Secureworks®
a SOPHOS company

Hopping Across Devices: Expanding Lateral Movement through Pass-the-Certificate Attack

Yuya Chudo

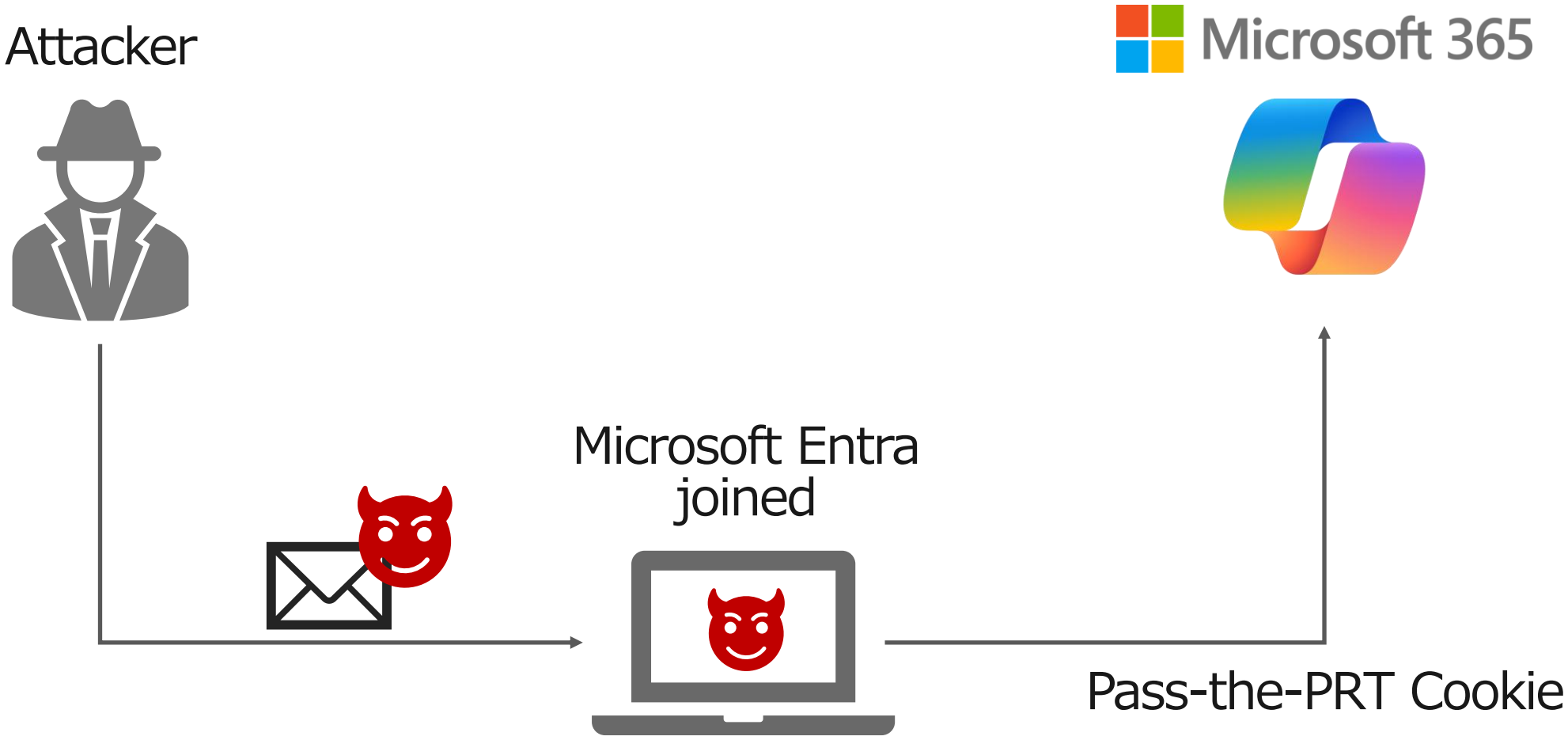
TROOPERS

WHOAMI

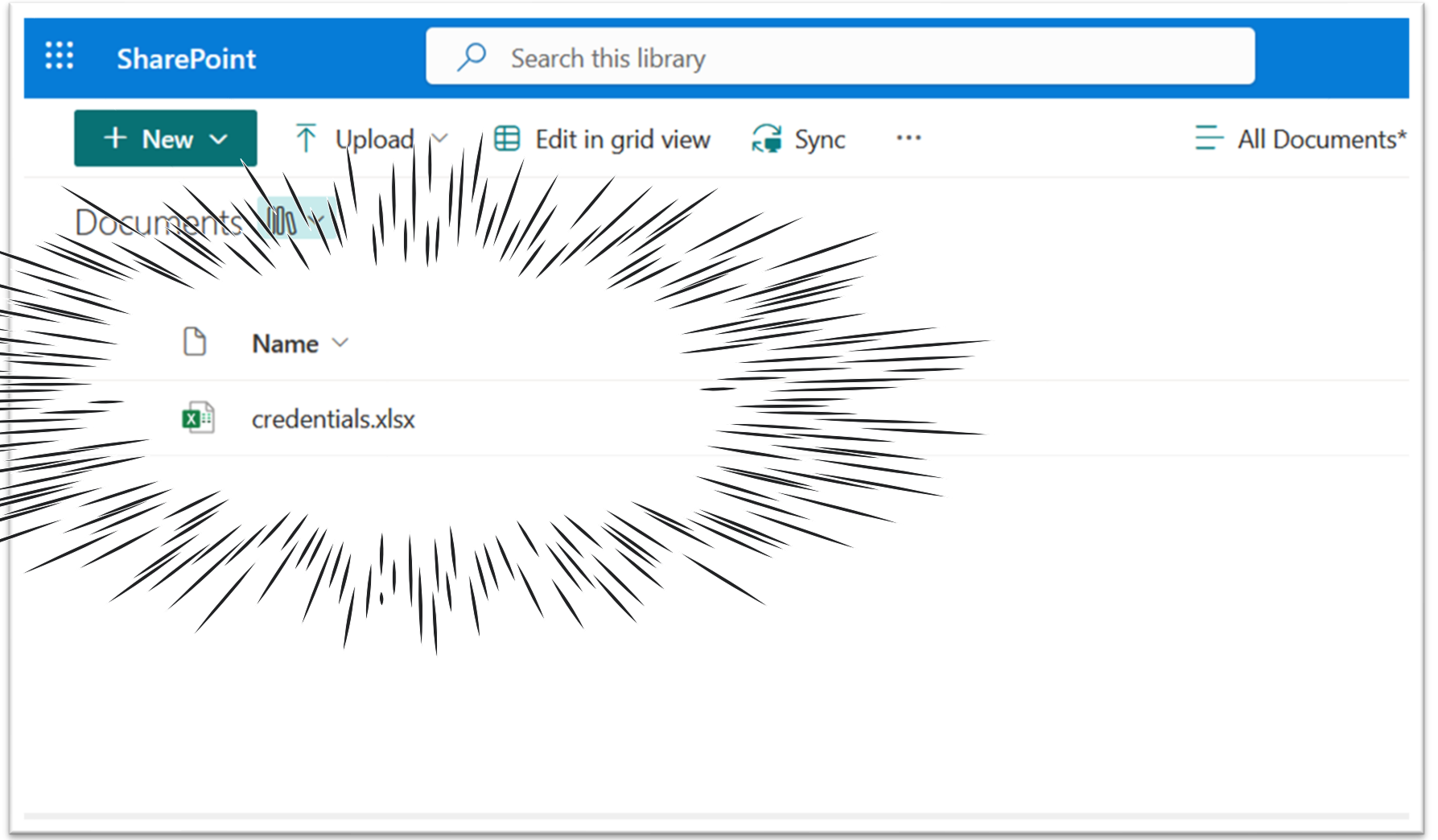
Yuya Chudo

- Japan
- Interested in Windows / Entra ID internals
- Red Team

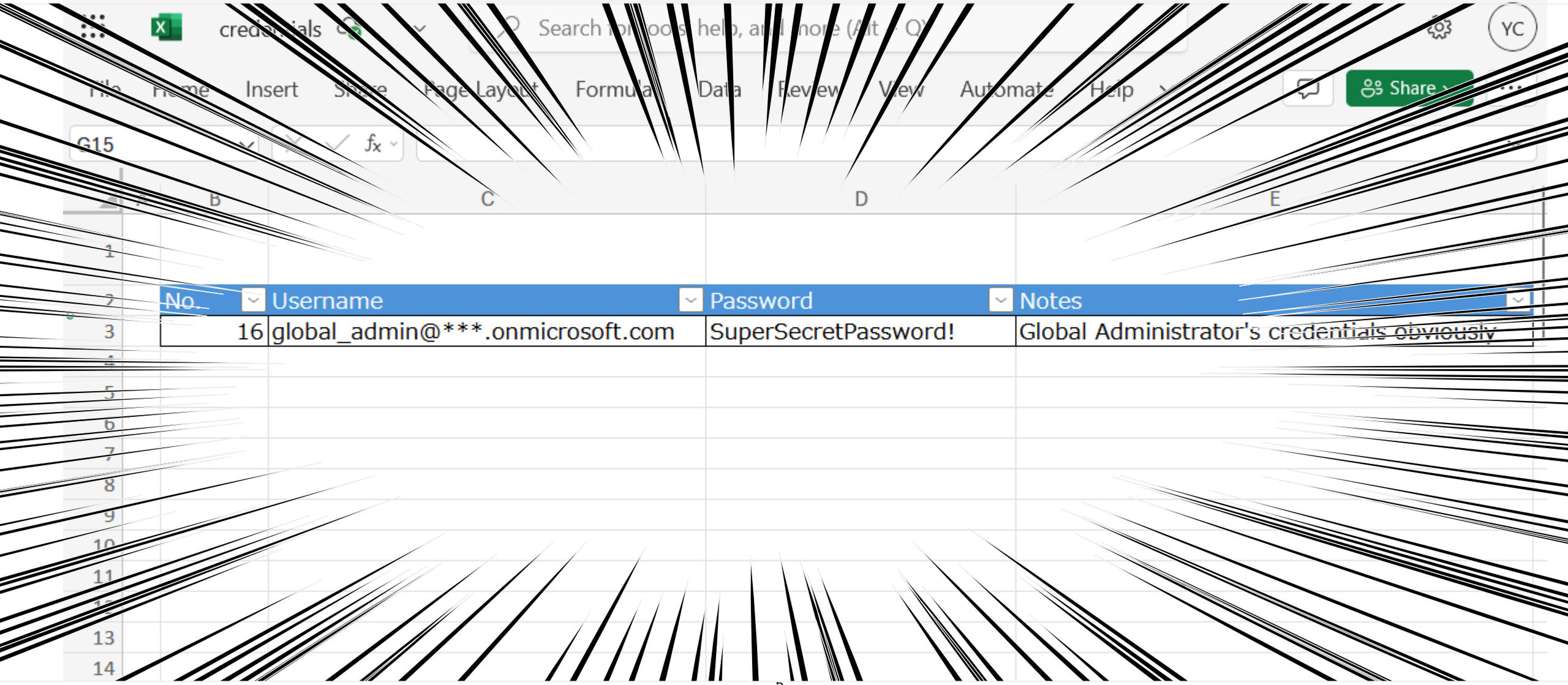
One Day in a Red Team Engagement



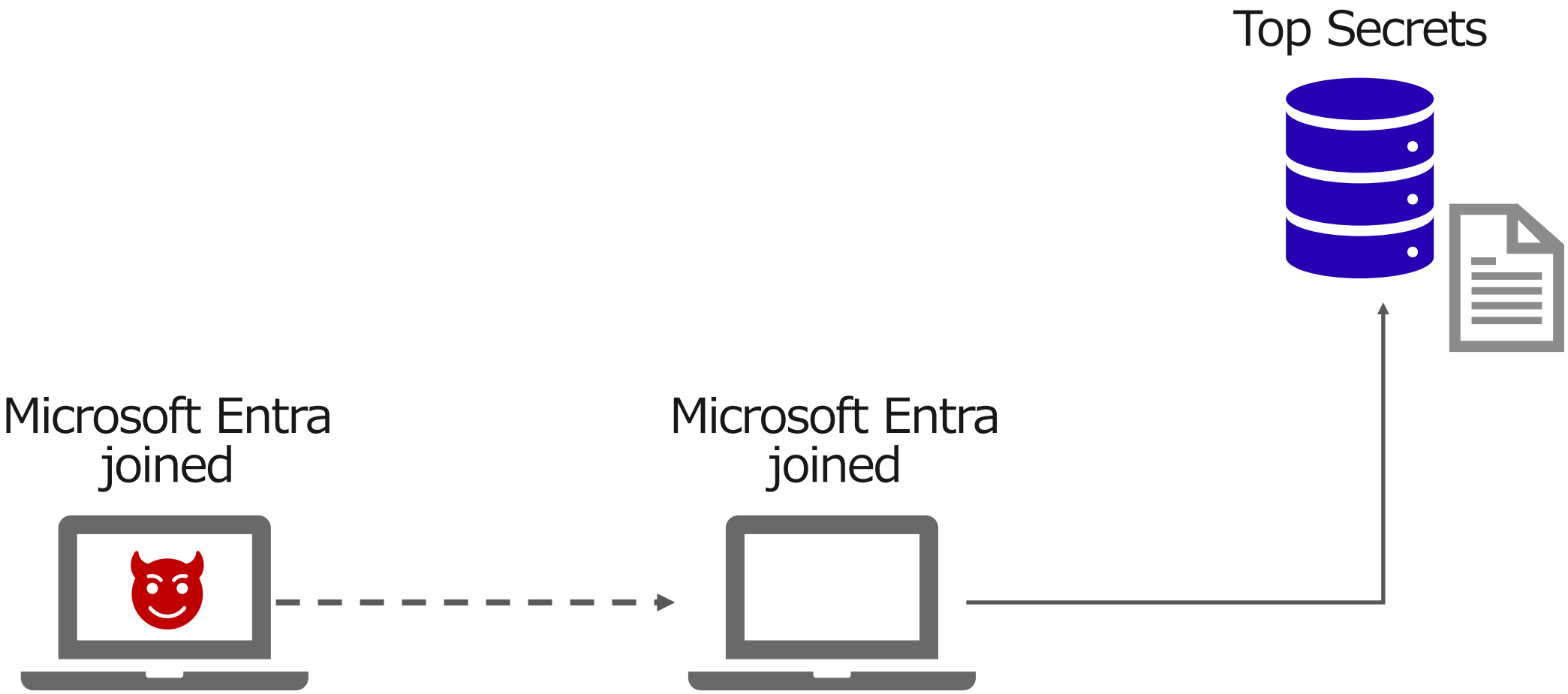
One Day in a Red Team Engagement



One Day in a Red Team Engagement



One Day in a Red Team Engagement



Lateral Movement in Microsoft Entra

Run OS commands through:

- ✓ **Microsoft Intune**
- ✓ **Azure portal/Azure CLI**
- ✓ **Defender** etc…

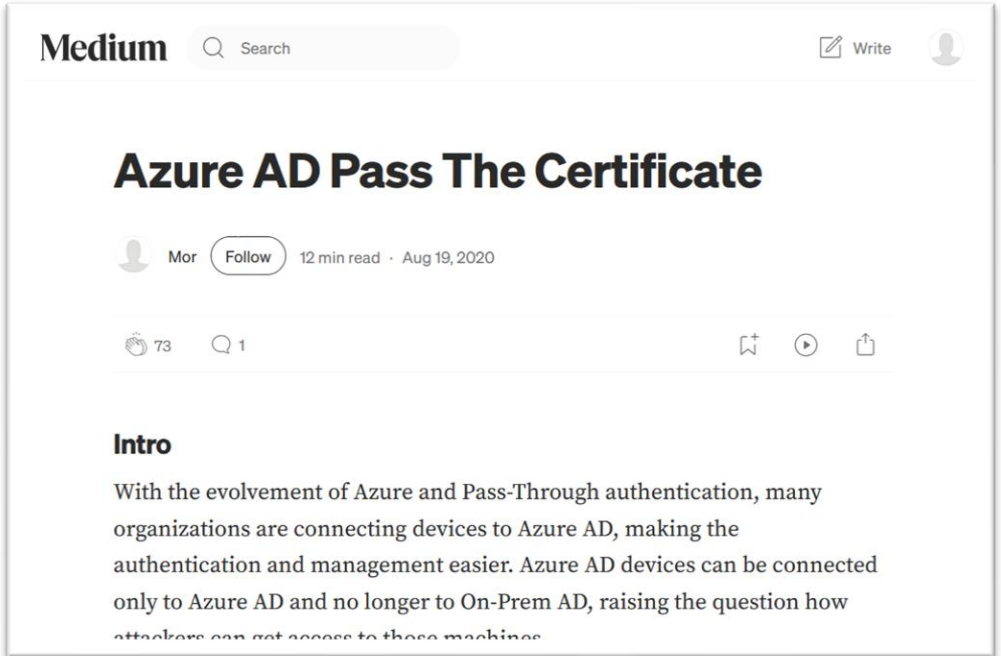
Certain conditions may prevent these methods

(MFA, device configurations···)

Lateral Movement in Microsoft Entra

Or you could also try **Pass-the-Certificate**?

- Introduced by Mor Rubin in 2020 in his blog and in Black Hat USA 2022



https://medium.com/@mor2464/azure-ad-pass-the-certificate-d0c5de624597

https://i.blackhat.com/USA-22/Wednesday/US-22-Rubin-AAD-Joined-Machines-New-Lateral-Movement.pdf

This talk is mainly about…

- What is “Pass-the-Certificate”?
- How can we utilize the technique?
- How can we prevent and detect it?

Dive into Pass-the-Certificate

Pass-the-Certificate

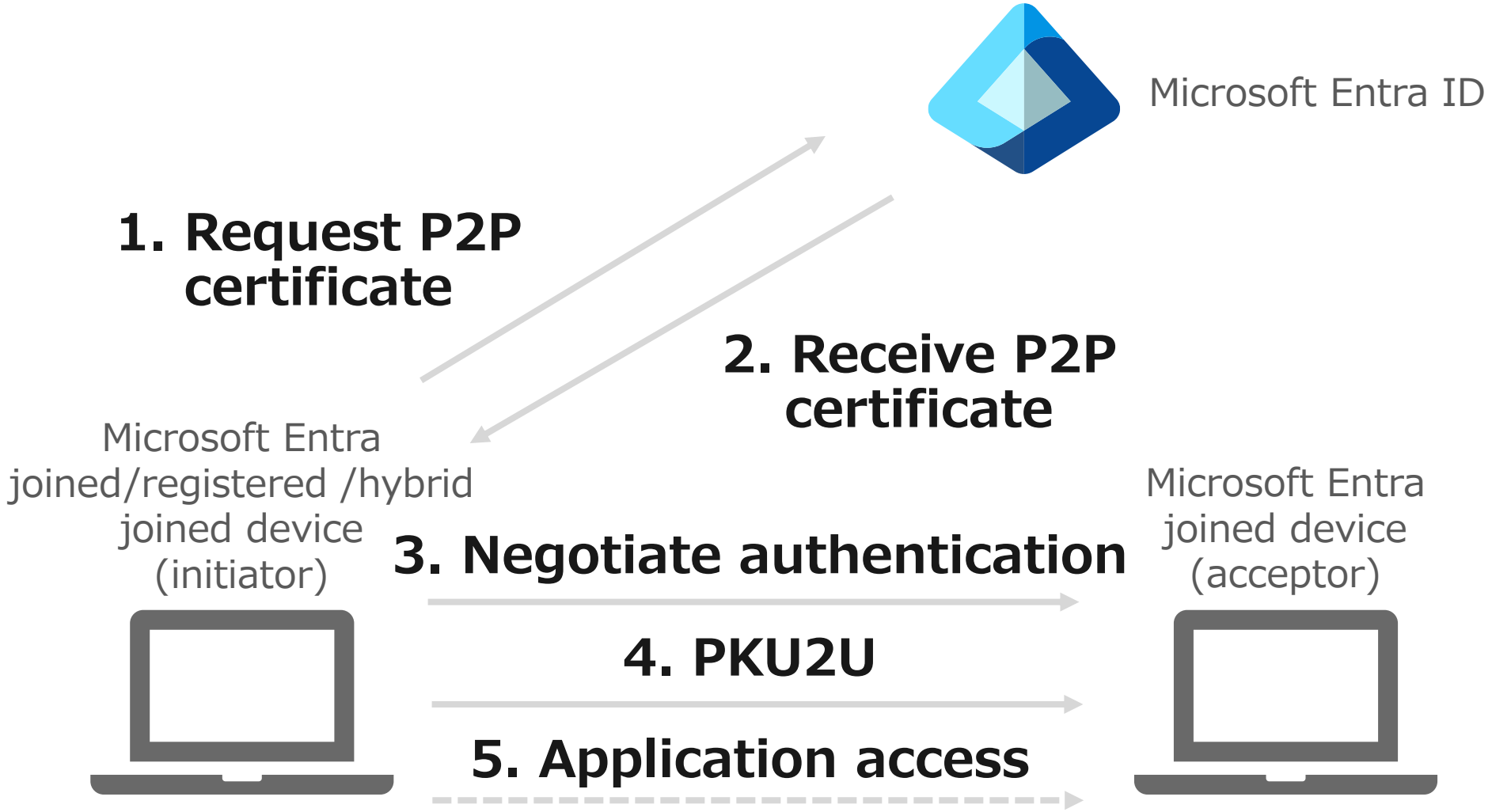
- Request **P2P certificate** of users that have admin access to a target Microsoft Entra joined device (Ex. Global Administrator, Local Device Administrator, and the owner of the device)
- **Pass the certificate** to the device for authentication and execute OS commands for lateral movement

