Start-ADEnum / Functions / Start-ADEnum.ps1

lkys37en Updates e237a73 · 5 years ago History

```
1   Function Start-ADEnum {
2       <#
3       .SYNOPSIS
4       Author: @lkys37en
5       Required Dependencies: Powershell Version 5.x, Windows 10 1803 or 1809
6       Optional Dependencies: None
7
8       .DESCRIPTION
9       This tool is used to automate Active Directory enumeration. The tool requires a dom
10
11      .PARAMETER ClientName
12      Enter clientname for folder structure.
13
14      .PARAMETER Path
15      Enter path where evidence will be placed. If folder doesn't already exist, the scri
16
17      .PARAMETER Domain
18      Enter individual domain to enumerate or let the script automatically identify all v
19
20      .PARAMETER Scan
21      Enter individual scan(s) to perform.
22
23      .EXAMPLE
24      PS C:\> Start-ADEnum  -ClientName lkylabs -Path C:\Projects -Scan All
25      Collects a list of all domain/forest trusts and runs all scans against each domain
26
27      .EXAMPLE
28      PS C:\> Start-ADEnum  -ClientName lkylabs -Path C:\Projects -Domain lkylabs.com  -S
29      Runs all scans against lkylabs.com and corp.lkylabs.com.
30
31      .EXAMPLE
32      PS C:\> Start-ADEnum  -ClientName lkylabs -Path C:\Projects -Domain lkylabs.com,cor
33      Runs PowerView and Bloodhound scans against lkylabs.com and corp.lkylabs.com domain
34
35      #>
36      [CmdletBinding()]
37      Param (
38          [Parameter(Mandatory = $true)]
39          [ValidateNotNullOrEmpty()]
40          [String]
41          $ClientName,
42
43          [Parameter(Mandatory = $true)]
44          [ValidateNotNullOrEmpty()]
45          [String]
46          $Path,
47
48          [Parameter(Mandatory = $false)]
49          [String[]]
50          $Domains,
51
52          [Parameter(Mandatory = $True)]
53          [ValidateSet("ADCS", "Bloodhound", "GPOReport", "PowerView", "PingCastle", "Pri
54          [String[]]
55          $Scan
56      )
```

```
56          )
57
58          Begin {
59              #Folders variable
60              $Folders = @(
61                  "PingCastle"
62                  "PowerView"
63                  "Bloodhound"
64                  "GPO"
65                  "Microsoft Services\Exchange"
66                  "Microsoft Services\ADCS"
67              )
68
69              #Installs all prereqs if missing
70              Write-Host -ForegroundColor Green "[*] Performing prereqs check"
71              Start-PrereqCheck
72
73              Import-Module C:\Tools\PowerSploit\Recon\PowerView.ps1
74
75              #Creating Path and evidence folder structure
```

```
330
331             $Folder = "$Path\$ClientName\$Domain\Bloodhound"
332             Set-Location $Folder ; Invoke-Bloodhound -Domain $Domain -CollectionMethod
333         }
334
335         #Running RSAT GPO Get-GPOReport commands
336         $GPOReportScriptBlock = {
337             $ClientName = $args[0]
338             $Path = $args[1]
339             $Domain = $args[2]
340             $Folder = "$Path\$ClientName\$Domain\GPO\"
341
342             #Importing needed modules
343             Import-Module "C:\Tools\Grouper\grouper.psm1"
344             Import-Module "C:\Tools\PowerSploit\Recon\PowerView.ps1"
345
346             Get-GPOReport -All -ReportType xml -Domain $Domain -Path ($Folder, $Domain
347             Get-GPOReport -All -ReportType Html -Domain $Domain -Path ($Folder, $Domain
348             Invoke-AuditGPOReport -Path ($Folder, $Domain + "_" + "GPOReport.xml" -join
349         }
350
351         #Checking domain for Exchange PrivExchange vulnerability
352         $PrivExchangeCheck = {
353             $ClientName = $args[0]
354             $Path = $args[1]
```

```
355             $Domain = $args[2]
356             $Folder = "$Path\$ClientName\$Domain\Microsoft Services\Exchange\"
357
358             <#Reference:https://eightwone.com/references/schema-versions/
359             https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/#>
360             $ExchangeVersions = @{
361                 "15.02.0397.003" = "Exchange Server 2019 CU2, Not Vulnerable"
362                 "15.02.0330.005" = "Exchange Server 2019 CU1, Not Vulnerable"
363                 "15.02.0221.012" = "Exchange Server 2019 RTM, Vulnerable to PrivExchang
364                 "15.02.0196.000" = "Exchange Server 2019 Preview, Vulnerable to PrivExc
365                 "15.01.1779.002" = "Exchange Server 2016 CU13, Not Vulnerable"
366                 "15.01.1713.005" = "Exchange Server 2016 CU12, Vulnerable to PrivExchan
