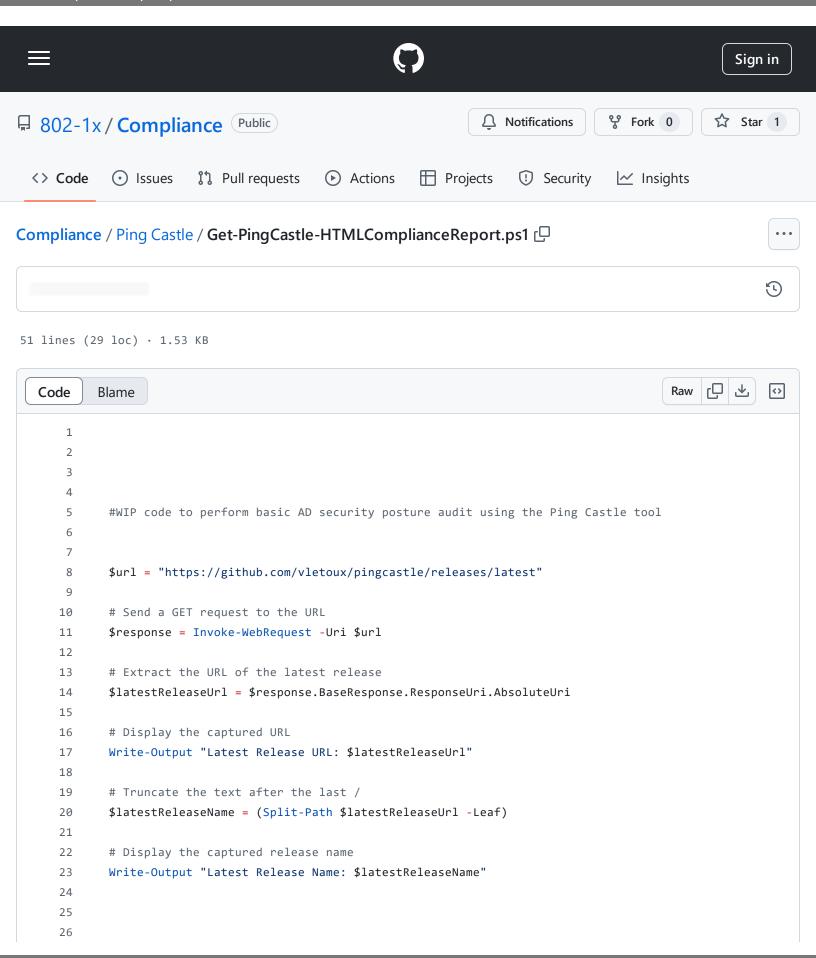
Compliance/Ping Castle/Get-PingCastle-HTMLComplianceReport.ps1 at 2e53df8b6e89686a0b91116b3f42c8f717dca820 · 802-1x/Compliance · GitHub - 31/10/2024 18:53 https://github.com/802-1x/Compliance/blob/2e53df8b6e89686a0b91116b3f42c8f717dca820/Ping%20Castle/Get-PingCastle-

HTMLComplianceReport.ps1#L8



Compliance/Ping Castle/Get-PingCastle-HTMLComplianceReport.ps1 at 2e53df8b6e89686a0b91116b3f42c8f717dca820 · 802-1x/Compliance · GitHub - 31/10/2024 18:53 https://github.com/802-1x/Compliance/blob/2e53df8b6e89686a0b91116b3f42c8f717dca820/Ping%20Castle/Get-PingCastle-

HTMLComplianceReport.ps1#L8

```
27
28
       #Generate report files.
       $fileExecutable = "C:\Users\<redacted>\Downloads\PingCastle_2.10.1.1\PingCastle.exe"
29
       Start-Process -NoNewWindow -FilePath "$fileExecutable" -ArgumentList "--healthcheck --server <redac
30
31
32
       # Load XML from file for analysis
       $xml = [xml](Get-Content -Path "C:\Users\<redacted>\Downloads\PingCastle_2.10.1.1\<redacted>.xml")
33
34
35
       $latestReleaseName
36
       $xml.HealthCheckData.EngineVersion
37
38
39
       $xml.HealthCheckData.DomainFQDN
40
41
       $xml.HealthCheckData.GlobalScore
       $xml.HealthCheckData.StaleObjectsScore
42
43
       $xml.HealthCheckData.PrivilegiedGroupScore
44
       $xml.HealthCheckData.TrustScore
       $xml.HealthCheckData.AnomalyScore
45
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"
```