Authentication to Microsoft Entra joined devices

How can we remotely authenticate to Microsoft Entra joined devices?

- ✓ **NTLM**
- ✓ **RDS AAD Auth**
- ✓ **Public Key Cryptography Based User-to-User (PKU2U)**

Connect to Microsoft Entra joined device via PKU2U



1. Request P2P certificate

- Send P2P certificate request to Microsoft Entra ID

```
POST /common/oauth2/token HTTP/1.1
Host: login.microsoftonline.com
User-Agent: python-requests/2.31.0
Accept-Encoding: gzip, deflate, br
Accept: */*
Connection: keep-alive
Content-Length: 3252
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer&request=eyJhbGciOiJIUzI1NiIsIm
N0eCI6Ik0wUnFtUlR5NzhSY1...Tld6cTl0ejFGRDFHeUkwajBXalRBcWRFTitndnFQR0NDOUw1ZHFPd09RcXlodEVLdk9y
In0.XyZfNx3gsZ0o-T63h2UQv-lTNh-oXH2mu1O2yT19yvQ&windows_api_version=2.2
```

1. Request P2P certificate

- JSON Web Tokens (JWT) containing user's PRT and CSR

Header

```
{
  "alg": "HS256",
  "ctx": "M0RqmRTy78RcWi6VAPtjgemQ4/nLQHWL",
  "kdf_ver": 2
}
.
```

Payload

```
{
  "iss": "aad:brokerplugin",
  "grant_type": "refresh_token",
  "aud": "login.microsoftonline.com",
  "request_nonce": "AwABEgEAAAADAOz_BQD0_3sDavUg2uEMCEoS5M9bBJTdMJBr0UlliKKcngBzgUIXhRxjrsvn8FNNNTQeN7q5Wcxxta_ZXcMb83WV89_DbBQgAA",
  "scope": "openid aza ugs",
  "refresh_token": "1.AT0A7mRQZG6b20OdRv6Bplz96pjt2SlppDZFreL5gbw...zJvDu5lxfCySowg7wq1Ahk-f4nw7MHIBvkYnlwZmbiSu3SwoR",
  "client_id": "38aa3b87-a06d-4817-b275-7a316988d93b",
  "cert_token_use": "user_cert",
  "csr_type": "http://schemas.microsoft.com/windows/pki/2009/01/enrollment#PKCS10",
  "csr": "MIICXTCCAUUCAQAwGDEWMBQGA1UEAwwNdXNlckB1c2VyLmNvbTCCASIwDQ... NWzq9tz1FD1GyI0j0WjTAqdEN+gvqPGCC9L5dqOwOQqyhtEKvOr"
}
```

