
    1 {Write-Host 'Var is 1'}
}
```

WHILE, DO WHILE, DO UNTIL

```
while ($var -eq 1) {
    Write-Host 'Var is 1'
    Start-Sleep -Seconds 5
}

do {
    Write-Host 'Var is 1'
    Start-Sleep -Seconds 5
} while ($var -eq 1)

do {
    Write-Host 'Var is not 1'
    Start-Sleep -Seconds 5
} until ($var -eq 1)
```

FOR AND FOREACH

```
for ($a = 0; $a -lt 10; $a++) {
    Write-Host ('$a is {0}' -f $a)
}

foreach ($Process in (Get-Process)) {
    Write-Host ('ProcessName is {0}' -f $Process.Name)
}

1..9 | ForEach-Object {Write-Host ('Number is {0}' -f $_)}
```

LAB

Q & A

POWERSHELL BASIC SCRIPTING
MODULE 7: SCRIPTING PREREQUISITES

Jan Marek | Cyber Rangers

CEH | CHFI | CompTIA Pentest+ | CEI | MVP | MCT | MCC | MCSA | MCSE

Ethical Hacker, Forensic Investigator & Security Engineer

jan@cyber-rangers.com | www.cyber-rangers.com

CYBER RANGERS

ARMORING YOUR BUSINESS