2025-11-17

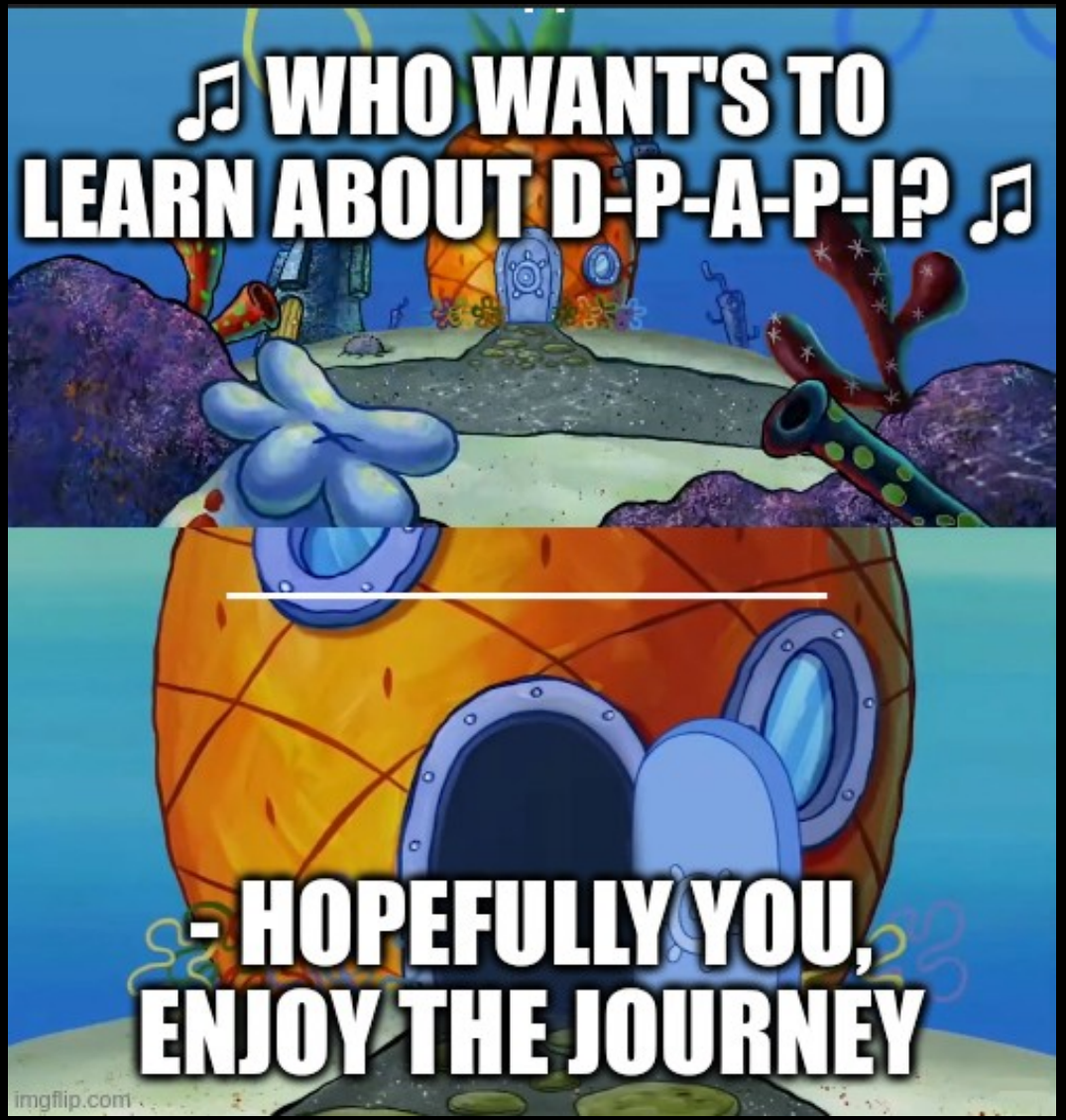
DPAPI DEMYSTIFIED

Abusing the Windows Data Protection API One Secret at a Time

Daniel Küppers

Red Teamer / Penetration Tester

I like credentials.


♫ WHO WANT'S TO LEARN ABOUT D-P-A-P-I? ♫
- HOPEFULLY YOU, ENJOY THE JOURNEY
imgflip.com

WHAT IS DPAPI?