Primary Refresh Token (PRT)

Certificate Signing Request (CSR)

Signature

```
.XyZfNx3gsZ0o-T63h2UQv-lTNh-oXH2mu1O2yT19yvQ
```

JWT Signature signed by session key

2. Receive P2P certificate

- JSON Web Encryption encrypted by session key

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache
...

eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiY3R4Ijoid2JabzNXWXU5MDdWMTZsS29kck1uYnpvWjU4Rm5JYVoifQ..
Hiq1nYda2ld7uv8GrOvKNA.I4zh8JTQFh9ZZnPQ7KBiBb1X2O3pn9VXJ6wEd68QYD4qNxbY7-Es1X-fsvlwa2IH20By_3MD
yt0ov61G_v58ikyIe7Rh8-B8WS47ssTX8XQKo-eDc3HcxNNyKlvZRva076GD8b78NM9_AlIy4wOUu1J_K4yAk3Vj7BxQIwe
vmKV7_hLo40kFACr0ZL055QJN3IZEa4VRBv8r9B3jk1-lCEJKDDeGH64YS99lgZcHsSKLiqA1wkZSddlhQ36a3wL_..._Fx
1wI5XqoKOZ_vOFm4uvln8TY5LfFrcySCyGqvsVgdTaukzDEBClYAIPx3S2HF5Uoupx253CTKWXgrrg_kEORX9TtFxvoL849
SLys1ESjRWJGeQT2OAR87LQZETAIPR7iLR35buQ02t-Udpw45xFXmTgY5B56JmOImU7xPynEm-rAY9kb6Xp9xRa3ZKacl45
PA4MW5Ko16-oV_AIfFQO84jTNojcAJZbs4GRCVjCAQTPQIIx8FgIFX6bXXNmMle7kJFk8bEmYMtCNVFLfxdqHBBs-iuHWpR
WAJaYz30g2oBi6BdbEqPHMSIbMuKJE6hUZzxOex8JufL_z36VnA91sibktiWi3AhEG57Bvnv2l4IWw6zGPKjk3Xjk6SE4pV
Zvf7f7e8mGsJHp3Boi5uMbpNSnLbuyYxd-g8jXmJ0Y4WABnWKYCRQigyAkXxDW2ollqNM0lNax06JO-hiriM-JPl82spsEx
pQQVntdMd557FIB0_fwcoyeBUZvg9mjiXmggnS4WhoWWsah1fNs.
```

2. Receive P2P certificate

- Decrypted JSON response

User certificate

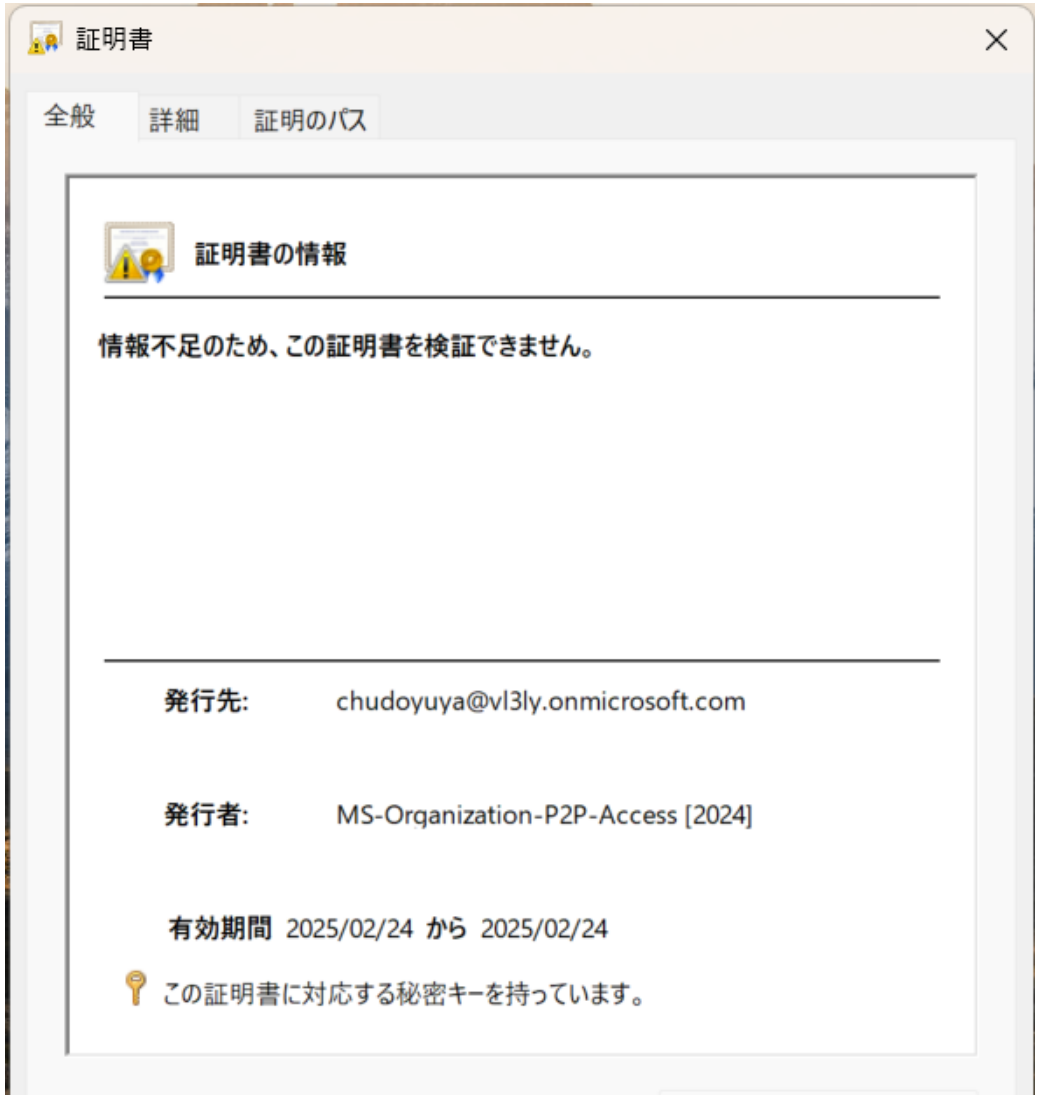

```
{
  'token_type': 'x509',
  'expires_in': '3599',
  'ext_expires_in': '0',
  'expires_on': '1745119064',
  'x5c': 'MIIEWzCCA0OgAwIBAgIRAPOu/8/dRjbtNt...bDdlg6rqC6H/WjaB7PazshrHJTlFy98EFnxtUY1vDsn',
  'cert_token_use': 'user_cert',
  'x5c_ca': 'MIIDPjCCAiagAwIBAgIJAMjSmUB+2H6rM...biD33/7PGfoY3bYuY1Q471cuOztyclw/BM5F9RFPdBMVznOCO+F1OT',
  'refresh_token': '1.AT0A7mRQZG6b20OdRv6Bplz96oc7qjhtoB..._-otZAQw7pmQ-ICpkG-QCHJpDeBiQEMhv5CAGK_yqtV_',
  'refresh_token_expires_in': 1209599,
  'resource': 'urn:p2p_cert',
  'id_token': 'eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lI... sInZlciI6IjEuMCIsInhtc19pZHJlbCI6IjEgOCJ9.'
}
```

2. Receive P2P certificate

P2P user certificate

- Issuer:

  MS-Organization-P2P-Access [2024]
- Subject:
  - Tenant ID
  - User SID
  - UserPrinciaplName (user@tenant.onmicrosoft.com)
- Validity: 1 hour



3. Negotiate authentication

- First, application layer uses **SPNEGO** (Simple and Protected GSS-API Negotiation Mechanism) to negotiate whether to use **NTLM** or **NEGOEX** as the authentication mechanism

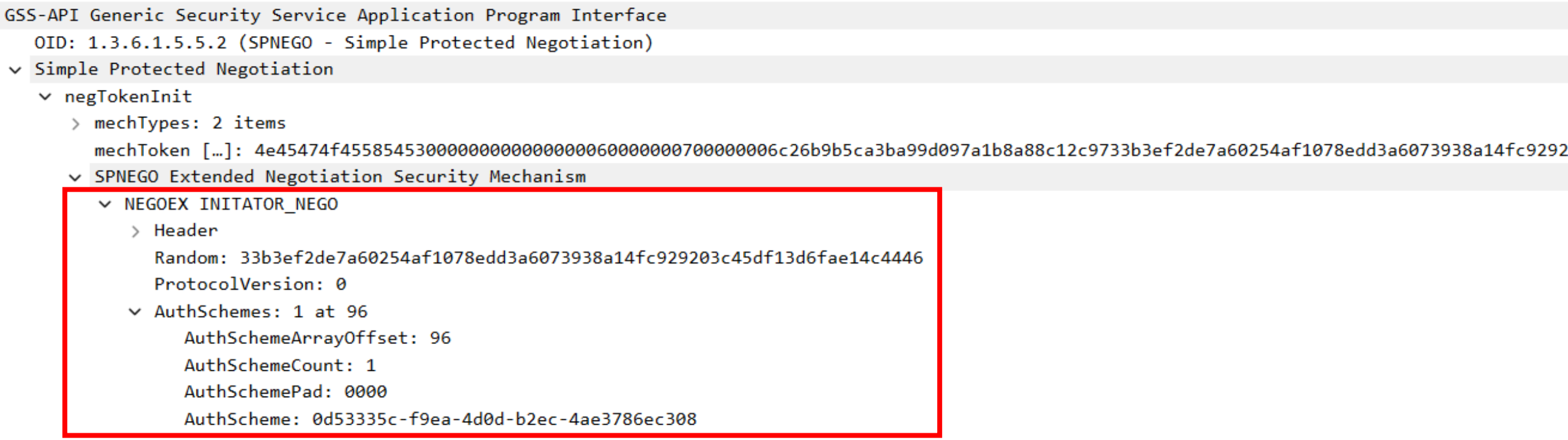
- In this case, **NEGOEX** (SPNEGO Extended Negotiation Security Mechanism) is selected for further communication

