Microsoft Security

Trusted by Design: How Windows Uses TPM to Secure PRTs

Dr Nestori Syynimaa

Who am I?

- Dr Nestori Syynimaa (DrAzureAD)
- Principal Identity Security Researcher
- Microsoft Security Research (MSecR)

nsyynimaa@microsoft.com
@DrAzureAD

Contents

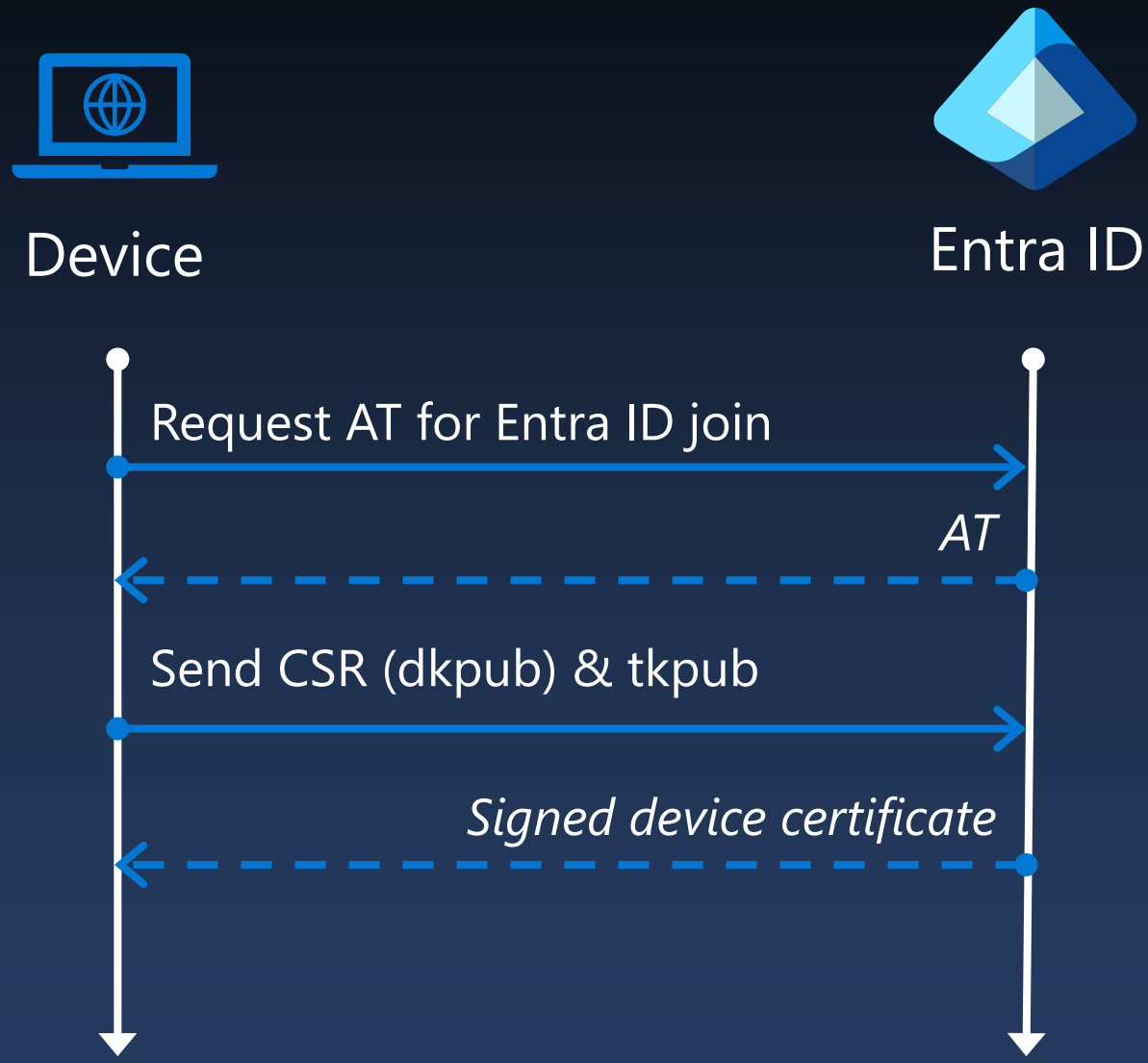
- Entra ID join recap
- Introduction to TPM
- Entra ID join with TPM
- Attacking TPM