"DPAPI IS NOT THAT WIDELY USED"
TASK SCHEDULER
LOGON CREDENTIALS
AADCONNECT SECRETS
THIRD-PARTY TOOLS
.NET SECRETS
WEB CREDENTIALS
imgflip.com

Data Protection Application Programming Interface

- has only **two** methods (C, C# / PowerShell)
  - `CryptProtectData / Protect`
  - `CryptUnprotectData / Unprotect`
- encryption at rest
- automatic key management

What Do We need for DPAPI - SYSTEM

Master Keys

```
C:\Windows\System32\Microsoft\Protect\{SID}\{GUID}
C:\Windows\System32\Microsoft\Protect\{SID}\User\{GUID}
```

SYSTEM Credential Files

```
C:\Windows\System32\config\systemprofile\AppData
  \{Local|Roaming}\Microsoft\Credentials\{GUID}
```

Master Key Directory - SYSTEM

```
C:\Windows\System32\Microsoft\Protect>dir /a /s

 Directory of C:\Windows\System32\Microsoft\Protect

11/03/2025  09:11 PM    <DIR>          .
11/03/2025  09:11 PM    <DIR>          ..
11/03/2025  09:16 PM    <DIR>          S-1-5-18

 Directory of C:\Windows\System32\Microsoft\Protect\S-1-5-18

11/03/2025  09:16 PM    <DIR>          .
11/03/2025  09:11 PM    <DIR>          ..
11/03/2025  09:16 PM               468 ba713425-587f-4ef9-a1de-fe547a78ed2d
11/03/2025  09:16 PM               468 bb885f4a-2fef-4896-83bc-436a56798f2c
11/03/2025  09:16 PM                24 Preferred
11/03/2025  09:16 PM    <DIR>          User

 Directory of C:\Windows\System32\Microsoft\Protect\S-1-5-18\User

11/03/2025  09:16 PM    <DIR>          .
11/03/2025  09:16 PM    <DIR>          ..
11/03/2025  09:16 PM               468 992e1dee-0fda-4e0d-9a81-8a24e2da3f70
11/03/2025  09:16 PM               468 c77a7b90-37f6-4ee3-81f8-7731a68c669e
11/03/2025  09:11 PM                 0 Diagnostic
11/03/2025  09:16 PM                24 Preferred
     8
```

What Do We need for DPAPI - User

Master Keys

```
C:\Users\{USER}\AppData\Roaming\Microsoft\Protect\{SID}\{GUID}
```

User Credential Files

```
C:\Users\{USER}\AppData\Local\Microsoft\Credentials\{GUID}
C:\Users\{USER}\AppData\Roaming\Microsoft\Credentials\{GUID}
```

Master Key Directory - User

```
C:\Users\user1\AppData\Roaming\Microsoft\Protect>dir /a /s
 Volume in drive C has no label.
 Volume Serial Number is A8E8-E00E

 Directory of C:\Users\user1\AppData\Roaming\Microsoft\Protect

11/06/2025  06:11 PM    <DIR>          .
11/06/2025  07:55 PM    <DIR>          ..
11/06/2025  06:11 PM                24 CREDHIST
11/06/2025  06:11 PM    <DIR>          S-1-5-21-1836656568-3465737956-2455510442-1110
11/06/2025  06:11 PM                76 SYNCHIST
               2 File(s)            100 bytes

 Directory of C:\Users\user1\AppData\Roaming\Microsoft\Protect\S-1-5-21-1836656568-3465737956-2455510442-1110

11/06/2025  06:11 PM    <DIR>          .
11/06/2025  06:11 PM    <DIR>          ..
11/06/2025  06:11 PM               876 33cf4a3a-9a04-4e0a-becd-4d267cf54e29
11/06/2025  06:11 PM               900 BK-ADLAB
11/06/2025  06:11 PM                24 Preferred
               3 File(s)          1,800 bytes
```

User Credentials

```
C:\Users\user1\AppData\Roaming\Microsoft\Credentials>dir /a
 Volume in drive C has no label.
 Volume Serial Number is A8E8-E00E

 Directory of C:\Users\user1\AppData\Roaming\Microsoft\Credentials

11/06/2025  07:44 PM    <DIR>          .
11/06/2025  07:55 PM    <DIR>          ..
11/06/2025  07:44 PM               486 D37BC528382E14FCAC7FB6DBF95656AF
               1 File(s)            486 bytes
```