```
GSS-API Generic Security Service Application Program Interface
    OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
  ⌄ Simple Protected Negotiation
    ⌄ negTokenInit
      ⌄ mechTypes: 2 items
            MechType: 1.3.6.1.4.1.311.2.2.30 (NEGOEX - SPNEGO Extended Negotiation Security Mechanism)
            MechType: 1.3.6.1.4.1.311.2.2.10 (NTLMSSP - Microsoft NTLM Security Support Provider)
```

3. Negotiate authentication

- Then, NEGOEX messages are exchanged to negotiate an authentication protocol once again

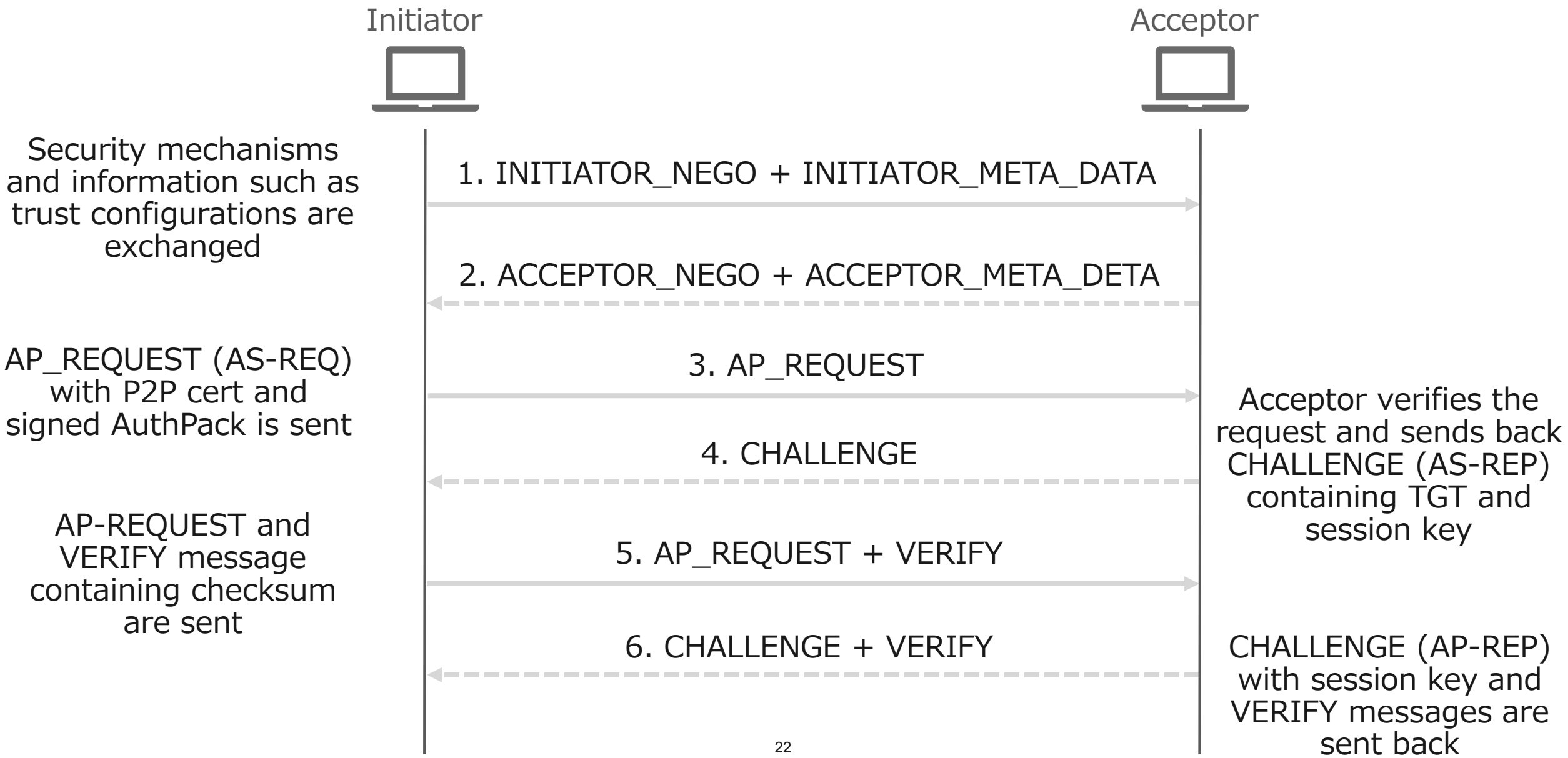
- **PKU2U** is selected and the authentication is carried out over NEGOEX messages.



4. PKU2U

- Kerberos messages are encapsulated in Exchange field of NEGOEX
- Acceptor acts as KDC and verifies authentication request
- realm is “WELLKNOWN:PKU2U”
- Username, “AzureAD¥¥username@tenant.onmicrosoft.com”, is specified in cname

4. PKU2U


Initiator
Acceptor
Security mechanisms and information such as trust configurations are exchanged
1. INITIATOR_NEGO + INITIATOR_META_DATA
2. ACCEPTOR_NEGO + ACCEPTOR_META_DETA
AP_REQUEST (AS-REQ) with P2P cert and signed AuthPack is sent
3. AP_REQUEST
Acceptor verifies the request and sends back CHALLENGE (AS-REP) containing TGT and session key
4. CHALLENGE
AP-REQUEST and VERIFY message containing checksum are sent
5. AP_REQUEST + VERIFY
6. CHALLENGE + VERIFY
CHALLENGE (AP-REP) with session key and VERIFY messages are sent back

5. Application access

- After the authentication, the application uses the session key sent by acceptor for further encryption if needed

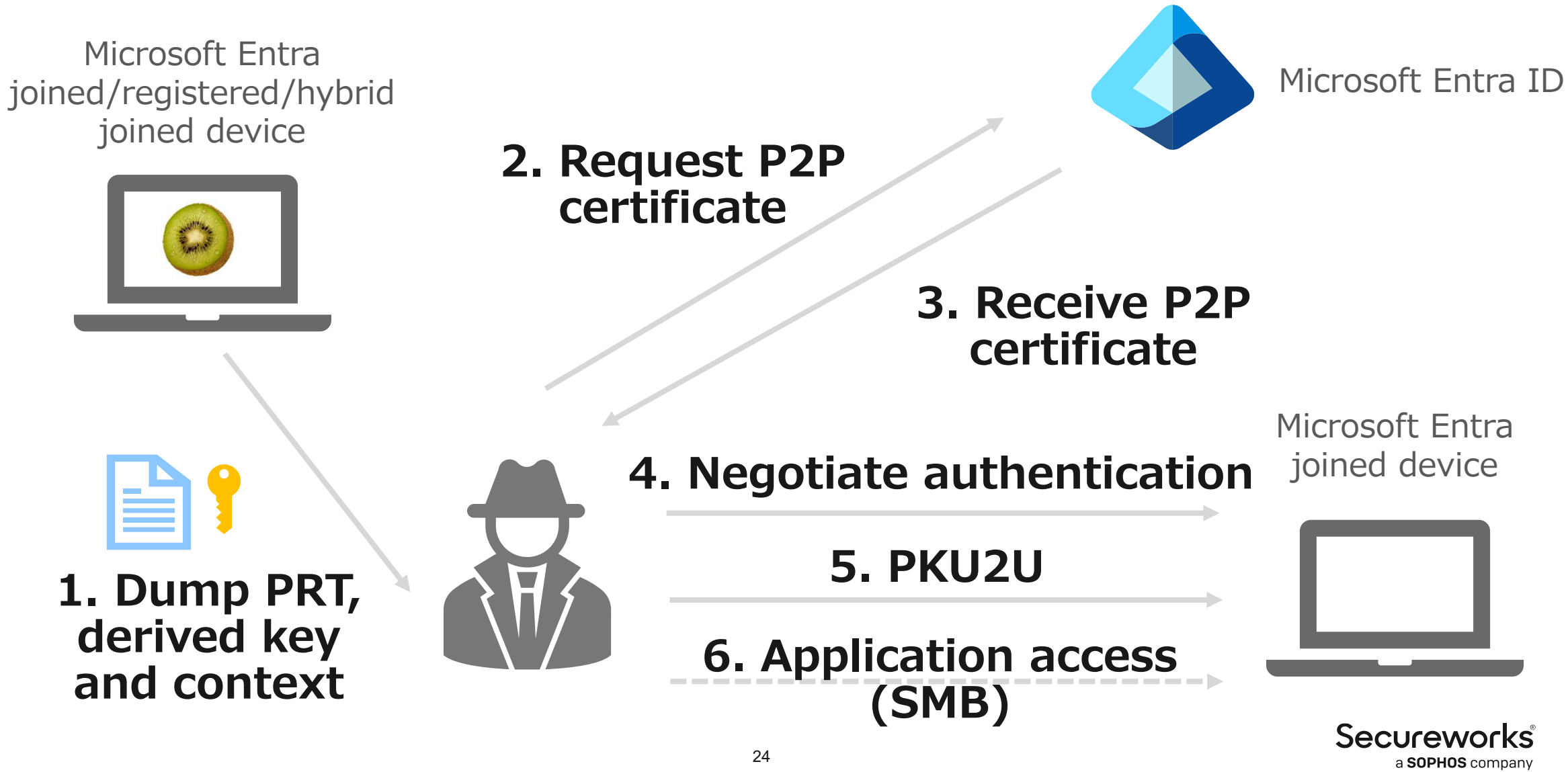
