



Sign in

802-1x / Compliance Public

Notifications

Fork 0

Star 1

<> Code Issues Pull requests Actions Projects Security Insights

Compliance / Ping Castle / Get-PingCastle-HTMLComplianceReport.ps1



51 lines (29 loc) · 1.53 KB

Code Blame

Raw



```
1
2
3
4
5     #WIP code to perform basic AD security posture audit using the Ping Castle tool
6
7
8     $url = "https://github.com/vletoux/pingcastle/releases/latest"
9
10    # Send a GET request to the URL
11    $response = Invoke-WebRequest -Uri $url
12
13    # Extract the URL of the latest release
14    $latestReleaseUrl = $response.BaseResponse.ResponseUri.AbsoluteUri
15
16    # Display the captured URL
17    Write-Output "Latest Release URL: $latestReleaseUrl"
18
19    # Truncate the text after the last /
20    $latestReleaseName = (Split-Path $latestReleaseUrl -Leaf)
21
22    # Display the captured release name
23    Write-Output "Latest Release Name: $latestReleaseName"
24
25
26
```

```
27
28 #Generate report files.
29 $fileExecutable = "C:\Users\<redacted>\Downloads\PingCastle_2.10.1.1\PingCastle.exe"
30 Start-Process -NoNewWindow -FilePath "$fileExecutable" -ArgumentList "--healthcheck --server <redacted>"
31
32 # Load XML from file for analysis
33 $xml = [xml](Get-Content -Path "C:\Users\<redacted>\Downloads\PingCastle_2.10.1.1\<redacted>.xml")
34
35
36 $latestReleaseName
37 $xml.HealthCheckData.EngineVersion
38
39
40 $xml.HealthCheckData.DomainFQDN
41 $xml.HealthCheckData.GlobalScore
42 $xml.HealthCheckData.StaleObjectsScore
43 $xml.HealthCheckData.PrivilegedGroupScore
44 $xml.HealthCheckData.TrustScore
45 $xml.HealthCheckData.AnomalyScore
46
47 $xml.HealthcheckData.RiskRules.HealthcheckRiskRule | where Category -eq "StaleObjects"
48 $xml.HealthcheckData.RiskRules.HealthcheckRiskRule | where Category -eq "PrivilegedAccounts"
49 $xml.HealthcheckData.RiskRules.HealthcheckRiskRule | where Category -eq "Trust"
50 $xml.HealthcheckData.RiskRules.HealthcheckRiskRule | where Category -eq "Anomalies"
```