Things To Do With a Master Key and Credential File

- password known
  - retrieve the secret from Credential file

- password unknown
  - use Master Key file for offline brute force


CRACKING IS LAME, AS MOST USERS WILL HAVE STRONG PASSWORDS
HOW DO WE TELL HIM
SUMMER2025!
USERS
imgflip.com

Credentials Hexdump

```
00000000: 0100 0000 da01 0000 0000 0000 0100 0000  ................
00000010: d08c 9ddf 0115 d111 8c7a 00c0 4fc2 97eb  .........z..O...
00000020: 0100 0000 3a4a cf33 049a 0a4e becd 4d26  ....:J.3...N..M&
00000030: 7cf5 4e29 0000 0020 5600 0000 4100 6e00  |.N)... V...A.n.
00000040: 6d00 6500 6c00 6400 6500 6900 6e00 6600  m.e.l.d.e.i.n.f.
00000050: 6f00 7200 6d00 6100 7400 6900 6f00 6e00  o.r.m.a.t.i.o.n.
00000060: 7300 6400 6100 7400 6500 6e00 2000 6600  s.d.a.t.e.n. .f.
00000070: fc00 7200 2000 5500 6e00 7400 6500 7200  ..r. .U.n.t.e.r.
00000080: 6e00 6500 6800 6d00 6500 6e00 0d00 0a00  n.e.h.m.e.n.....
00000090: 0000 1066 0000 0001 0000 2000 0000 5bc1  ...f...... ...[.
000000a0: adb6 b70e bca5 0405 e402 6199 fd06 e9c5  ..........a.....
000000b0: 5561 14f7 52e2 77da 7796 a49c ba26 0000  Ua..R.w.w....&..
000000c0: 0000 0e80 0000 0002 0000 2000 0000 eb1f  .......... .....
000000d0: 6a06 ae23 9c8f 96dc 9bc0 175c 57b1 0929  j..#.......\W..)
000000e0: 07ad 98f9 528a 57a5 5873 bbed ed81 b000  ....R.W.Xs......
000000f0: 0000 8c1e dab8 94a1 34fe 97bd c236 8bdd  ........4....6..
00000100: c3d1 959d 0e7a a029 b3c3 9792 6d5f 0f00  .....z.)....m_..
00000110: 6974 af9e 0bbd 1544 25e3 41ce 5269 4705  it.....D%.A.RiG.
00000120: b83f daac 85c9 044f 5dd8 a55e 0eea 8548  .?.....O]..^...H
00000130: be44 f569 642e b3f1 3b68 1fff 53be ac6e  .D.id...;h..S..n
00000140: 46d3 e280 4a5a a4da c9ab ab66 a8ff e871  F...JZ.....f...q
00000150: 694e 2d4c 10c9 ac4c 6672 adca 189a 6c5b  iN-L...Lfr....l[
00000160: a8cb 8921 c368 db1b 1188 6273 e66f 6623  ...!.h....bs.of#
[...]
```