```
SMB2    Negotiate Protocol Response
SMB2    Negotiate Protocol Request
SMB2    Negotiate Protocol Response
SMB2    Session Setup Request, INITATOR_NEGO, INITIATOR_META_DATA, AP_REQUEST
SMB2    Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, ACCEPTOR_NEGO, ACCEPTOR_META_DATA, CHALLENGE
SMB2    Session Setup Request, AP_REQUEST, VERIFY
SMB2    Session Setup Response, CHALLENGE, VERIFY
SMB2    Tree Connect Request Tree: \\192.168.153.137\C$
SMB2    Tree Connect Response
SMB2    Create Request File:
SMB2    Create Response File:
SMB2    Find Request File:  SMB2_FIND_FULL_DIRECTORY_INFO Pattern: *
SMB2    Find Response
SMB2    Find Request File:  SMB2_FIND_FULL_DIRECTORY_INFO Pattern: *
SMB2    Find Response, Error: STATUS_NO_MORE_FILES
```

Negotiate authentication

PKU2U

Application Access

Classic Pass-the-Certificate



Weaponize Pass-the-Certificate

Challenges in classic Pass-the-Certificate

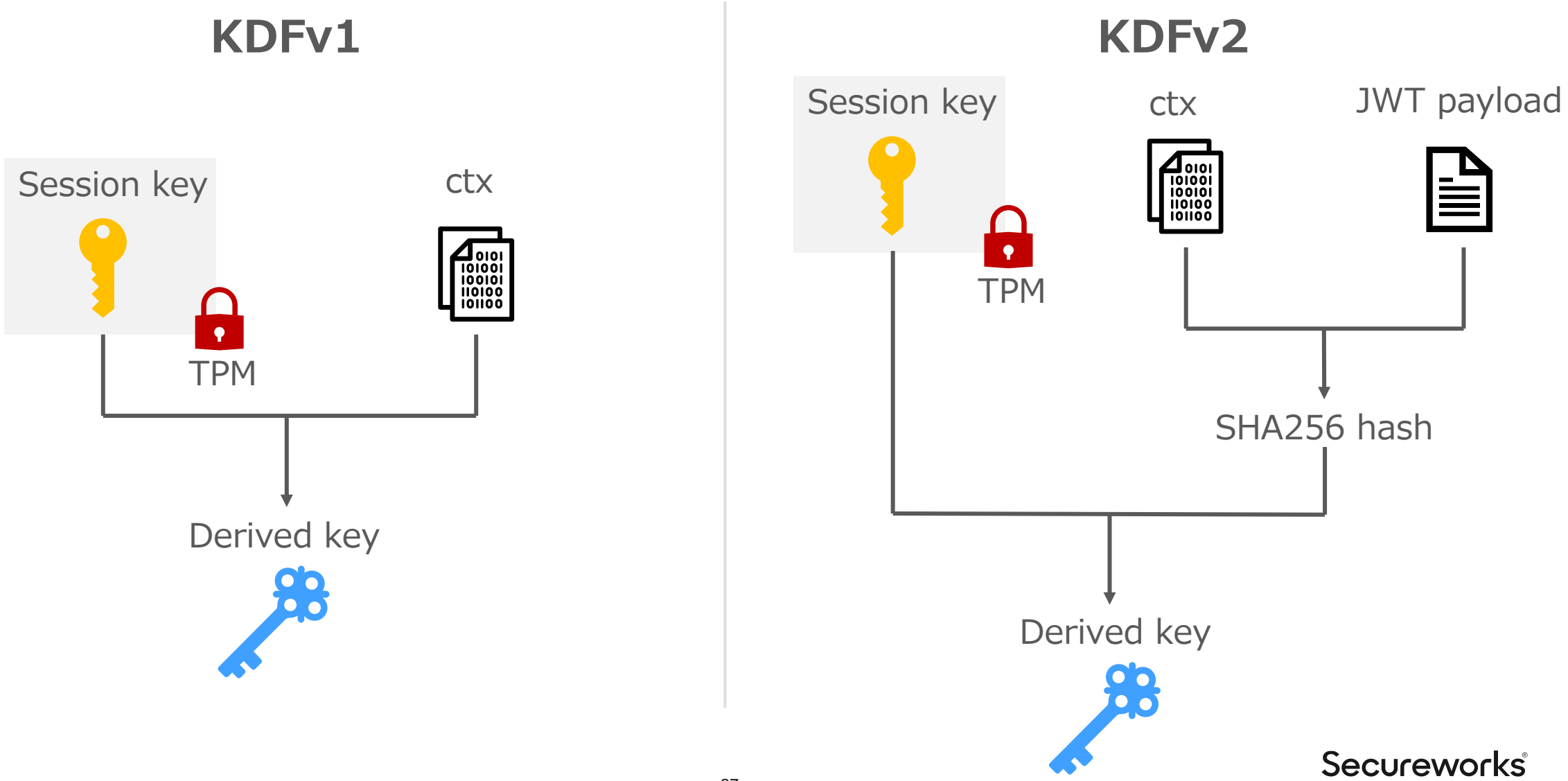
1. Upgrade P2P certificate request flow

- The way of deriving a key from session key to sign the JWT has changed since CVE-2021-33779
- Requesting P2P cert with the dumped PRT, derived key and context doesn’t work anymore

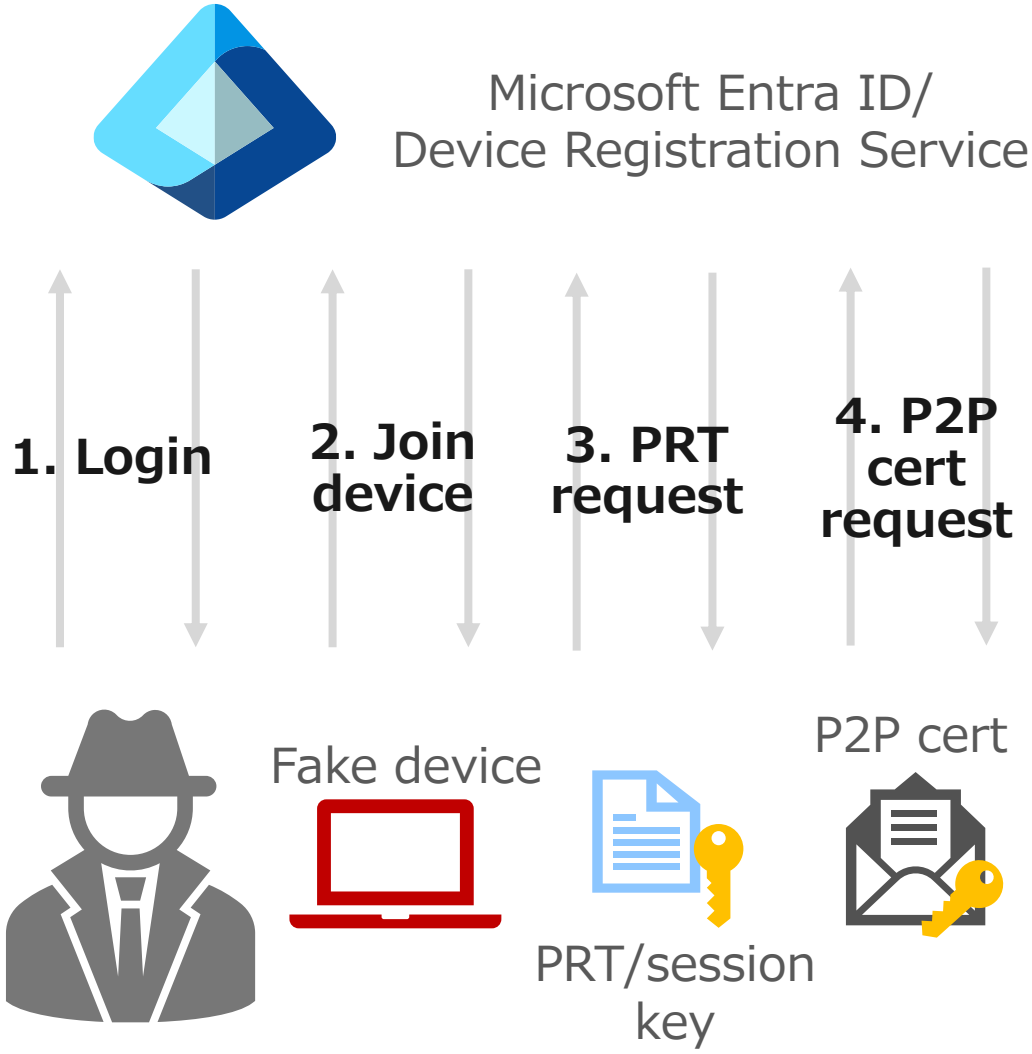
2. Multi protocol support

- Who opens SMB port in Entra joined devices?
- RDP or WinRM would be good targets

Upgrade P2P certificate request flow



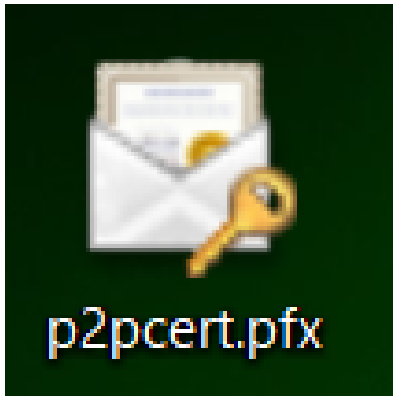
Upgrade P2P certificate request flow



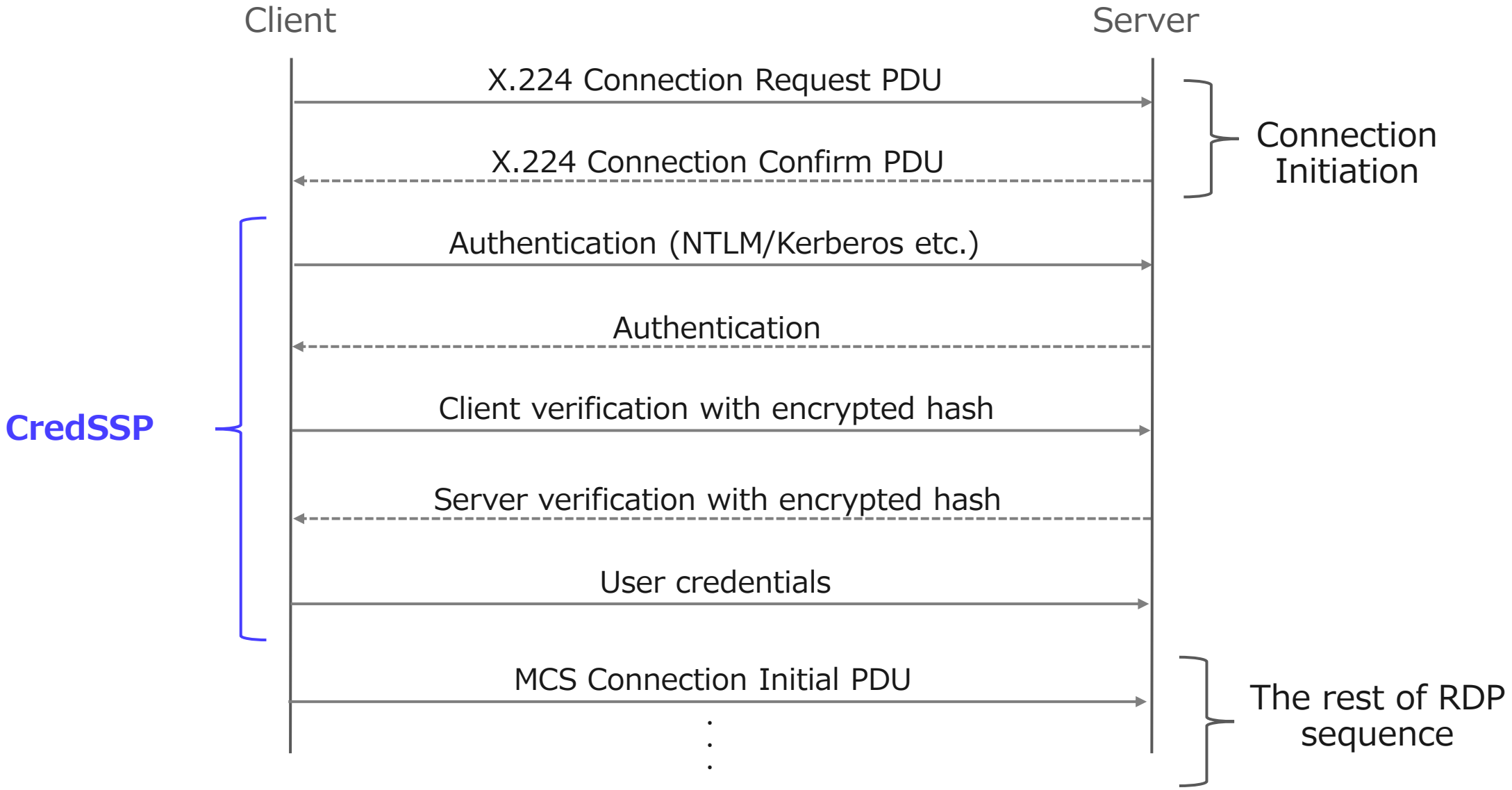
Upgrade P2P certificate request flow

Successfully implemented P2P certificate request function

