406 lines (345 loc) · 12.7 KB

Preview Code Blame Raw

external help file	Module Name	online version
DSInternals.PowerShell.dll-Help.xml	DSInternals	https://github.com/MichaelGrafnetter/DSInternals/blob/ ADDBAccount.md

Get-ADDBAccount

SYNOPSIS

Reads one or more accounts from a ntds.dit file, including secret attributes.

SYNTAX

All

```
Get-ADDBAccount [-All] [-BootKey <Byte[]>] -DatabasePath <String> [-LogPath <Strin
```

ByName

```
Get-ADDBAccount [-BootKey <Byte[]>] [-SamAccountName] <String> -DatabasePath <Stri
 [<CommonParameters>]
```

BySID

```
Get-ADDBAccount [-BootKey <Byte[]>] -ObjectSid <SecurityIdentifier> -DatabasePath
 [<CommonParameters>]
```

ByDN

```
Get-ADDBAccount [-BootKey <Byte[]>] -DistinguishedName <String> -DatabasePath <Str
 [<CommonParameters>]
```

ByGuid

```
Get-ADDBAccount [-BootKey <Byte[]>] -ObjectGuid <Guid> -DatabasePath <String> [-Lo
 [<CommonParameters>]
```

DESCRIPTION

Reads one or more accounts from an Active Directory database file. When provided with a boot key (AKA SysKey or system key), it also decrypts secret attributes.

EXAMPLES

Example 1

```
PS C:\> Get-ADDBAccount -SamAccountName Administrator `
                        -DatabasePath 'C:\IFM Backup\Active Directory\ntds.dit'
<# Sample Output:
DistinguishedName: CN=Administrator,CN=Users,DC=contoso,DC=com
Sid: S-1-5-21-1236425271-2880748467-2592687428-500
Guid: b3d02974-6b1c-484c-9103-fd2f60d592c4
```

```
SamAccountName: Administrator
SamAccountType: User
UserPrincipalName:
PrimaryGroupId: 513
SidHistory:
Enabled: True
UserAccountControl: NormalAccount, PasswordNeverExpires
SupportedEncryptionTypes: Default
AdminCount: True
Deleted: False
LastLogonDate: 2/23/2015 10:27:18 AM
DisplayName:
GivenName:
Surname:
Description: Built-in account for administering the computer/domain
ServicePrincipalName:
SecurityDescriptor: DiscretionaryAclPresent, SystemAclPresent, DiscretionaryAclAut
Owner: S-1-5-21-1236425271-2880748467-2592687428-512
Secrets
  NTHash:
  LMHash:
  NTHashHistory:
  LMHashHistory:
  SupplementalCredentials:
Key Credentials:
Credential Roaming
  Created:
  Modified:
  Credentials:
#>
```

Retrieves information about a single account from an Active Directory database. Secret attributes are not decrypted as no boot key is provided.