Entra ID join recap

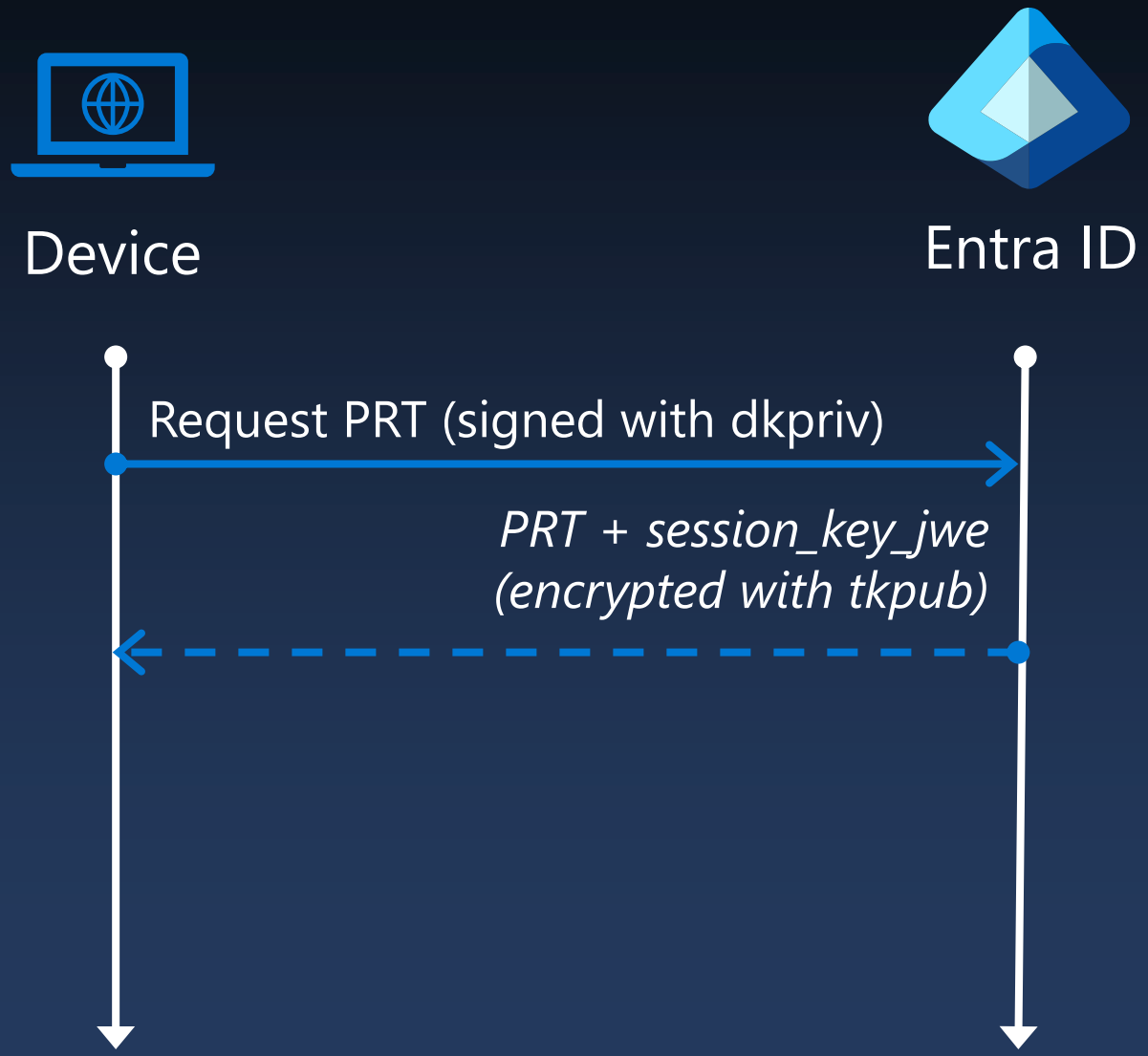
Entra ID Device Join process

- Two key pairs (*dkpub/dkpriv* and *tkpub/tkpriv*) are generated
- Public keys sent to Entra ID
- Device key (certificate) represents the device (*dkpub/dkpriv*)
- Transport key (*tkpub/tkpriv*) is used to encrypt/decrypt PRT *session key* (aka proof-of-possession (PoP) key)

Entra ID join

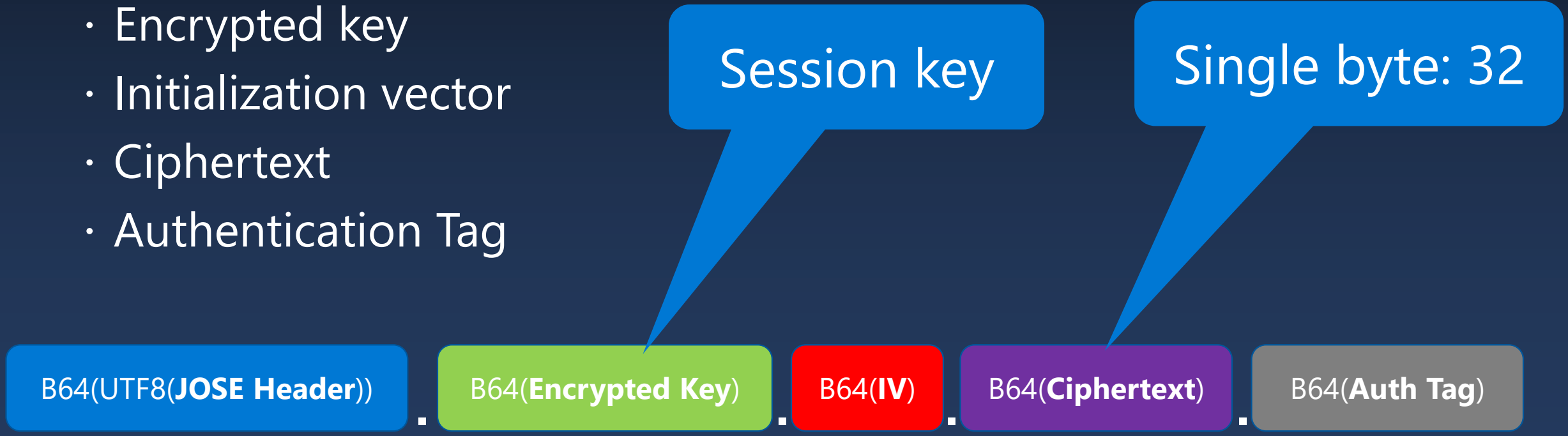


Request PRT



JSON Web Encryption (JWE)

- Used in Entra ID to return keys and encrypted data
- Five parts
  - JOSE Header
  - Encrypted key
  - Initialization vector
  - Ciphertext
  - Authentication Tag



https://www.rfc-editor.org/rfc/rfc7516.html

Where are the certs/keys stored (w/o TPM)?

Item	Entra?	Location on device
dkpub[1]	Yes	Cert:\LocalMachine\My\
dkpriv		C:\ProgramData\Microsoft\Crypto\Keys
tkpub	Yes	N/A
tkpriv		C:\ProgramData\Microsoft\Crypto\SystemKeys
PRT	Yes	C:\system32\config\systemprofile\AppData\local\microsoft\windows\CloudAPCache\AzureAD\<user's cache dir[2]>\Cache\CacheData
session key	Yes	

- All these are exportable by *Local Administrator*!

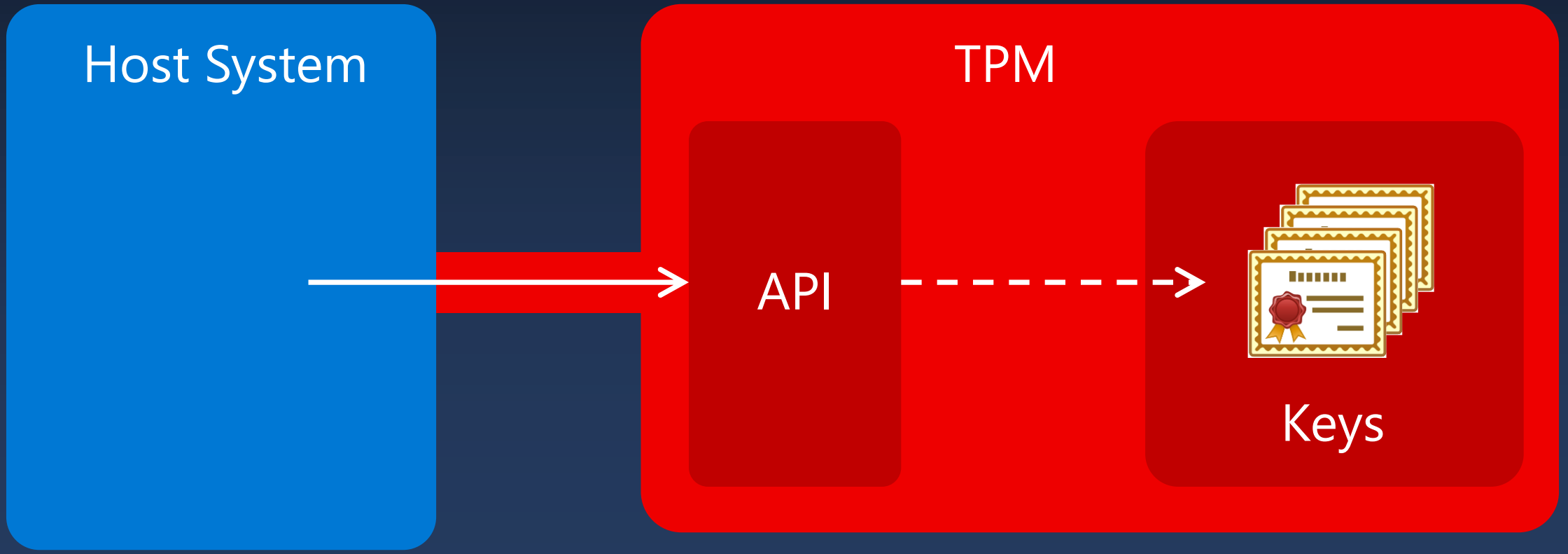
1. Part of device certificate
2. HKLM:\SOFTWARE\Microsoft\IdentityStore\LogonCache\B16898C6-A148-4967-9171-64D755DA8520\Name2Sid\