```
[*] requesting P2P cert ...
[+] successfully acquired P2P cert!
[*] here is your p2p cert pfx : p2pcert.pfx
```



Multi protocol support
RDP



Multi protocol support
RDP

CredSSP Packet structure

```
Credential Security Support Provider
▾ TSRequest
    version: 6
  ▾ negoTokens: 1 item
    ▾ NegoData item
        negoToken […]: 6082017c06062b0601050502a08201703082016ca01a3018060a2b06010401823702021e060a2b0
      ▾ GSS-API Generic Security Service Application Program Interface
          OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
        ▾ Simple Protected Negotiation
          ▾ negTokenInit
            › mechTypes: 2 items
              mechToken […]: 4e45474f4558545300000000000000006000000070000005d2be37278303e6099e11d
            ▾ SPNEGO Extended Negotiation Security Mechanism
              › NEGOEX INITATOR_NEGO
              › NEGOEX INITIATOR_META_DATA
```

Multi protocol support
RDP

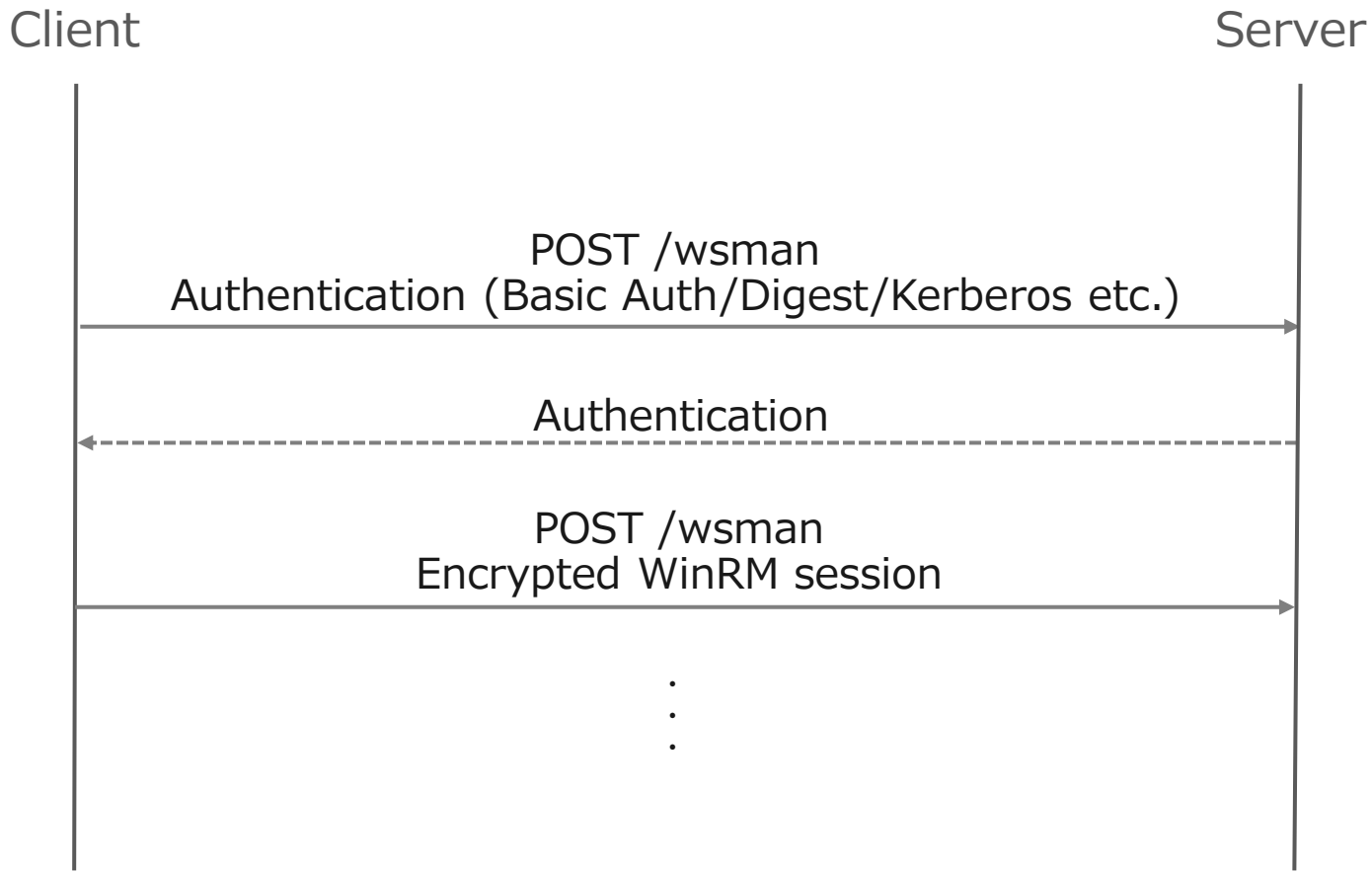
Source	Destination	Protocol	Length	Info
192.168.153.151	192.168.153.132	CredSSP	523	INITATOR_NEGO, INITIATOR_META_DATA
192.168.153.132	192.168.153.151	CredSSP	523	ACCEPTOR_NEGO, ACCEPTOR_META_DATA
192.168.153.151	192.168.153.132	CredSSP	2715	AP_REQUEST
192.168.153.132	192.168.153.151	CredSSP	467	CHALLENGE
192.168.153.151	192.168.153.132	CredSSP	1819	AP_REQUEST, VERIFY
192.168.153.132	192.168.153.151	CredSSP	475	CHALLENGE, VERIFY
192.168.153.151	192.168.153.132	CredSSP	251	
192.168.153.132	192.168.153.151	CredSSP	219	
192.168.153.151	192.168.153.132	CredSSP	315	
192.168.153.151	192.168.153.132	RDP	571	ClientData
192.168.153.132	192.168.153.151	RDP	235	ServerData Encryption: None (None)
192.168.153.151	192.168.153.132	RDP	555	ClientInfo
192.168.153.132	192.168.153.151	RDP	155	Error Alert
192.168.153.132	192.168.153.151	RDP	155	MultiTransportRequest
192.168.153.151	192.168.153.132	RDP	139	MultiTransport response
192.168.153.132	192.168.153.151	RDP	587	Demand Active PDU
192.168.153.151	192.168.153.132	RDP	731	Confirm Active PDU
192.168.153.151	192.168.153.132	RDP	155	RDP PDU Type: Synchronize

PKU2U

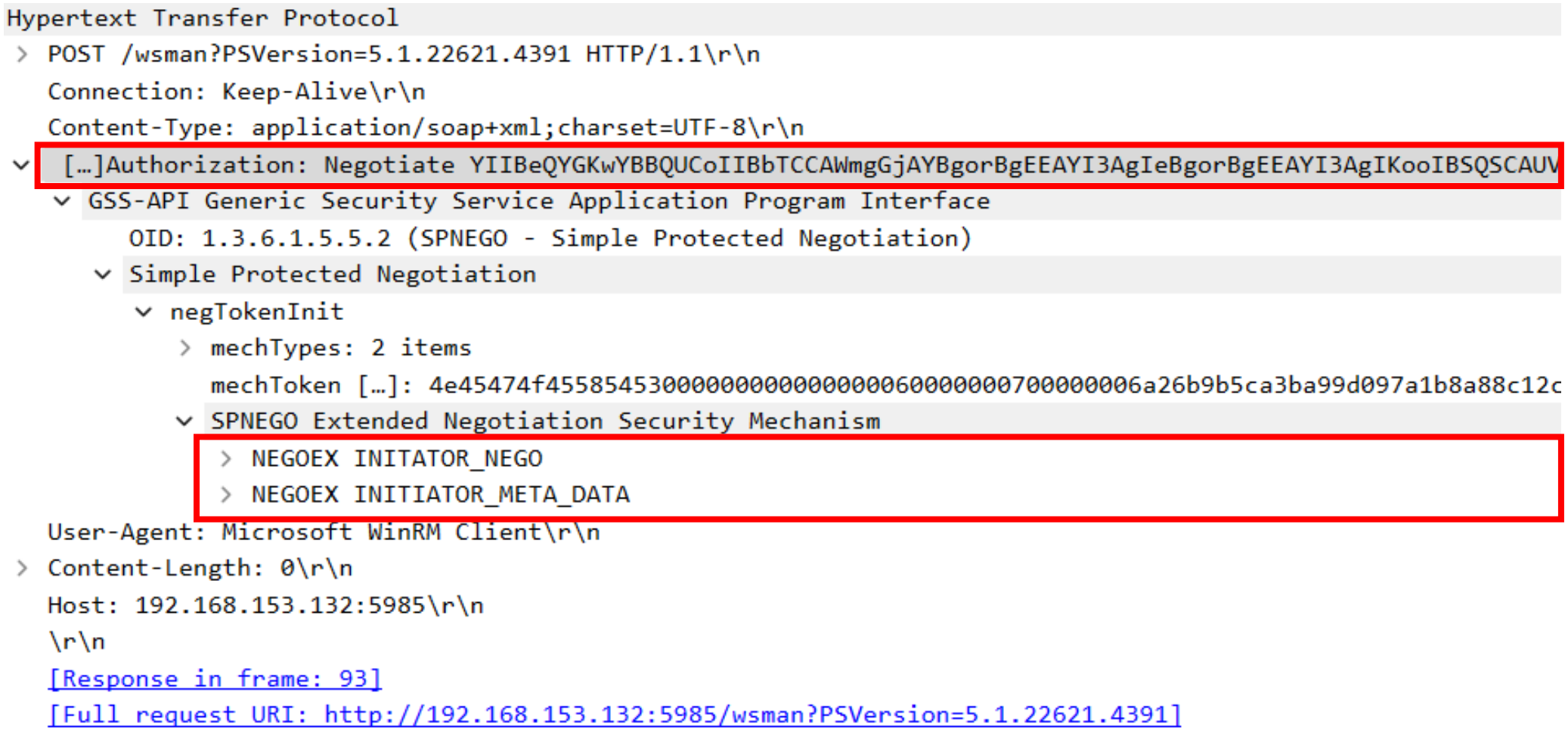
Client-server verification

Credential delegation

Multi protocol support
WinRM (HTTP)



Multi protocol support
WinRM (HTTP)



Multi protocol support
WinRM (HTTP)

Source	Destination	Protocol	Info
192.168.153.137	192.168.153.132	HTTP	POST /wsman?PSVersion=5.1.22621.4391 HTTP/1.1 , INITATOR_NEGO, INITIATOR_META_DATA
192.168.153.132	192.168.153.137	HTTP	HTTP/1.1 401 , ACCEPTOR_NEGO, ACCEPTOR_META_DATA
192.168.153.137	192.168.153.132	HTTP	POST /wsman?PSVersion=5.1.22621.4391 HTTP/1.1 , AP_REQUEST, VERIFY
192.168.153.132	192.168.153.137	HTTP	HTTP/1.1 200 , CHALLENGE, VERIFY
192.168.153.137	192.168.153.132	HTTP	POST /wsman?PSVersion=5.1.22621.4391 HTTP/1.1 (application/http-spnego-session-encrypted)
192.168.153.132	192.168.153.137	HTTP	HTTP/1.1 200 (application/http-spnego-session-encrypted)
192.168.153.137	192.168.153.132	HTTP	POST /wsman?PSVersion=5.1.22621.4391 HTTP/1.1 (application/http-spnego-session-encrypted)
192.168.153.132	192.168.153.137	HTTP	HTTP/1.1 200 (application/http-spnego-session-encrypted)
192.168.153.137	192.168.153.132	HTTP	POST /wsman?PSVersion=5.1.22621.4391 HTTP/1.1 (application/http-spnego-session-encrypted)
192.168.153.132	192.168.153.137	HTTP	HTTP/1.1 200 (application/http-spnego-session-encrypted)
192.168.153.137	192.168.153.132	HTTP	POST /wsman?PSVersion=5.1.22621.4391 HTTP/1.1 (application/http-spnego-session-encrypted)
192.168.153.132	192.168.153.137	HTTP	HTTP/1.1 200 (application/http-spnego-session-encrypted)

Multi protocol support
Extra mile: RPC