367                 "15.01.1591.010" = "Exchange Server 2016 CU11, Vulnerable to PrivExchan
368                 "15.01.1531.003" = "Exchange Server 2016 CU10, Vulnerable to PrivExchan
369                 "15.01.1466.003" = "Exchange Server 2016 CU9, Vulnerable to PrivExchang
370                 "15.01.1415.002" = "Exchange Server 2016 CU8, Vulnerable to PrivExchang
371                 "15.01.1261.035" = "Exchange Server 2016 CU7, Vulnerable to PrivExchang
372                 "15.01.1034.026" = "Exchange Server 2016 CU6, Vulnerable to PrivExchang
373                 "15.01.0845.034" = "Exchange Server 2016 CU5, Vulnerable to PrivExchang
374                 "15.01.0669.032" = "Exchange Server 2016 CU4, Vulnerable to PrivExchang
375                 "15.01.0544.027" = "Exchange Server 2016 CU3, Vulnerable to PrivExchang
376                 "15.01.0466.034" = "Exchange Server 2016 CU2, Vulnerable to PrivExchang
377                 "15.01.0396.030" = "Exchange Server 2016 CU1, Vulnerable to PrivExchang
378                 "15.01.0225.042" = "Exchange Server 2016 RTM, Vulnerable to PrivExchang
379                 "15.01.0225.016" = "Exchange Server 2016 Preview, Vulnerable to PrivExc
380                 "15.00.1497.002" = "Exchange Server 2013 CU23, Not Vulnerable"
381                 "15.00.1473.003" = "Exchange Server 2013 CU22, Not Vulnerable!"
382                 "15.00.1395.004" = "Exchange Server 2013 CU21, Vulnerable to PrivExchan
383                 "15.00.1367.003" = "Exchange Server 2013 CU20, Vulnerable to PrivExchan
384                 "15.00.1365.001" = "Exchange Server 2013 CU19, Vulnerable to PrivExchan
385                 "15.00.1347.002" = "Exchange Server 2013 CU18, Vulnerable to PrivExchan
386                 "15.00.1320.004" = "Exchange Server 2013 CU17, Vulnerable to PrivExchan
387                 "15.00.1293.002" = "Exchange Server 2013 CU16, Vulnerable to PrivExchan
388                 "15.00.1263.005" = "Exchange Server 2013 CU15, Vulnerable to PrivExchan
389                 "15.00.1236.003" = "Exchange Server 2013 CU14, Vulnerable to PrivExchan
390                 "15.00.1210.003" = "Exchange Server 2013 CU13, Vulnerable to PrivExchan
391                 "15.00.1178.004" = "Exchange Server 2013 CU12, Vulnerable to PrivExchan
392                 "15.00.1156.006" = "Exchange Server 2013 CU11, Vulnerable to PrivExchan
393                 "15.00.1130.007" = "Exchange Server 2013 CU10, Vulnerable to PrivExchan
394                 "15.00.1104.005" = "Exchange Server 2013 CU9, Vulnerable to PrivExchang
395                 "15.00.1076.009" = "Exchange Server 2013 CU8, Vulnerable to PrivExchang
396                 "15.00.1044.025" = "Exchange Server 2013 CU7, Vulnerable to PrivExchang
397                 "15.00.0995.029" = "Exchange Server 2013 CU6, Vulnerable to PrivExchang
398                 "15.00.0913.022" = "Exchange Server 2013 CU5, Vulnerable to PrivExchang
399                 "15.00.0847.032" = "Exchange Server 2013 SP1, Vulnerable to PrivExchang
400                 "15.00.0775.038" = "Exchange Server 2013 CU3, Vulnerable to PrivExchang
401                 "15.00.0712.024" = "Exchange Server 2013 CU2, Vulnerable to PrivExchang
402                 "15.00.0620.029" = "Exchange Server 2013 CU1, Vulnerable to PrivExchang
403                 "15.00.0516.032" = "Exchange Server 2013 RTM, Vulnerable to PrivExchang
404             }
405             $RootDSE = "DC=$($Domain.Replace('.', ',DC='))"
406             $CN = $Domain.Split('.')[0]
407             $ExchangeVersion = ([ADSI]"LDAP://cn=$CN,cn=Microsoft Exchange,cn=Services,
408             $ExchangeVersions[$ExchangeVersion] | Out-File ($Folder, $Domain + "_" + "E
409         }
410
411         #Command to extract Active Directory Certificate Services usable certificates
412         $ADCS = {
413             $ClientName = $args[0]
414             $Path = $args[1]
415             $Domain = $args[2]
416             $Folder = "$Path\$ClientName\$Domain\Microsoft Services\ADCS\"
417
418             #Identify instances of Active Directory Certificate Service
419             $RootDSE = "DC=$($Domain.Replace('.', ',DC='))"
420             $CAs = ([ADSI]"LDAP://CN=Enrollment Services,CN=Public Key Services,CN=Serv
421
422             #Dumping list of available certificates
423             foreach ($CA in $CAs) {
424                 Write-Host -ForegroundColor Green "[+] Extracting a list of available c
425                 $CA.certificateTemplates | Out-File ($Folder, $Domain + "_" + $CA.displ
426             }
427         }
428
```