Entra ID Join Demo

Introduction to TPM

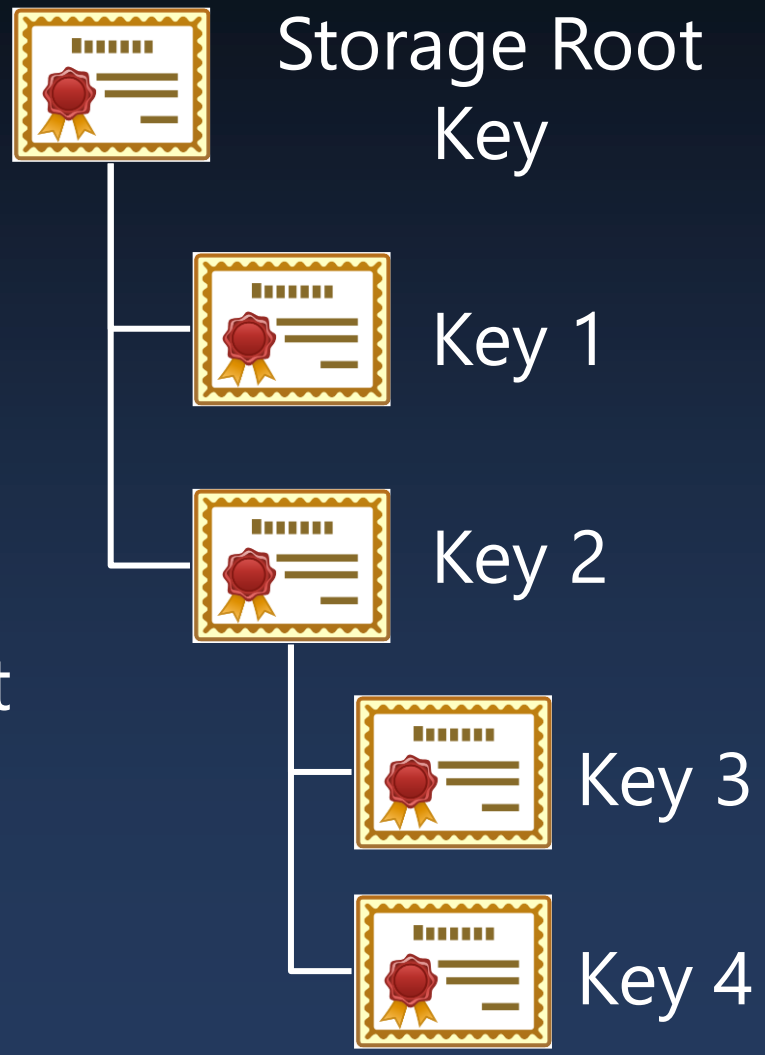
What is TPM?

- Trusted Platform Module
- A separate “computing unit”, only accessible via a “slow” I/O channel
- Host system can only call TPM API, no access to keys inside TPM



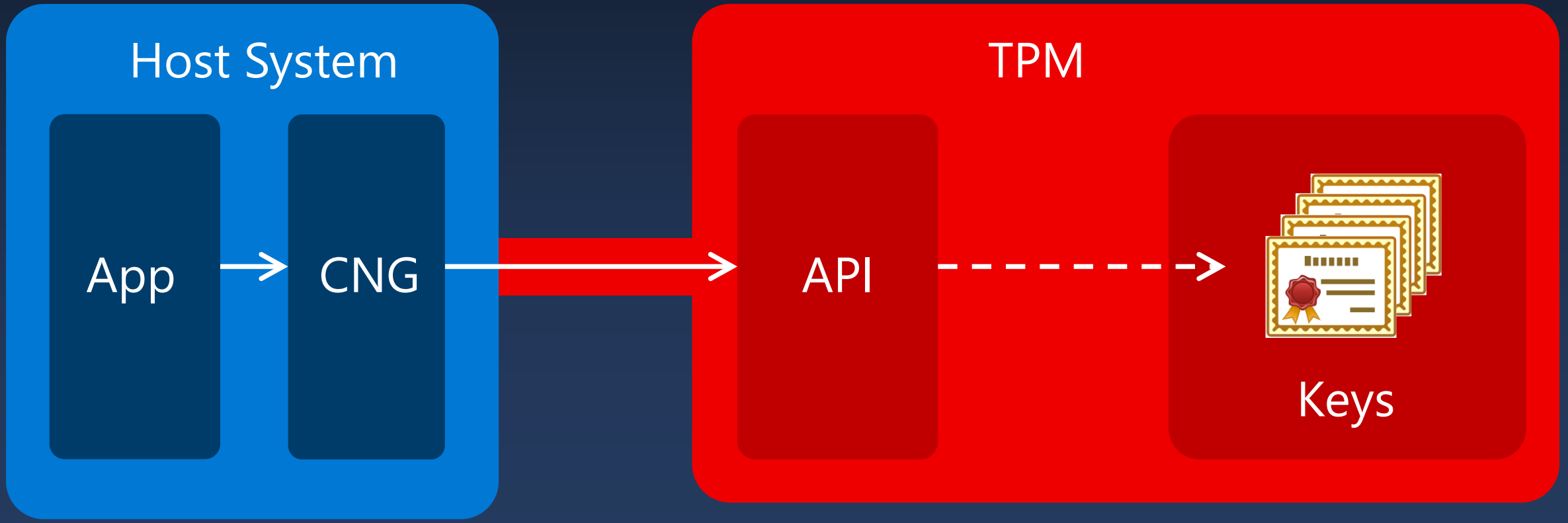
How keys are protected by TPM?

- When a key is **created**, it is encrypted with TPM **Storage Key** and returned to Host System
  - Keys are organised in a hierarchy – they have parents
  - Keys can also be **imported**
  - Keys can be persisted on TPM (not typical as space is limited)
- After creation or import, key can be **loaded** to TPM
- Parent object **must** be loaded before child object
- After loading, key can be used for cryptographic functions



How to use TPM on Windows - The easy way

- Use Cryptograpny API: Next Generation (CNG) with “Microsoft Platform Crypto Provider”
- CNG works the same way regardless are keys TPM backed or not



But what happens under-the-hood?

- Fiddler/Burp suite
  - Shows what happens between the endpoint and Entra ID
- Rohitab API Monitor:
  - Shows API calls, parameters, return values, etc.
  - Not all calls are shown, especially when monitoring LSASS
- x64dbg:
  - Live debugging
  - Shows everything..