```
Info
Bind: call_id: 2, Fragment: Single, 3 context items: EPMv4 V3.0 (32bit NDR), EPMv4 V3.0 (64bit NDR), EPMv4 V3.0 (6cb71c2c-9812-4540-0300-000000000000)
Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5840 max_recv: 5840, 3 results: Provider rejection, Acceptance, Negotiate ACK
Map request, SVCCTL, 32bit NDR
Map response, SVCCTL, 32bit NDR
Bind: call_id: 2, Fragment: Single, 2 context items: SVCCTL V2.0 (32bit NDR), SVCCTL V2.0 (6cb71c2c-9812-4540-0300-000000000000), INITATOR_NEGO, INITIATOR_META_DATA
Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5840 max_recv: 5840, 2 results: Acceptance, Negotiate ACK, ACCEPTOR_NEGO, ACCEPTOR_META_DATA
Alter_context: call_id: 2, Fragment: Single, 1 context items: SVCCTL V2.0 (32bit NDR), AP_REQUEST, VERIFY
Alter_context_resp: call_id: 2, Fragment: Single, max_xmit: 5840 max_recv: 5840, 1 results: Acceptance, CHALLENGE, VERIFY
Alter_context: call_id: 2, Fragment: Single, 1 context items: SVCCTL V2.0 (32bit NDR), AP_REQUEST
Alter_context_resp: call_id: 2, Fragment: Single, max_xmit: 5840 max_recv: 5840, 1 results: Acceptance
OpenSCManager2 request
OpenSCManager2 response
```

EntraPassTheCert

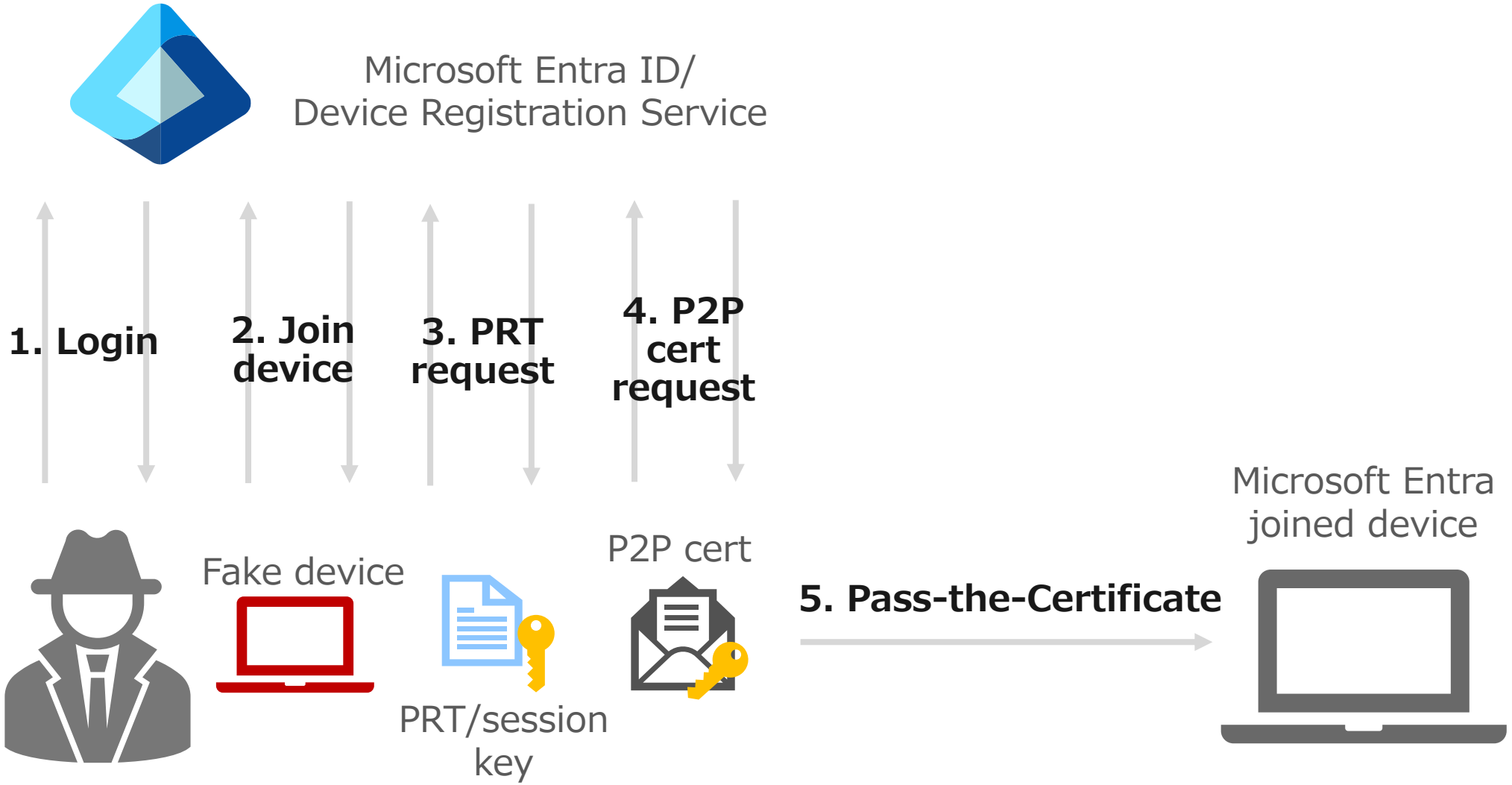
Allows attackers to request P2P certificate and connect to Microsoft Entra joined devices through SMB, RPC, RDP and WinRM (https://github.com/temp43487580/EntraPassTheCert)

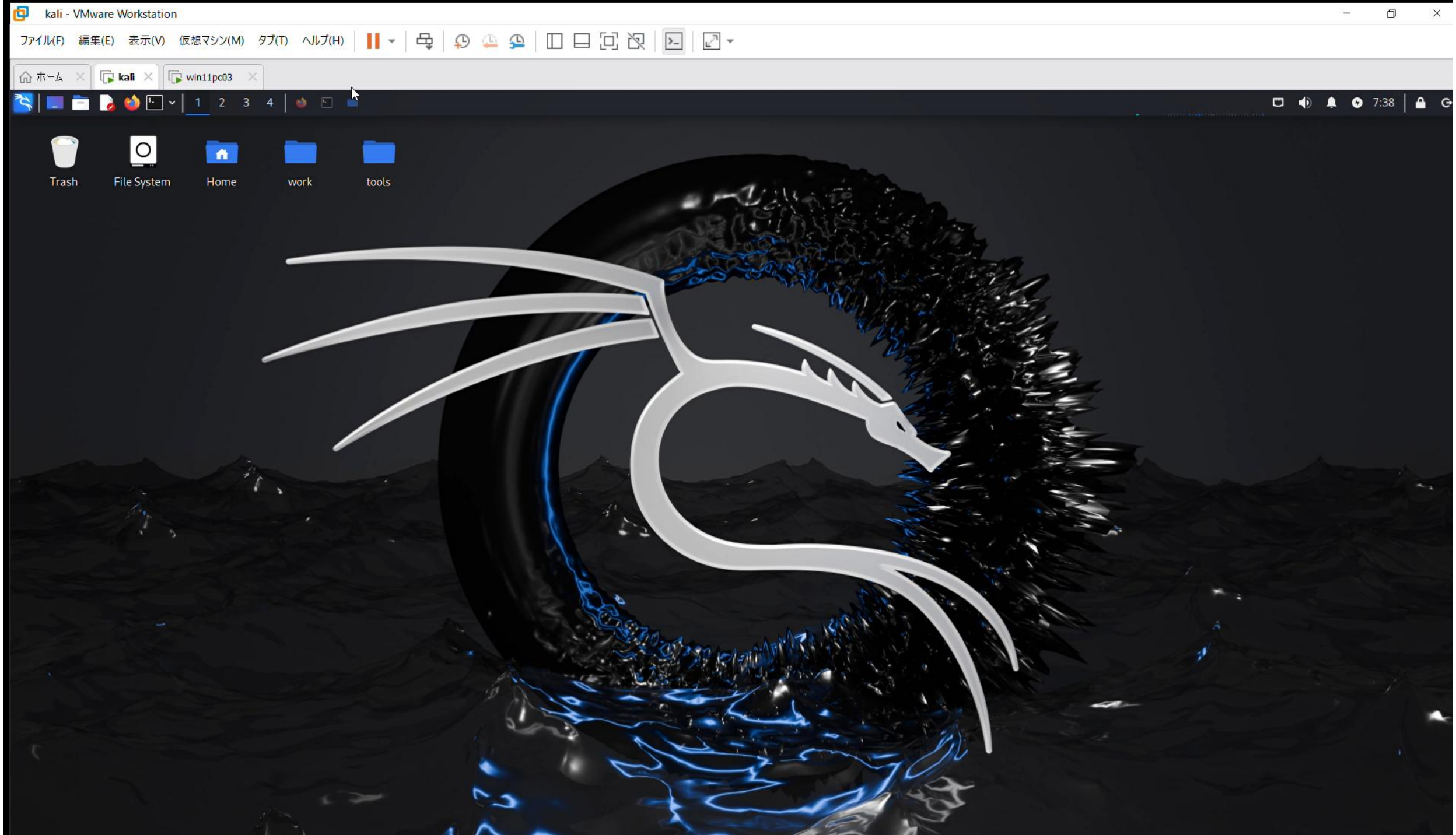
```
usage: entraptc.py [-h] {request_p2pcert,smb,rdp,winrm,rpc} ...

post-exploitation tool for requesting p2p cert and authenticate with it

positional arguments:
  {request_p2pcert,smb,rdp,winrm,rpc}
                        Available commands
    request_p2pcert     request P2P cert with PRT and SessionKey
    smb                 SMB to Entra joinned machine with P2P cert
    rdp                 RDP to Entra joinned machine with P2P cert
    winrm               WinRM to Entra joinned machine with P2P cert
    rpc                 RPC to Entra joinned machine with P2P cert