Example 2

```
PS C:\> $key = Get-BootKey -SystemHiveFilePath 'C:\IFM Backup\registry\SYSTEM'
PS C:\> Get-ADDBAccount -DistinguishedName: 'CN=Joe Smith,OU=Employees,DC=contoso,
                        -BootKey $key `
                        -DatabasePath 'C:\IFM Backup\Active Directory\ntds.dit'
<# Sample Output:
DistinguishedName: CN=Joe Smith,OU=Employees,DC=contoso,DC=com
Sid: S-1-5-21-1236425271-2880748467-2592687428-1110
Guid: 6fb7aca4-fe85-4dc5-9acd-b5b2529fe2bc
```

```
SamAccountName: joe
SamAccountType: User
UserPrincipalName: joe@contoso.com
PrimaryGroupId: 513
SidHistory:
Enabled: True
UserAccountControl: NormalAccount, PasswordNeverExpires
SupportedEncryptionTypes: Default
AdminCount: False
Deleted: False
LastLogonDate:  2/23/2015 10:27:18 AM
DisplayName: Joe Smith
GivenName: Joe
Surname: Smith
Description:
ServicePrincipalName:
SecurityDescriptor: DiscretionaryAclPresent, SystemAclPresent, DiscretionaryAclAut
Owner: S-1-5-21-1236425271-2880748467-2592687428-512
Secrets
  NTHash: 92937945b518814341de3f726500d4ff
  LMHash:
  NTHashHistory:
    Hash 01: 92937945b518814341de3f726500d4ff
  LMHashHistory:
    Hash 01: 30ce97eef1084cf1656cc4be70d68600
  SupplementalCredentials:
    ClearText:
    NTLMStrongHash: 2c6d57beebeafdae65b3f40f2a0d5430
    Kerberos:
      Credentials:
        DES_CBC_MD5
          Key: 7f16bc4ada0b8a52
      OldCredentials:
      Salt: CONTOSO.COMjoe
      Flags: 0
    KerberosNew:
      Credentials:
        AES256_CTS_HMAC_SHA1_96
          Key: cd541be0838c787b5c6a34d7b19274aee613545a0e6cc6f5ac5918d8a464d24f
          Iterations: 4096
        AES128_CTS_HMAC_SHA1_96
          Key: 5c88972747bd454704c117ae52c474e4
          Iterations: 4096
        DES_CBC_MD5
          Key: 7f16bc4ada0b8a52
          Iterations: 4096
      OldCredentials:
      OlderCredentials:
      ServiceCredentials:
      Salt: CONTOSO.COMjoe
      DefaultIterationCount: 4096
      Flags: 0
    WDigest:
      Hash 01: 61fed940f0e8d03a49d3727f55800497
      Hash 02: a1d54499dda6a6b5431f29a8d741a640
      Hash 03: b6cdf00bc0c4578992f718de81251721
      Hash 04: 61fed940f0e8d03a49d3727f55800497
      Hash 05: a1d54499dda6a6b5431f29a8d741a640
      Hash 06: 9a8991bd99763df2e37f1e1e67d71cc8
      Hash 07: 61fed940f0e8d03a49d3727f55800497
      Hash 08: 8a9fe94883c8ccf3bcfc6591ddd2288f
      Hash 09: 8a9fe94883c8ccf3bcfc6591ddd2288f
      Hash 10: 1b7b16b49ecd8d9d59c1d0db6fa2cc36
      Hash 11: d4c24695cfa4dc3810a469d5efb8ecaf
      Hash 12: 8a9fe94883c8ccf3bcfc6591ddd2288f
      Hash 13: a5b8aa5088280298c8c27fa99dcaa1e3
      Hash 14: d4c24695cfa4dc3810a469d5efb8ecaf
      Hash 15: 1aa8e567622fe53d6fb36f1f34f12aaa
      Hash 16: 1aa8e567622fe53d6fb36f1f34f12aaa
      Hash 17: 2af425244079f8f45927c34fa115e45b
      Hash 18: cf283a35102b820e25003b1ddf270221
      Hash 19: b98c902c57449253e6f06b5d585866bd
      Hash 20: 2a690b1eeda9cb8f3157a4a3ba0be9c3
      Hash 21: af2654776d5f9f27f3283ecb0aa25011
      Hash 22: af2654776d5f9f27f3283ecb0aa25011
      Hash 23: ba6fe0513ed2a60ec253a41bbde6a837
      Hash 24: 8bf5a67b598087be948e040f85c72b4d
      Hash 25: 8bf5a67b598087be948e040f85c72b4d
      Hash 26: aa5ff46d23a5c7ebd603e1793225350d
      Hash 27: 656b6a7f5b52d05b3ce9168a2b7ac8ac
      Hash 28: ae884c92ecd87e8d54f1844f09c5a519
      Hash 29: a500a9e26afc9f817df8a07e15771577
Key Credentials:
  Usage=NGC, Source=ActiveDirectory, Device=1966d4da-14da-4581-a7a7-5e8e07e93ad9,
  Usage=NGC, Source=ActiveDirectory, Device=cfe9a872-13ff-4751-a777-aec88c30a762,
Credential Roaming
  Created: 3/12/2017 9:15:56 AM
  Modified: 3/13/2017 10:01:18 AM
  Credentials:
    DPAPIMasterKey: joe\Protect\S-1-5-21-1236425271-2880748467-2592687428-1110\470
    DPAPIMasterKey: joe\Protect\S-1-5-21-1236425271-2880748467-2592687428-1110\7fc
    CryptoApiCertificate: joe\SystemCertificates\My\Certificates\574E468713399854
```

```
    CNGCertificate: joe\SystemCertificates\My\Certificates\3B83BFA7037F6A79B3F3D17
    RSAPrivateKey: joe\Crypto\RSA\S-1-5-21-1236425271-2880748467-2592687428-1110\7
    CNGPrivateKey: joe\Crypto\Keys\E8F13C2BA0209401C4DFE839CD57375E26BBE38F
#>
```

Retrieves information about a single account from an Active Directory database. Secret attributes are decrypted using the provided boot key.