Entra ID Join Demo 2

Tbsi_GetDeviceInfo function (tbs

02/22/2024

Obtains the version of the TPM on the computer.

Syntax

```
C++

TBS_RESULT Tbsi_GetDeviceInfo(
  [in]  UINT32 Size,
  [out] PVOID  Info
);
```

Parameters

`[in] Size`

Size of the memory location.

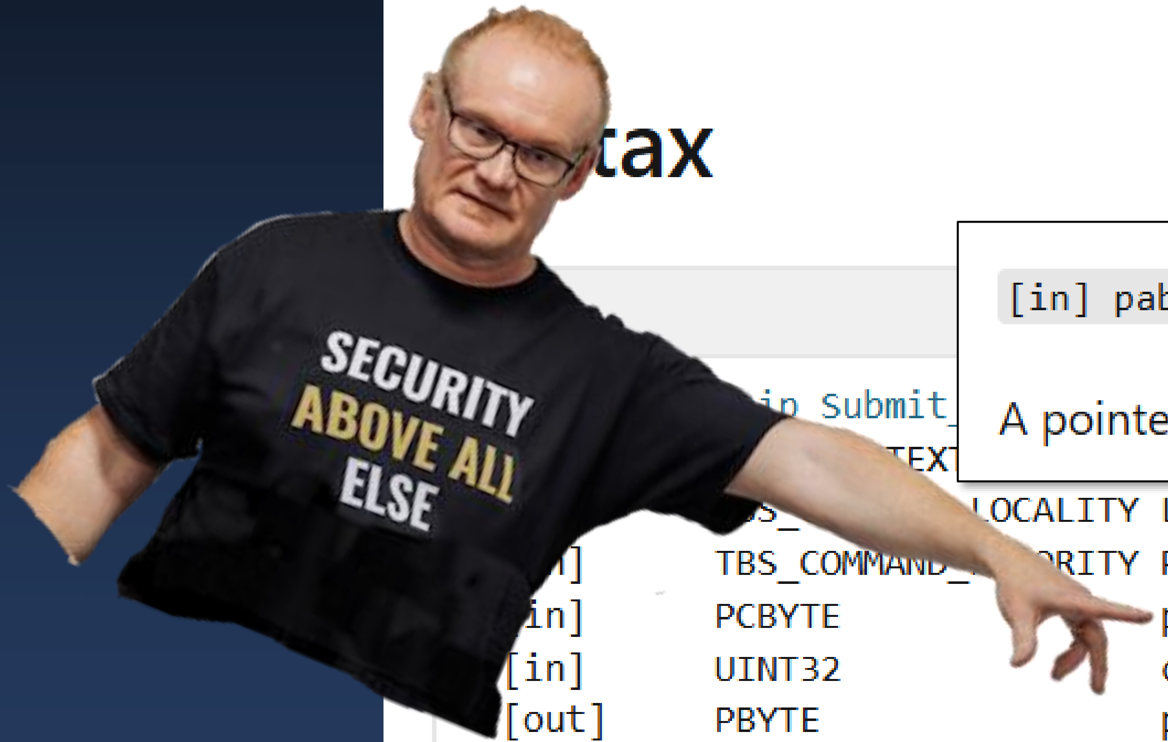
`[out] Info`

A pointer to a TPM_DEVICE_INFO structure is returned containing the ve

Tbsip_Submit_Command function (tbs.h)

Submits a Trusted Platform Module (TPM) command to TPM Base Services (TBS) for processing.

tax



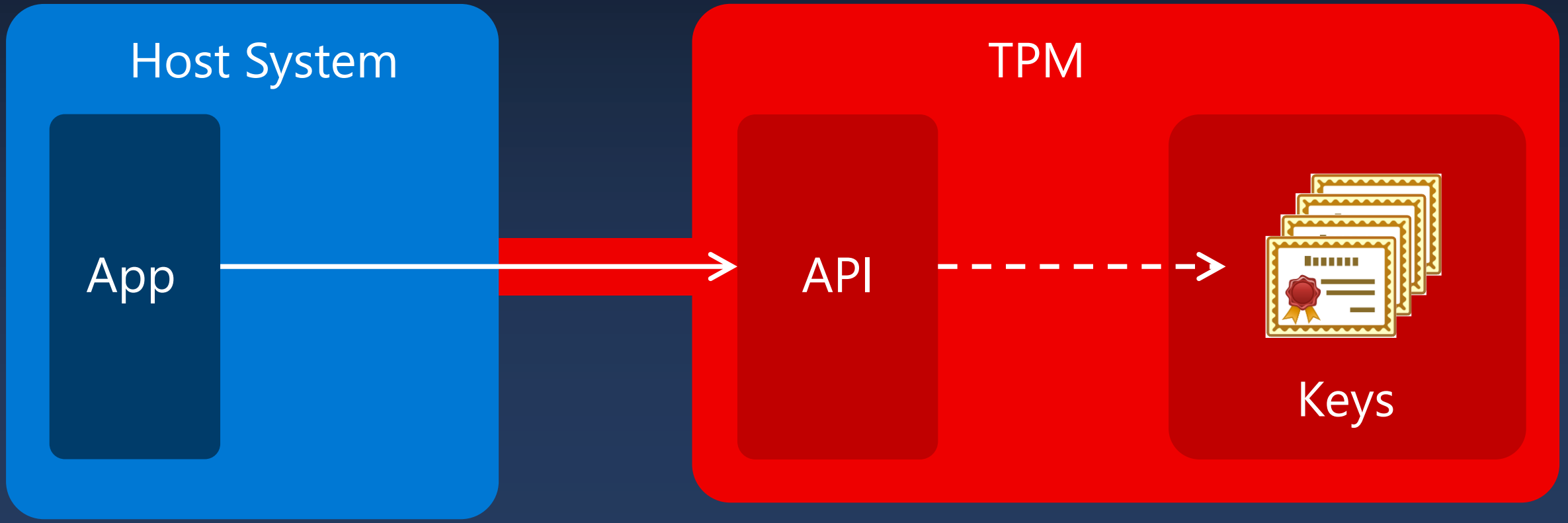
`[in] pabCommand`

A pointer to a buffer that contains the TPM command to process.

```
ip Submit_
TEXT
S_ LOCALITY Locality,
] TBS_COMMAND_ RITY Priority,
in] PCBYTE pabCommand,
[in] UINT32 cbCommand,
[out] PBYTE pabResult,
[in, out] PUINT32 pcbResult
);
```