Version

Size

Master Key GUID

Little-Endian

Impacket dpapi.py

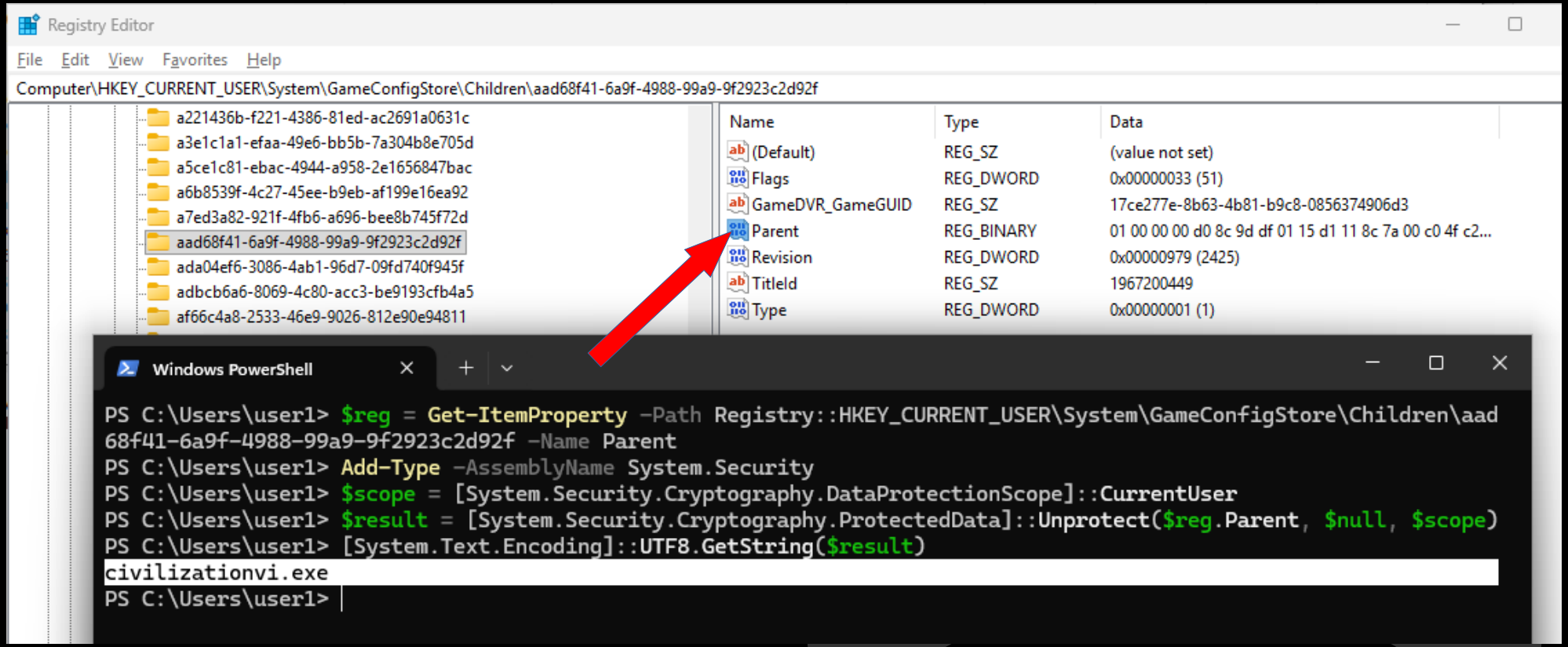
```
$ dpapi.py credential -file D37BC528382E14FCAC7FB6DBF95656AF
Impacket v0.13.0 - Copyright Fortra, LLC and its affiliated companies

[BLOB]
Version          :         1 (1)
Guid Credential  : DF9D8CD0-1501-11D1-8C7A-00C04FC297EB
MasterKeyVersion :         1 (1)
Guid MasterKey   : 33CF4A3A-9A04-4E0A-BECD-4D267CF54E29
Flags            : 20000000 (CRYPTPROTECT_SYSTEM)
Description      : Anmeldeinformationsdaten für Unternehmen

CryptAlgo        : 00006610 (26128) (CALG_AES_256)
Salt             : b'5bc1adb6b70ebca50405e4026199fd06e9c5556114f752e277da7796a49cba26'
HMacKey          : b''
HashAlgo         : 0000800e (32782) (CALG_SHA_512)
HMac             : b'eb1f6a06ae239c8f96dc9bc0175c57b1092907ad98f9528a57a55873bbeded81'
Data             :
b'8c1edab894a134fe97bdc2368bddc3d1959d0e7aa029b3c397926d5f0f006974af9e0bbd154425e341ce52694705b83fdaac85c
9044f5dd8a55e0eea8548be44f569642eb3f13b681fff53beac6e46d3e2804a5aa4dac9abab66a8ffe871694e2d4c10c9ac4c6672
adca189a6c5ba8cb8921c368db1b11886273e66f66239ff5e4bdc93c223dfafdf8aa6391b76c716f31fd96e7f6701d4ff46834757
a87e49f6d7ab3e459878446a229ef0992651044'
Sign             :
b'ed31e828543c917f1a0cbe58cd826bd252f3e5847d1d41d42b8d80f1f46f7cdb7176b91e4c920905c388d8dc89948270d5d8cb1
a909e9982aa4bb515e9218f54'
```