Example 3

```
PS C:\> $results = Get-ADDBAccount -DatabasePath '.\Active Directory\ntds.dit' `
                                  -BootKey acdba64a3929261b04e5270c3ef973cf `
                                  -All  |
                   Test-PasswordQuality -WeakPasswordHashesSortedFile pwned-passwo
```

Performs an offline credential hygiene audit of AD database against HIBP.

Example 4

```
PS C:\> Get-ADDBAccount -All -DatabasePath ntds.dit -BootKey $key |
            Format-Custom -View PwDump |
            Out-File -FilePath users.pwdump -Encoding ascii
```

Exports NT and LM password hashes from an Active Directory database to a pwdump file.

Example 5

```
PS C:\> Get-ADDBBackupKey -DatabasePath '.\ADBackup\Active Directory\ntds.dit' `
                          -BootKey 0be7a2afe1713642182e9b96f73a75da |
            Save-DPAPIBlob -DirectoryPath '.\Output'
PS C:\> Get-ADDBAccount -All -DatabasePath '.\ADBackup\Active Directory\ntds.dit'
            Save-DPAPIBlob -DirectoryPath '.\Output'
```

Extracts DPAPI backup keys and roamed credentials (certificates, private keys, and DPAPI master keys) from an Active Directory database file and saves them to the Output directory. Also creates a file called kiwiscript.txt that contains mimikatz commands needed to decrypt the private keys.

Example 6

```
PS C:\> Get-ADDBAccount -All -DatabasePath '.\ADBackup\Active Directory\ntds.dit'
            Select-Object -ExpandProperty KeyCredentials |
            Where-Object Usage -eq NGC |
            Format-Table -View ROCA
<# Sample Output:

Usage IsWeak Source  DeviceId                             Created    Owner
----- ------ ------  --------                             -------    -----
NGC   True   AzureAD fd591087-245c-4ff5-a5ea-c14de5e2b32d 2017-07-19 CN=John Doe,C
NGC   False  AD      1966d4da-14da-4581-a7a7-5e8e07e93ad9 2019-08-01 CN=Jane Doe,C
#>
```

Lists weak public keys registered in Active Directory that were generated on ROCA-vulnerable TPMs.

Example 7

```
PS C:\> $dc = Get-ADDBDomainController -DatabasePath '.\ADBackup\Active Directory\
PS C:\> $adminSid = '{0}-500' -f $dc.DomainSid
PS C:\> $account = Get-ADDBAccount -Sid $adminSid `
                                   -DatabasePath '.\ADBackup\Active Directory\ntds
                                   -BootKey 0be7a2afe1713642182e9b96f73a75da
```

Retrieves information about a the the built-in Administrator account, even if it was renamed.

PARAMETERS

-All

Indicates that all accounts will be read from the selected database.

```
Type: SwitchParameter
Parameter Sets: All
Aliases: AllAccounts, ReturnAllAccounts

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-BootKey

Specifies the boot key (AKA system key) that will be used to decrypt values of secret attributes.

```
Type: Byte[]
Parameter Sets: (All)
Aliases: key, SysKey, SystemKey

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-DatabasePath

Specifies the path to a domain database, for instance, C:\Windows\NTDS\ntds.dit.

```
Type: String
Parameter Sets: (All)
Aliases: Database, DBPath, DatabaseFilePath, DBFilePath

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-DistinguishedName

Specifies the identifier of an account that will be retrieved from the database.

```
Type: String
Parameter Sets: ByDN
Aliases: dn

Required: True
Position: Named
Default value: None
```

```
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False
```

-LogPath

Specifies the path to a directory where the transaction log files are located. For instance, C:\Windows\NTDS. The default log directory is the one that contains the database file itself.

```
Type: String
Parameter Sets: (All)
Aliases: Log, TransactionLogPath

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ObjectGuid

Specifies the identifier of an account that will be retrieved from the database.

```
Type: Guid
Parameter Sets: ByGuid
Aliases: Guid

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False
```

-ObjectSid

Specifies the identifier of an account that will be retrieved from the database.

```
Type: SecurityIdentifier
Parameter Sets: BySID
Aliases: Sid
```

```
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False
```

-SamAccountName

Specifies the identifier of an account that will be retrieved from the database.

```
Type: String
Parameter Sets: ByName
Aliases: Login, sam

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

System.String

System.Security.Principal.SecurityIdentifier

System.Guid

OUTPUTS

DSInternals.Common.Data.DSAccount

NOTES

RELATED LINKS

Get-BootKey Get-ADReplAccount Get-ADSIAccount Test-PasswordQuality Save-DPAPIBlob Get-ADKeyCredential