How to use TPM on Windows - The hard way

- Communicate with TPM directly via Win32 TPM Base Services (tbs.dll)
- **Tbsip_Submit_Command** is used to send native (binary) TPM commands



me, going down the rabbit hole

Specification

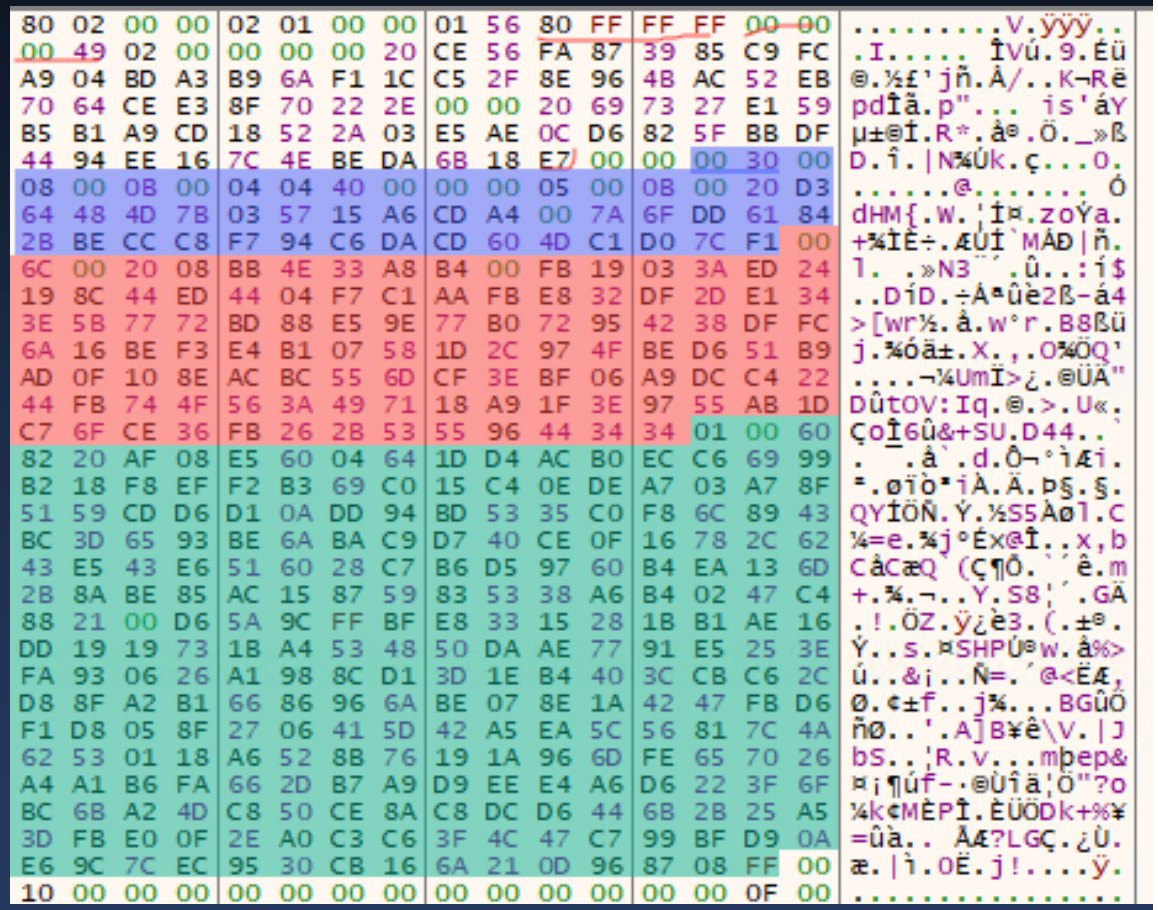
- Part 0: Introduction
- Part 1: Architecture
- Part 2: Structures
- Part 3: Commands

x64dbg Demo

TPM_CC_Load (TransportKey)