Source: https://github.com/fortra/impacket

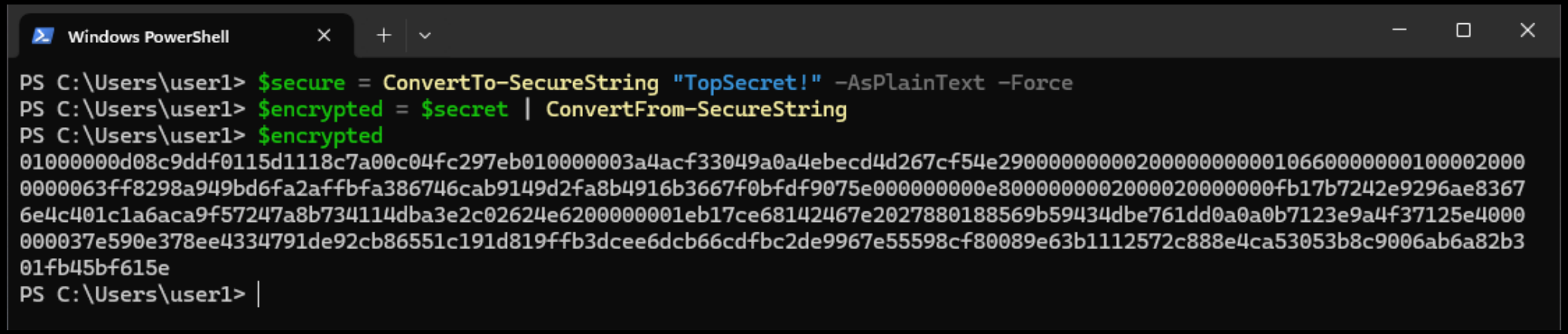
How Credentials Could Look Like

- binary in Registry value
- plain SecureString in a file
- PSCredential serialized via CLIXML
- programmatic blob
- sometimes multiple layers of Base64

Binary in Registry Value



Plain SecureString in a File



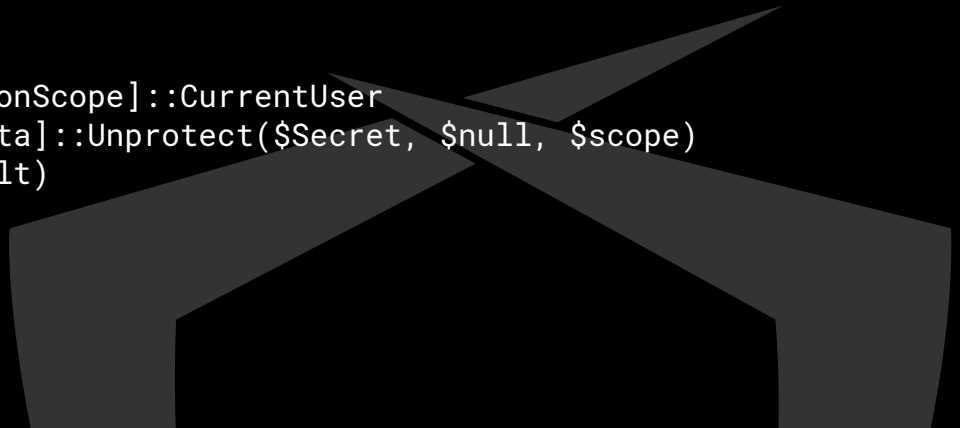
PSCredential serialized via CLIXML

Windows PowerShell

```
PS C:\Users\user1> $psobj = Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
PS C:\Users\user1> $psobj | Export-Clixml -Path ./secure.xml
PS C:\Users\user1> Get-Content .\secure.xml
<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>System.Management.Automation.PSCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>System.Management.Automation.PSCredential</ToString>
    <Props>
      <S N="UserName">superuser</S>
      <SS N="Password">01000000d08c9ddf0115d1118c7a00c04fc297eb010000003a4acf33049a0a4ebecd4d267cf54e290000000020000000
0001066000000010000200000009273b4c29bdd5c98ccedfbcabc45f6ce2437b56315cbaa67d7ffb3707b4ad6b9000000000e8000000002000020000
000fe93f10f16b0dd08bd664ebec657c01ade1eaf8fe66dc3e73231518b5385830020000000f7e7a3821f45e9b0f0a27ac3cfe15465bd420d8a385ea
ce7f71972afc3b950bd40000000644df9b8c80b649a8fc9133445e09e2e26f988413934a78f01442b8c9a63edf79fa0512b0f50556565353dc50223c
4db72f4b5299c40009d16dff20dc7907337</SS>
    </Props>
  </Obj>
</Objs>
```