Files

e237a73

Go to file

Functions
Install-Tools.ps1
Start-ADEnum.ps1
Start-Prereqcheck.ps1
LICENSE
README.md
Start-ADEnum.psd1
Start-ADEnum.psm1

```
429             #Running PingCastle commands
430             $PingCastleScriptBlock = {
```

Start-ADEnum / Functions / Start-ADEnum.ps1 ↑ Top

Code Blame 520 lines (435 loc) · 26.9 KB Raw

```
435
436                 #PingCastle scanner commands
437                 $Arguments = @(
438                     "--server $Domain --healthcheck --no-enum-limit"
439                     "--scanner laps_bitlocker --server $Domain"
440                     "--scanner nullsession --server $Domain"
441                     "--scanner nullsession-trust --server $Domain"
442                     "--scanner share --server $Domain"
443                     "--scanner smb --server $Domain"
444                     "--scanner spooler --server $Domain"
445                     "--scanner startup --server $Domain"
446                 )
447
448                 #Running all scanner commands
449                 foreach ($Argument in $Arguments) {
450                     Set-Location $Folder ; Start-Process C:\tools\PingCastle\PingCastle.exe
451                 }
452
453                 $Output = (Get-ChildItem $Folder -Exclude *html*, *xml*).FullName
454
455                 #Converting PingCastle text files into CSV's
456                 foreach ($Item in $Output) {
457                     Import-Csv -Path $Item -Delimiter "`t" | Export-Csv -Path  ($Folder + (
458                 }
459
460                 #Remove original text files
461                 (Get-ChildItem $Folder).FullName | Remove-Item -Include *.txt
462             }
463
464             switch ($Scan) {
465                 "All" {
466                     foreach ($Domain in $Domains) {
467                         Write-Host -ForegroundColor Green "[+] Starting All AD Enum for $Do
468                         Start-Job -ScriptBlock $PowerViewScriptBlock -ArgumentList $ClientN
469                         Start-Job -ScriptBlock $PingCastleScriptBlock -ArgumentList $Client
470                         Start-Job -ScriptBlock $BloodhoundScriptBlock -ArgumentList $Client
471                         Start-Job -ScriptBlock $GPOReportScriptBlock -ArgumentList $ClientN
472                         Start-Job -ScriptBlock $PrivExchangeCheck -ArgumentList $ClientName
473                         Start-Job -ScriptBlock $ADCS -ArgumentList $ClientName, $Path, $Dom
474                     }
475                 }
476
477                 "ADCS" {
478                     foreach ($Domain in $Domains) {
479                         Write-Host -ForegroundColor Green "[+] Starting AD Certificate Serv
480                         Start-Job -ScriptBlock $ADCS -ArgumentList $ClientName, $Path, $Dom
481                     }
482                 }
483
484                 "PowerView" {
485                     foreach ($Domain in $Domains) {
486                         Write-Host -ForegroundColor Green "[+] Starting PowerView Enum for
487                         Start-Job -ScriptBlock $PowerViewScriptBlock -ArgumentList $ClientN
488                     }
489                 }
490
491                 "PingCastle" {
492                     foreach ($Domain in $Domains) {
493                         Write-Host -ForegroundColor Green "[+] Starting Ping Castle AD Enum
494                         Start-Job -ScriptBlock $PingCastleScriptBlock -ArgumentList $Client
495                     }
496                 }
497
498                 "Bloodhound" {
499                     foreach ($Domain in $Domains) {
500                         Write-Host -ForegroundColor Green "[+] Starting Bloodhound AD Enum
501                         Start-Job -ScriptBlock $BloodhoundScriptBlock -ArgumentList $Client
502                     }
503                 }
```

```
504
505                 "GPOReport" {
506                     foreach ($Domain in $Domains) {
507                         Write-Host -ForegroundColor Green "[+] Starting GPO Report AD Enum
508                         Start-Job -ScriptBlock $GPOReportScriptBlock -ArgumentList $ClientN
509                     }
510                 }
511
512                 "PrivExchange" {
513                     foreach ($Domain in $Domains) {
514                         Write-Host -ForegroundColor Green "[+] Starting PrivExchange AD Enu
515                         Start-Job -ScriptBlock $PrivExchangeCheck -ArgumentList $ClientName
516                     }
517                 }
518             }
519         }
520     }
```