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	50	43	50	4D	38	00	00	00	02	00	00	00	02	00	00	00	PCPM8.....
00000010	3C	01	00	00	E0	00	00	00	00	00	00	00	00	00	00	00	<...à.....
00000020	B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	°.....
00000030	00	00	00	00	00	00	00	00	01	3A	00	01	00	0B	00	03	.....:.....
00000040	04	72	00	20	9D	FF	CB	F3	6C	38	3A	E6	99	FB	98	68	.r. .ÿËól8:æ™û˜h
00000050	DC	6D	CB	89	D7	15	38	84	BE	28	03	92	2C	12	41	58	ÜmË‰×.8„¾(.’,.AX
00000060	BF	AD	22	AE	00	06	00	80	00	43	00	10	08	00	00	00	¿.”®...€.C.....
00000070	00	00	01	00	A8	B7	52	BF	78	FA	A9	8A	1A	06	CF	E4	....¨·R¿xú©Š..Ïä
00000080	EB	07	29	03	6F	4B	A0	A9	C7	51	03	6F	A8	C4	E9	D9	ë.).oK ©ÇQ.o¨ÄéÙ
00000090	5F	A4	DC	3B	2C	82	B8	0D	2E	F5	2E	42	02	61	34	07	_¤Ü;‚¸..õ.B.a4.
000000A0	DC	13	C0	07	69	42	37	DA	53	14	5A	89	A3	DC	B6	04	Ü.À.iB7ÚS.Z‰£Ü¶.
000000B0	DA	D9	50	8B	2E	CF	FE	8F	18	FD	84	9E	FA	0C	BB	28	ÚÙP‹.Ïþ..ý„žú.»(
000000C0	DF	6A	A7	8E	D1	E4	B3	D4	5E	26	E9	FD	41	0D	61	8D	ßj§ŽÑä³Ô^&éýA.a.
000000D0	3F	4D	48	2E	64	56	E6	BC	8D	50	00	FC	80	C3	8E	1E	?MH.dVæ¼.P.ü€ÃŽ.
000000E0	07	DE	38	44	4B	05	F9	10	DD	04	F9	34	F5	37	C7	B3	.Þ8DK.ù.Ý.ù4õ7Ç³
000000F0	91	27	61	72	AB	C9	4F	AE	4D	C4	54	3C	72	F4	89	20	‘'ar«ÉO®MÄT<rô‰
00000100	6F	1E	85	87	9D	D3	08	FE	7F	B8	E3	ED	80	12	8C	E3	o.…‡.Ó.þ..¸ãí€.Œã
00000110	9E	82	15	11	05	8B	B3	72	B6	26	C8	1A	23	75	14	51	ž‚...‹³r¶&È.#u.Q
00000120	DB	BE	F6	F4	8D	73	CB	45	F5	62	A6	BD	71	72	CC	54	Û¾öô.sËEõb¦½qrÌT
00000130	3C	73	FA	03	57	E1	D4	0D	FF	88	D3	B9	4F	40	F7	53	<sú.WáÔ.ÿˆÓ¹O@÷S
00000140	6D	BB	9A	DF	E0	00	8F	42	35	5A	31	44	4A	2B	AA	1A	m»šßà..B5Z1DJ+ª.
00000150	50	A3	06	0F	C9	26	58	86	2F	63	08	1A	23	21	30	39	P£..É&X†/c..#!09
00000160	1F	FC	14	4C	EF	6F	61	36	66	9F	B3	13	2C	DE	29	32	.ü.Lïoa6fŸ³.,Þ)2
00000170	2E	30	0E	E5	00	DE	00	20	BE	14	55	85	C2	7A	A2	77	.0.å.Þ. ¾.U…Âz¢w
00000180	7B	5E	BA	3F	A1	26	0E	A8	71	EA	18	15	6C	8E	EA	68	{^º?¡&.¨qê..lŽêh
00000190	E7	EE	4B	8C	1B	6B	3A	7A	00	10	57	58	25	75	01	09	çîKŒ.k:z..WX%u..
000001A0	AE	E6	C9	95	DC	01	D0	D1	97	33	55	51	46	BB	BE	27	®æÉ•Ü.ÐÑ—3UQF»¾'
000001B0	A8	8E	CE	2F	C1	3F	4A	08	CE	35	9E	48	3C	1C	7A	A4	¨ŽÎ/Á?J.Î5žH<.z¤
000001C0	9A	A6	26	A1	69	68	05	8A	AA	23	BB	AA	A5	F2	38	DF	š¦&¡ih.Šª#»ª¥ò8ß
000001D0	FD	2A	49	7A	E8	B2	7B	82	B7	7C	3C	AE	40	BF	11	B1	ý*Izè²{‚·|<®@¿.±
000001E0	3E	47	FF	8C	95	AB	47	25	56	7C	83	99	2A	A0	49	CE	>GÿŒ•«G%V|ƒ™* IÎ
000001F0	17	2A	50	9C	9F	FE	03	1A	FF	DB	48	8B	8C	08	D4	C8	.*PœŸþ..ÿÛH‹Œ.ÔÈ
00000200	2E	DA	26	69	25	9C	2B	A7	31	92	76	F6	88	E8	9B	E4	.Ú&i%œ+§1’vö^è›ä
00000210	A3	A3	F9	B0	0C	26	A8	4E	A0	44	20	95	11	F2	DD	3C	££ù°.&¨N D •.òÝ<
00000220	B6	83	D1	8C	0D	48	BA	EC	97	F5	D0	E8	C4	1E	BD	5E	¶ƒÑŒ.Hºì—õÐèÄ.½^
00000230	B9	92	65	B4	D8	45	D9	FD	27	49	EC	A6	CB	DD	58	58	¹’e´ØEÙý'Ìì¦ËÝXX
00000240	65	AB	AE	6D	A2	77	9C	5C	74	12	F0	AB	BA	0C	94	C5	e«®m¢wœ\t.ð«º.”Å
00000250	15	6E	2F	61	00	00	00	06	00	20	8F	CD	21	69	AB	92	.n/a..... .Í!i«’
00000260	69	4E	0C	63	3F	1A	B7	72	84	2B	82	41	BB	C2	02	88	iN.c?.·r„+‚A»Â.ˆ
00000270	98	1F	C7	AC	1E	DD	C1	FD	DB	0E	00	20	E5	29	F5	D6	˜.Ç¬.ÝÁýÛ.. å)õÖ
00000280	11	28	72	95	4E	8E	D6	60	51	17	B7	57	E2	37	C6	E1	.(r•NŽÖ`Q.·Wâ7Æá
00000290	95	13	A9	49	FE	E1	F2	04	C4	58	02	3A	00	20	AF	2C	•.©Iþáò.ÄX.:. ¯,
000002A0	A5	69	69	9C	43	6A	21	00	6F	1C	B8	A2	75	6C	98	BC	¥iiœCj!.o.¸¢ul˜¼
000002B0	1C	76	5A	35	59	C5	FE	1C	3F	5E	72	28	A7	E7	00	20	.vZ5YÅþ.?^r(§ç.
000002C0	C4	13	A8	47	B1	11	12	B1	CB	DD	D4	EC	A4	DA	AA	15	Ä.¨G±..±ËÝÔì¤Úª.
000002D0	A1	85	2C	1C	3B	BA	57	46	1D	25	76	05	F3	D5	AF	53	¡…,.;ºWF.%v.óÕ¯S
000002E0	00	00	00	20	04	8E	9A	3A	CE	08	58	3F	79	F3	44	FF	... .ŽšÎ.X?yóDÿ
000002F0	78	5B	BE	A9	F0	7A	C7	FA	33	25	B3	D4	9A	21	DD	51	x[¾©ðzÇú3%³Ôš!ÝQ
00000300	94	C6	58	50	86	B7	70	94	9C	C3	92	69	E2	4A	E1	D8	”ÆXP†·p”œÃ’iâJáØ

TPM_CC_Import (SessionKey, aka PoPKey)



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	01	00	60	82	20	AF	08	E5	60	04	64	1D	D4	AC	B0	EC	..`, ¯.å`.d.Ô¬°ì
00000010	C6	69	99	B2	18	F8	EF	F2	B3	69	C0	15	C4	0E	DE	A7	Æi™².øïò³iÀ.Ä.Þ§
00000020	03	A7	8F	51	59	CD	D6	D1	0A	DD	94	BD	53	35	C0	F8	.§.QYÍÖÑ.Ý"½S5Àø
00000030	6C	89	43	BC	3D	65	93	BE	6A	BA	C9	D7	40	CE	0F	16	l‰C¼=e"¾j°É×@Î..
00000040	78	2C	62	43	E5	43	E6	51	60	28	C7	B6	D5	97	60	B4	x,bCåCæQ`(Ç¶Õ—`´
00000050	EA	13	6D	2B	8A	BE	85	AC	15	87	59	83	53	38	A6	B4	ê.m+Š¾…¬.‡Yƒ S8¦´
00000060	02	47	C4	88	21	00	D6	5A	9C	FF	BF	E8	33	15	28	1B	.GÄˆ!.ÖZœÿ¿è3.(.
00000070	B1	AE	16	DD	19	19	73	1B	A4	53	48	50	DA	AE	77	91	±®.Ý..s.¤SHPÚ®w'
00000080	E5	25	3E	FA	93	06	26	A1	98	8C	D1	3D	1E	B4	40	3C	å%>ú".&¡˜ŒÑ=.´@<
00000090	CB	C6	2C	D8	8F	A2	B1	66	86	96	6A	BE	07	8E	1A	42	ËÆ,Ø.¢±f†–j¾.Ž.B
000000A0	47	FB	D6	F1	D8	05	8F	27	06	41	5D	42	A5	EA	5C	56	GûÖñØ..'.A]B¥ê\V
000000B0	81	7C	4A	62	53	01	18	A6	52	8B	76	19	1A	96	6D	FE	.|JbS..¦R‹v..–mþ
000000C0	65	70	26	A4	A1	B6	FA	66	2D	B7	A9	D9	EE	E4	A6	D6	ep&¤¡¶úf-·©ÙîäÖ
000000D0	22	3F	6F	BC	6B	A2	4D	C8	50	CE	8A	C8	DC	D6	44	6B	"?o¼k¢MÈPÎŠÈÜÖDk
000000E0	2B	25	A5	3D	FB	E0	0F	2E	A0	C3	C6	3F	4C	47	C7	99	+%¥=ûà.. ÃÆ?LGÇ™
000000F0	BF	D9	0A	E6	9C	7C	EC	95	30	CB	16	6A	21	0D	96	87	¿Ù.æœ|ì•0Ë.j!.–‡
00000100	08	FF	00	6C	00	20	08	BB	4E	33	A8	B4	00	FB	19	03	.ÿ.l. .»N3¨´.û..
00000110	3A	ED	24	19	8C	44	ED	44	04	F7	C1	AA	FB	E8	32	DF	:í$.ŒDíD.÷Áªûè2ß
00000120	2D	E1	34	3E	5B	77	72	BD	88	E5	9E	77	B0	72	95	42	-á4>[wr½ˆåžw°r•B
00000130	38	DF	FC	6A	16	BE	F3	E4	B1	07	58	1D	2C	97	4F	BE	8ßüj.¾óä±.X.,—O¾
00000140	D6	51	B9	AD	0F	10	8E	AC	BC	55	6D	CF	3E	BF	06	A9	ÖQ¹...Ž¬¼UmÏ>¿.©
00000150	DC	C4	22	44	FB	74	4F	56	3A	49	71	18	A9	1F	3E	97	ÜÄ"DûtOV:Iq.©.>—
00000160	55	AB	1D	C7	6F	CE	36	FB	26	2B	53	55	96	44	34	34	U«.ÇoÎ6û&+SU–D44
00000170	00	30	00	08	00	0B	00	04	04	40	00	00	00	05	00	0B	.0.......@......
00000180	00	20	D3	64	48	4D	7B	03	57	15	A6	CD	A4	00	7A	6F	. ÓdHM{.W.¦Í¤.zo
00000190	DD	61	84	2B	BE	CC	C8	F7	94	C6	DA	CD	60	4D	C1	D0	Ýa„+¾ÌÈ÷”ÆÚÍ`MÁÐ
000001A0	7C	F1															|ñ