Example Functions for Protect and Unprotect

```
function Get-Protect {
    param (
        [String]$Secret
    )

    Add-Type -AssemblyName System.Security
    $secretBytes = [System.Text.Encoding]::UTF8.GetBytes($Secret)
    $scope = [System.Security.Cryptography.DataProtectionScope]::CurrentUser
    $result = [System.Security.Cryptography.ProtectedData]::Protect($secretBytes, $null, $scope)
    return $result
}

function Get-Unprotect {
    param (
        [System.Byte[]]$Secret
    )

    Add-Type -AssemblyName System.Security
    $scope = [System.Security.Cryptography.DataProtectionScope]::CurrentUser
    $result = [System.Security.Cryptography.ProtectedData]::Unprotect($Secret, $null, $scope)
    return [System.Text.Encoding]::UTF8.GetString($result)
}
```



Example Encryption and Decryption

```
PS C:\> $secret = Get-Protect -Secret "SecretPassword!"
PS C:\> $secret | Format-Hex

           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   01 00 00 00 D0 8C 9D DF 01 15 D1 11 8C 7A 00 C0  ....Ðß..Ñ.z.À
00000010   4F C2 97 EB 01 00 00 00 3A 4A CF 33 04 9A 0A 4E  OÂë....:JÏ3..N
00000020   BE CD 4D 26 7C F5 4E 29 00 00 00 00 02 00 00 00  ¾ÍM&|õN).......
00000030   00 00 10 66 00 00 00 01 00 00 20 00 00 00 72 0E  ...f...... ...r.
00000040   58 58 82 7F EA B2 C8 EE 68 75 93 82 B8 D5 4D 02  XXê²Èîhu¸ÕM.
00000050   F4 97 D2 ED 45 F3 7C 90 F0 33 ED 71 B8 06 00 00  ôÒíEó|ð3íq¸...
00000060   00 00 0E 80 00 00 00 02 00 00 20 00 00 00 24 B0  ......... ...$°
00000070   C7 AD DE AB 45 4A CE 9E C9 61 AC 28 EB E6 46 11  Ç Þ«EJÎÉa¬(ëæF.
00000080   26 4E CD 39 49 D4 9B 4F C4 90 74 6D 7E D0 10 00  &NÍ9IÔOÄtm~Ð..
00000090   00 00 1D 74 72 57 04 60 3C 68 BF C8 35 FF A9 07  ...trW.`<h¿È5.©.
000000A0   89 A0 40 00 00 00 5D 7B 17 66 3C 70 CF D5 16 F4   @...]{.f<pÏÕ.ô
000000B0   0F 47 2B EC 52 F6 0B AF 6C 54 64 15 46 BB 7F 73  .G+ìRö.¯lTd.F»s
000000C0   06 B6 F9 53 EB 92 DD 74 4D BC D7 1A 30 B3 58 19  .¶ùSëÝtM¼×.0³X.
000000D0   DB 76 C1 FA A6 91 35 8B 99 19 D3 D3 CC B9 0E 4D  ÛvÁú¦5.ÓÓÌ¹.M
000000E0   09 BC 7C 5B 97 94                              .¼|[

PS C:\> Get-Unprotect -Secret $secret
SecretPassword!
```

Master Key GUID

WHAT IS DPAPI ENTROPY?

Entropy in DPAPI

- `Protect($secret, $null, $scope)`
- application-specific input
- concept to prevent secret decryption
- by other apps in current user context
- must be known for `Unprotect`

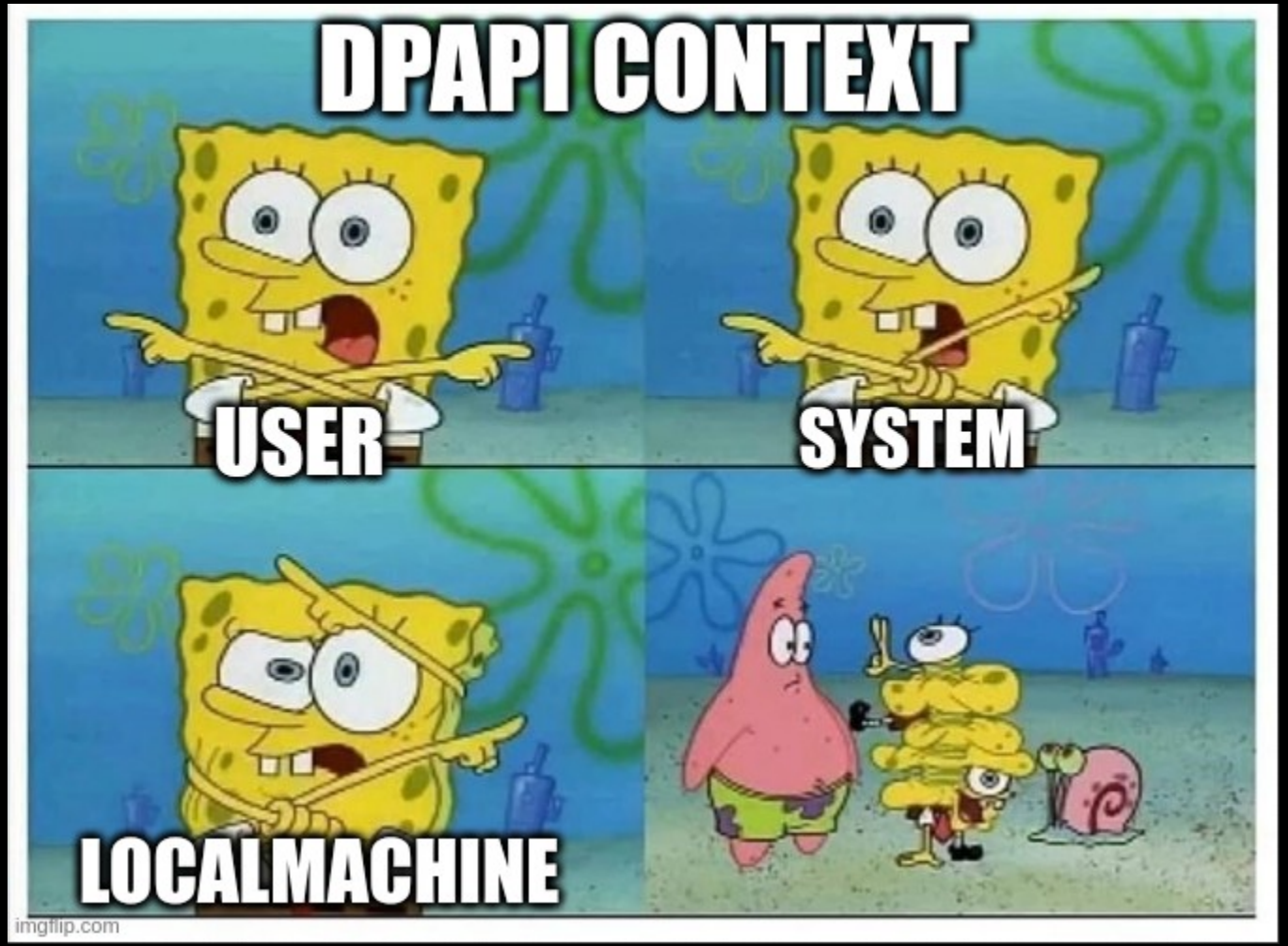
WHAT IS DATAPROTECTIONSCOPE?

DataProtectionScope

- C# knows only two fields for that enum
  - `CurrentUser = 0`
- encrypts with user's Master Key
  - `LocalMachine = 1`
- encrypts with machine-wide Master Key

> *The LocalMachine enumeration value allows multiple accounts to unprotect data. Use this value only when you trust every account on a computer. For most situations, you should use the CurrentUser value.*

Source: https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography.dataprotectionscope


DPAPI CONTEXT
USER
SYSTEM
LOCALMACHINE
imgflip.com

Gotcha - LocalMachine

- `Protect` with LocalMachine
  - → all users of the system can `Unprotect`

Gotcha - Unprotect

- `Protect` with LocalMachine
- `Unprotect` with CurrentUser
  - → all users of the system can `Unprotect`

Source: https://blog.amberwolf.com/blog/2024/september/skeleton-cookie-breaking-into-safeguard-with-cve-2024-45488/

Questions?

BSIDES MUNICH 9

CODE WHITE

FINEST HACKING