Entra ID join with TPM

Entra ID Device Join process w/ TPM

- Two key pairs (*dkpub*/*dkpriv* and *tkpub*/*tkpriv*) are generated by TPM
- Key blobs are sent to Entra ID
- Session key is a three-part blob to be **imported** to TPM
  - Resulting blob is encrypted with TPM and can be **loaded** to TPM

* Session key blob parts

Part	Explanation
objectPublic	The public area of TPM object (session key)
duplicate	The symmetrically encrypted TPM object (session key)
inSymSeed	The seed for the symmetric key and HMAC key, encrypted using **tkpub**

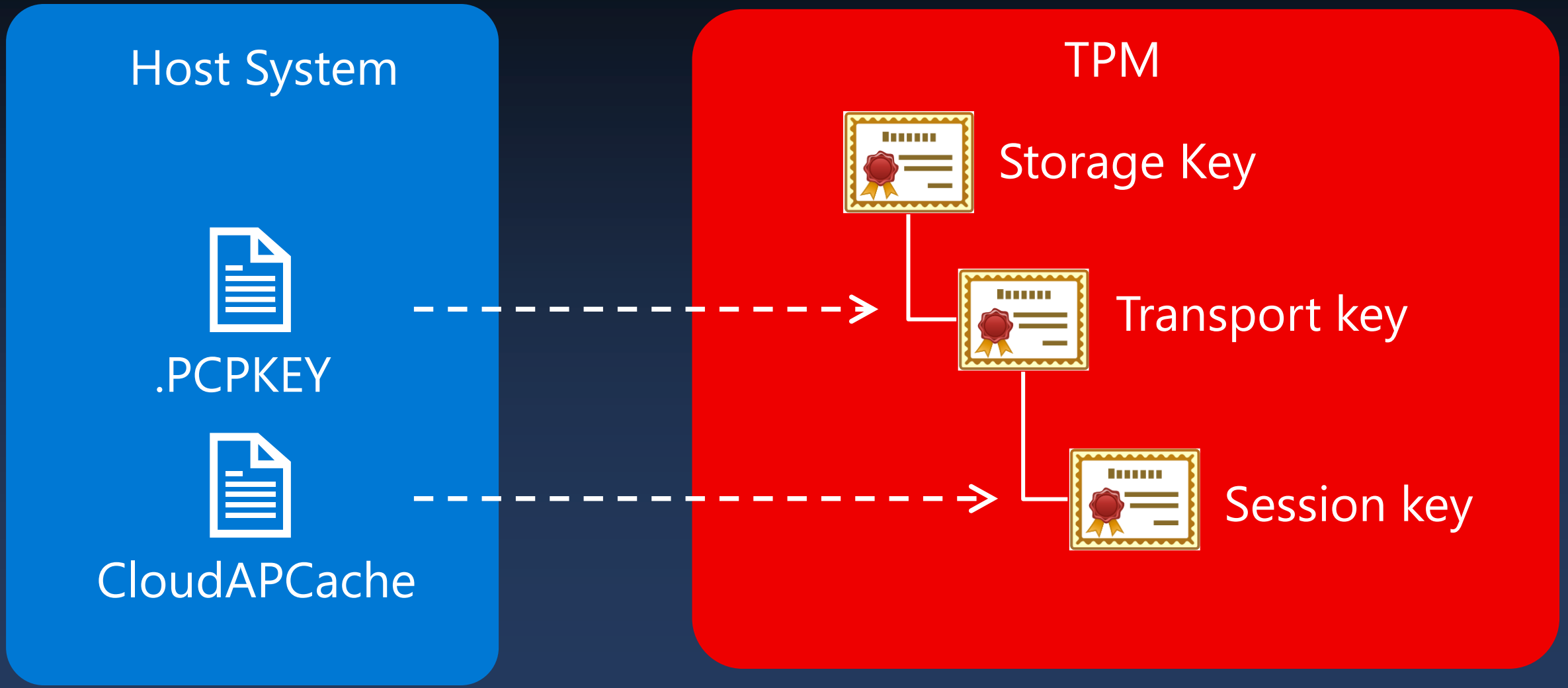
Where are the certs/keys stored (w/ TPM)?

Item	Entra?	Location on device
dkpub*	Yes	Cert:\LocalMachine\My\
dkpriv	Yes	C:\ProgramData\Microsoft\Crypto\PCPKSP\<path>\<keyname>.PCPKEY
tkpub	Yes	C:\Windows\System32\config\systemprofile\AppData\Local\Microsoft\Crypto\PCPKSP\<path>\<keyname>.PCPKEY
tkpriv	Yes	
PRT	Yes	C:\system32\config\systemprofile\AppData\local\microsoft\windows\CloudAPCache\AzureAD\<user's cache dir>\Cache\CacheData
session key	Yes	

- All these are exportable by *Local Administrator* but unusable w/o access to TPM

* Part of device certificate

How keys are used with TPM



Attacking TPM

Research setup

- When on target device, attacker can:
  - Request access tokens with PRT & session key
  - Request new PRT and session key with device key
- Research question
  - Can session key be exported from TPM?
- Four tests
  - Duplicate session key
  - Duplicate a key with authPolicy
  - Try to provide authPolicy when importing session key
  - Try to decrypt session key private parts with tkpriv

Test 1: Duplicate session key

- **TPM2_Duplicate** command dupli[illegible]tes loa[illegible]bject to be used with other parent (can b[illegible]utside TPM)
- Documentation:
  - "If *fixedParent* of *objectHandle* attr[illegible]tes [illegible]ET, the TPM shall return TPM_RC_ATTRIBUTES"
  - "The *policySession* command[illegible]e p[illegible]meter in the policy session is required to be **TPM_CC_Duplicat[illegible]** to indicate that auth[illegible]tion for duplication has been provided. "
  - "If the compared ha[illegible]s are not the same, then the TPM shall return TPM_RC_POLICY_FAIL."
- Attributes has n[illegible]ixedTPM/FixedParent set!
- **authPolicy** is empty

FAILED

```
FixedTPM              : False
StClear               : False
FixedParent           : False
SensitiveDataOrigin   : False
UserWithAuth          : True
```

```
0x0000009d: Format Error:. Handle: 0x0. Error: TPM_RC_POLICY_FAIL.  Description: a policy check failed
At C:\Users\PCAdministrator\Desktop\Invoke_commands.ps1:195 char:13
+            throw "$($errorValue): $([TpmRcDecoder.Decoder]::new().De ...
+            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (0x0000009d: For...cy check failed:String) [], RuntimeException
    + FullyQualifiedErrorId : 0x0000009d: Format Error:. Handle: 0x0. Error: TPM_RC_POLICY_FAIL.  Description: a policy check failed
```

Test 2: Duplicate a key with authPolicy set

- Used **TPM2_Create** to create a new RSA key
  - **authPolicy** with **TPM[illegible]_Duplica[illegible]** w[illegible] provided
- **TPM_CC_D[illegible]cate** [illegible]ccess[illegible]ul

PASSED

```
ins[illegible]tive
[illegible] 07 size
00 [illegible] us[illegible]er[illegible] s[illegible]e
66 [illegible] 6[illegible] bu[illegible]r
00 [illegible] data s[illegible]

[illegible]Pu[illegible]c
[illegible]0 36 si[illegible]
00 [illegible]ype (TPM_ALG_RSA)
[illegible]0 0B nameAlg (TPM_ALG_SHA256)
00 02 00 60 objectAttributes
(sensitiveDataOrigin|userWithAuth|decrypt
00 20 authPolicy size
authPolicy (TPM_CC_Duplicate)
BE F5 6B 8C 1C C8 4E 11 ED D7 17 52 8D 2C D9 93
56 BD 2B BF 8F 01 52 09 C3 F8 4A EE AB A8 E8 A2
00 10 algorithm (TPM_ALG_NULL)
00 10 scheme (TPM_ALG_NULL)
08 00 keyBits (2048)
00 00 00 00 exponent (default, 65537)
00 00 RSA unique size (empty buffer)
```

Test 3: Provide authPolicy when importing session key

- The original *inpublic* **authPolicy** was empty buffer

```
00 30 Size
00 08 Type (TPM_ALG_KEYEDHASH)
00 0B Name alg (TPM_ALG_SHA256 )
00 04 04 40 objectAttributes (userWithAuth|NoDA|sign)
00 00 authPolicy (empty buffer)
00 05 scheme: hashAlg (ALG_HMAC_VALUE)
00 0B scheme details: hashAlg (TPM_ALG_SHA256)
unique
00 20 D3 64 48 4D 7B 03 57 15 A6 CD A4 00 7A 6F DD
61 84 2B BE CC C8 F7 94 C6 DA CD 60 4D C1 D0 7C F1
```

- Replaced by **TPM_CC_Duplicate** policy

```
00 50
00 08
00 0B
00 04 04 40
00 20  size
authPolicy TPM_CC_Duplicate
BE F5 6B 8C 1C C8 4E 11 ED D7 17 52 8D 2C D9 93
56 BD 2B BF 8F 01 52 09 C3 F8 4A EE AB A8 E8 A2
00 05
00 0B
00 20 D3 64 48 4D 7B 03 57 15 A6 CD A4 00 7A 6F DD
61 84 2B BE CC C8 F7 94 C6 DA CD 60 4D C1 D0 7C F1
```

Test 3: Provide authPolicy when importing session key

- Documentation:
  - "TPM will use a **secure hash fun** to validat at the ed as properly protected and not altered. If the integrit eck fails th PM re ns an e r and does not load the object."
- The *inSymS* is enc pte wit kp u but as don't know the content, hash ouldn t re alculated
- **TPM_CC_Load** nsuc essful

FAILED

```
0x0000009f: Format Error:. Handle: 0x0. Error: TPM_RC_INTEGRITY.  Description: integrity check failed
At C:\Users\PCAdministrator\Desktop\Invoke_commands.ps1:195 char:13
+             throw "$($errorValue): $([TpmRcDecoder.Decoder]::new().De ...
+             ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (0x0000009f: For...ty check failed:String) [], RuntimeException
    + FullyQualifiedErrorId : 0x0000009f: Format Error:. Handle: 0x0. Error: TPM_RC_INTEGRITY.  Description: integrity check failed
```

Test 4: Decrypt session key private parts with tkpriv

- **TPM2_RSA_Decrypt** command allows decrypting using TPM protected keys
- Documentation:
  - "The key referenced by *keyHandle* shall be an RSA key (TPM_RC_KEY) with *restricted* CLEAR and *decrypt* SET (TPM_RC_ATTRIBUTES)."
  - "A key that has both *decrypt* and *restricted* attributes SET only accepts data that has a specific structure. The encrypted data block must have as its first element an integrity value for the remainder of the structure. **This integrity value is an HMAC of the encrypted data.** This format allows the TPM to prevent misuse of the restricted decryption keys that are the basis of the protected storage hierarchy."

Test 4: Decrypt session key private parts with tkpriv

- Attributes had *restricted* AND *decrypt* set
- **TPM_CC_RSA_Decrypt** *inSymSeed* failed

FAILED

```
FixedTPM              : True
StClear               : False
FixedParent           : True
SensitiveDataOrigin   : True
UserWithAuth          : True
AdminWithPolicy       : False
NoDA                  : True
EncryptedDuplication  : False
Restricted            : True
Decrypt               : True
Sign                  : False
```

```
0x00000082: Format Error:. Handle: 0x0. Error: TPM_RC_ATTRIBUTES.  Description: inconsistent attributes
At C:\Users\PCAdministrator\Desktop\Invoke_commands.ps1:195 char:13
+             throw "$($errorValue): $([TpmRcDecoder.Decoder]::new().De ...
+             ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (0x00000082: For...tent attributes:String) [], RuntimeException
    + FullyQualifiedErrorId : 0x00000082: Format Error:. Handle: 0x0. Error: TPM_RC_ATTRIBUTES.  Description: inconsistent attributes
```

Summary

Summary

- w/o TPM threat actor can
  - Export dkpub/dkpriv, tkpriv, PRT, and session key and
  - Use them locally and offline
- w/ TPM threat actor can
  - Export dkpub/dkpriv, tkpub/tkpriv, PRT, and session key and
  - Use them locally
- Keys are safe with TPM 😊
- To detect token theft/replay:
  - Monitor access to key material (endpoint)
  - Monitor authentication events (Entra ID)

Microsoft Security

DANKE SCHÖN!

Questions?