options:
  -h, --help            show this help message and exit
```

Attack scenarios

1. Join a fake device
2. Phishing for P2P certificate
3. Living off the Land Key

Join a fake device




kali - VMware Workstation
ファイル(F) 編集(E) 表示(V) 仮想マシン(M) タブ(T) ヘルプ(H)
ホーム
kali
win11pc03
1 2 3 4
7:38
Trash
File System
Home
work
tools

Join a fake device

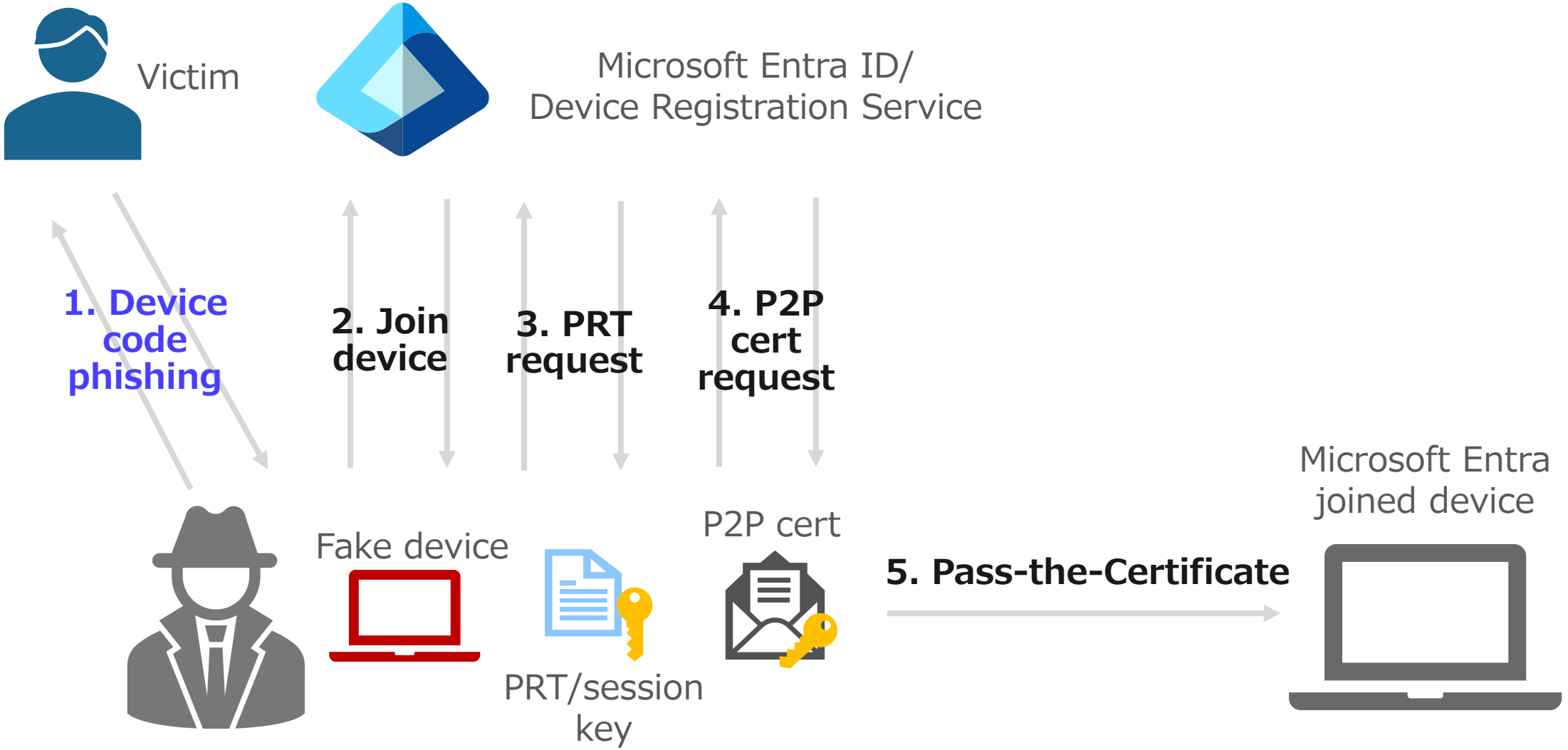
Pros

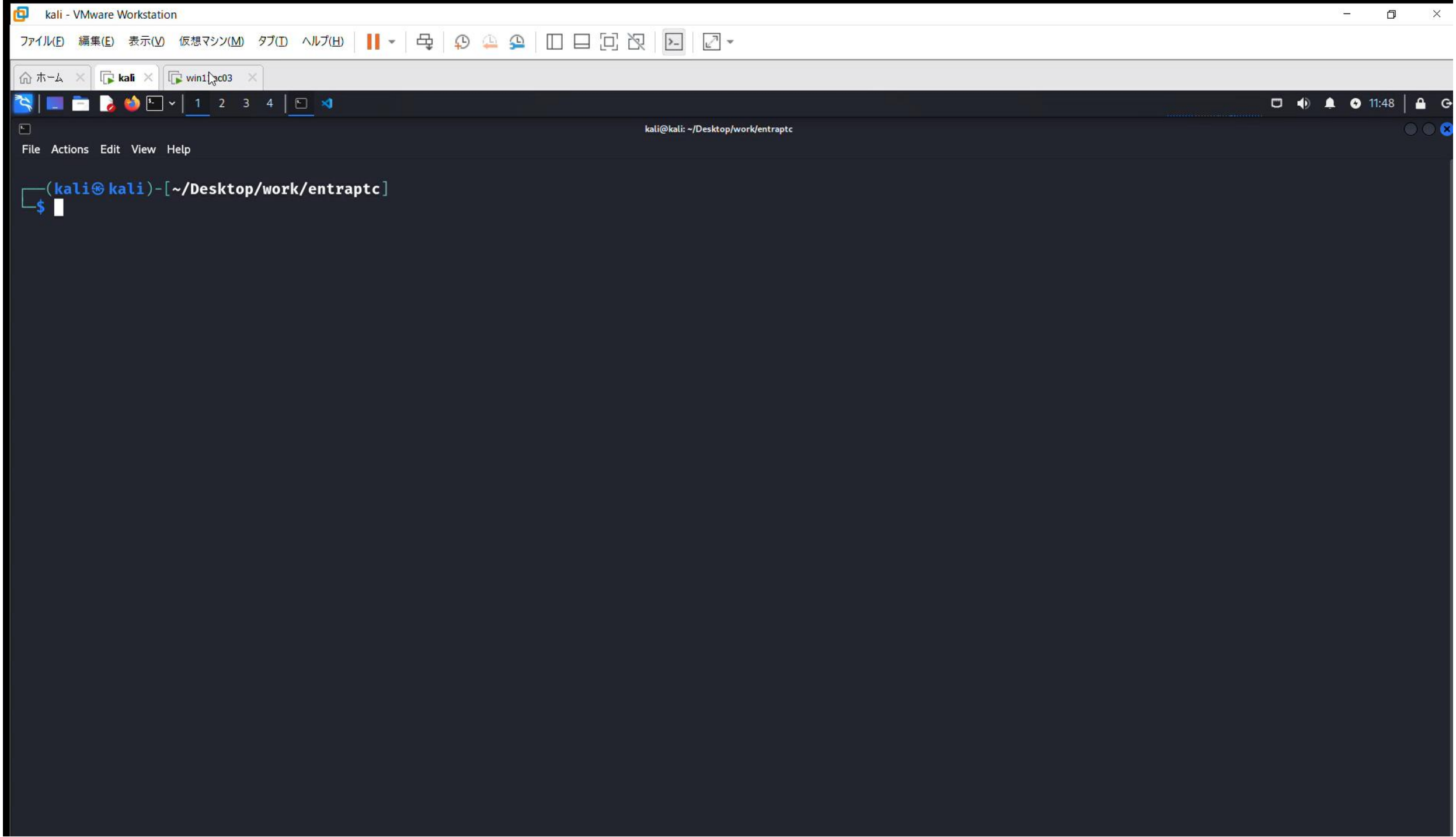
- MFA is not enforced if Conditional Access is not properly configured (more on this later)

Cons

- Need to find credentials

Phishing for P2P cert




kali - VMware Workstation
ファイル(F) 編集(E) 表示(V) 仮想マシン(M) タブ(T) ヘルプ(H)
ホーム
kali
win1 c03
1 2 3 4
11:48
kali@kali: ~/Desktop/work/entraptc
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/work/entraptc]
$

Phishing for P2P cert

Pros

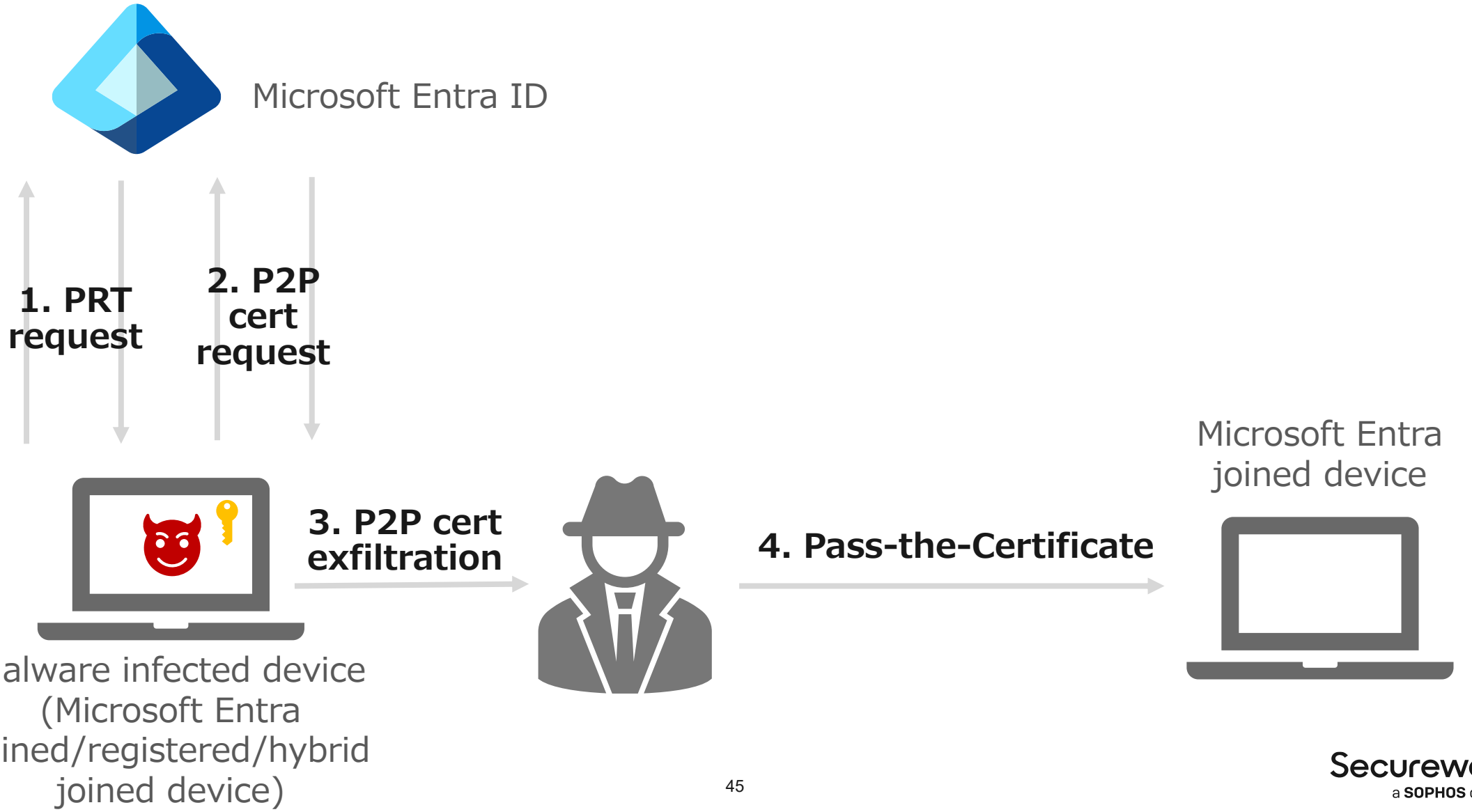
- No need for hunting credentials
- Authentication succeeds even when MFA is properly configured

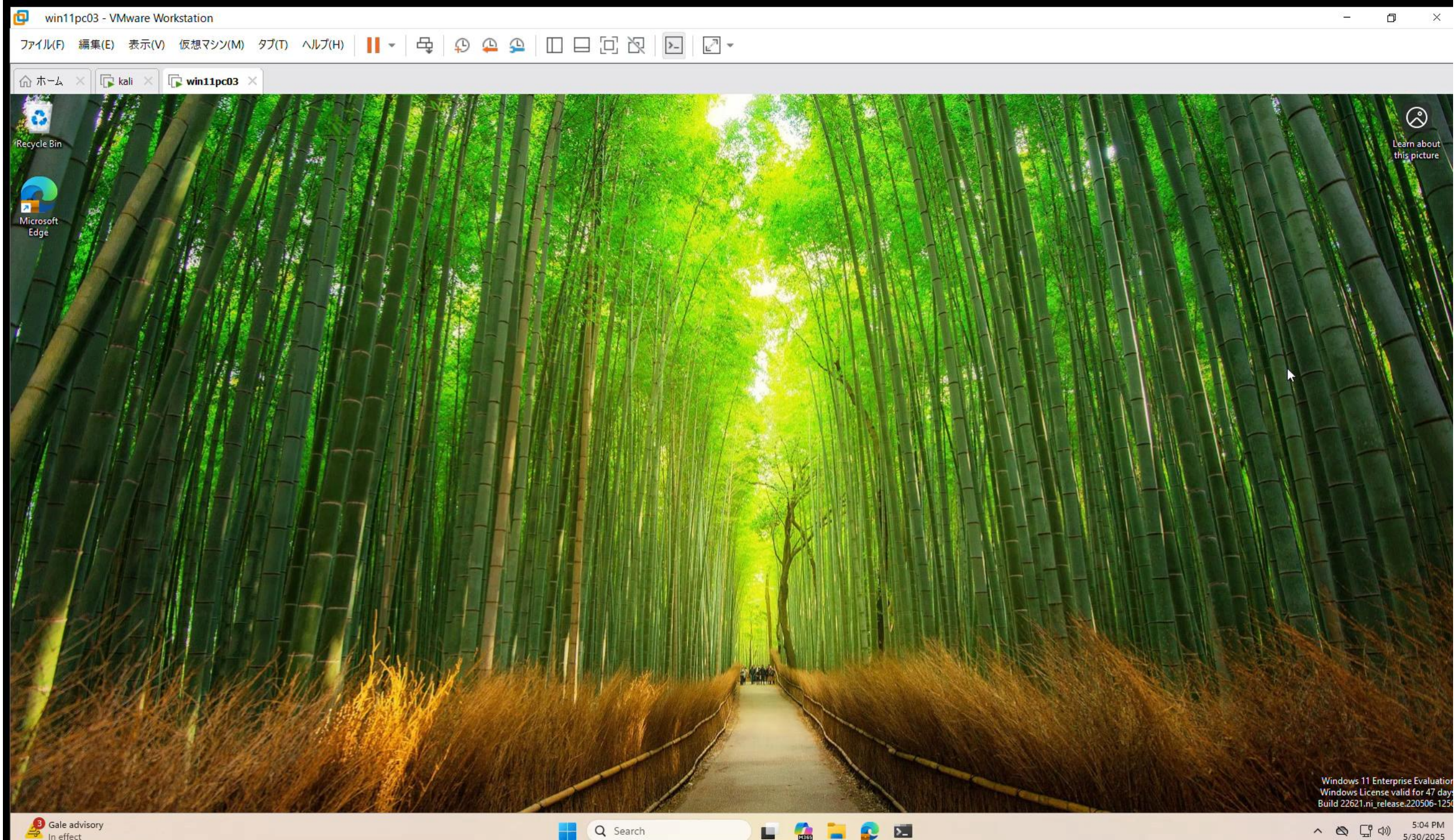
Cons

- Social engineering is necessary
- Device code authentication is coming to an end?

  (because we had so much fun)

Living off the Land Key




win11pc03 - VMware Workstation
ファイル(F) 編集(E) 表示(V) 仮想マシン(M) タブ(T) ヘルプ(H)
ホーム
kali
win11pc03
Recycle Bin
Microsoft Edge
Learn about this picture
Windows 11 Enterprise Evaluation
Windows License valid for 47 days
Build 22621.ni_release.220506-125
Gale advisory
In effect
Search
5:04 PM
5/30/2025

Living off the Land Key

Pros

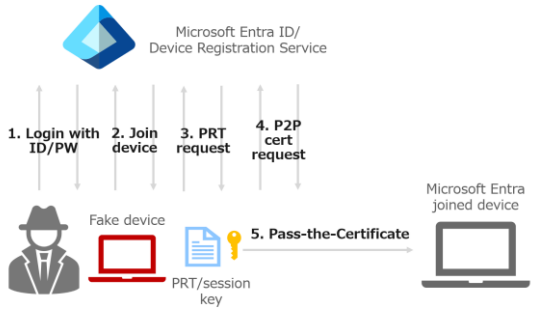
- Effective when the other attack scenarios are blocked by MFA and device code auth

Cons

- Malware infection is necessary
- Need to find credentials

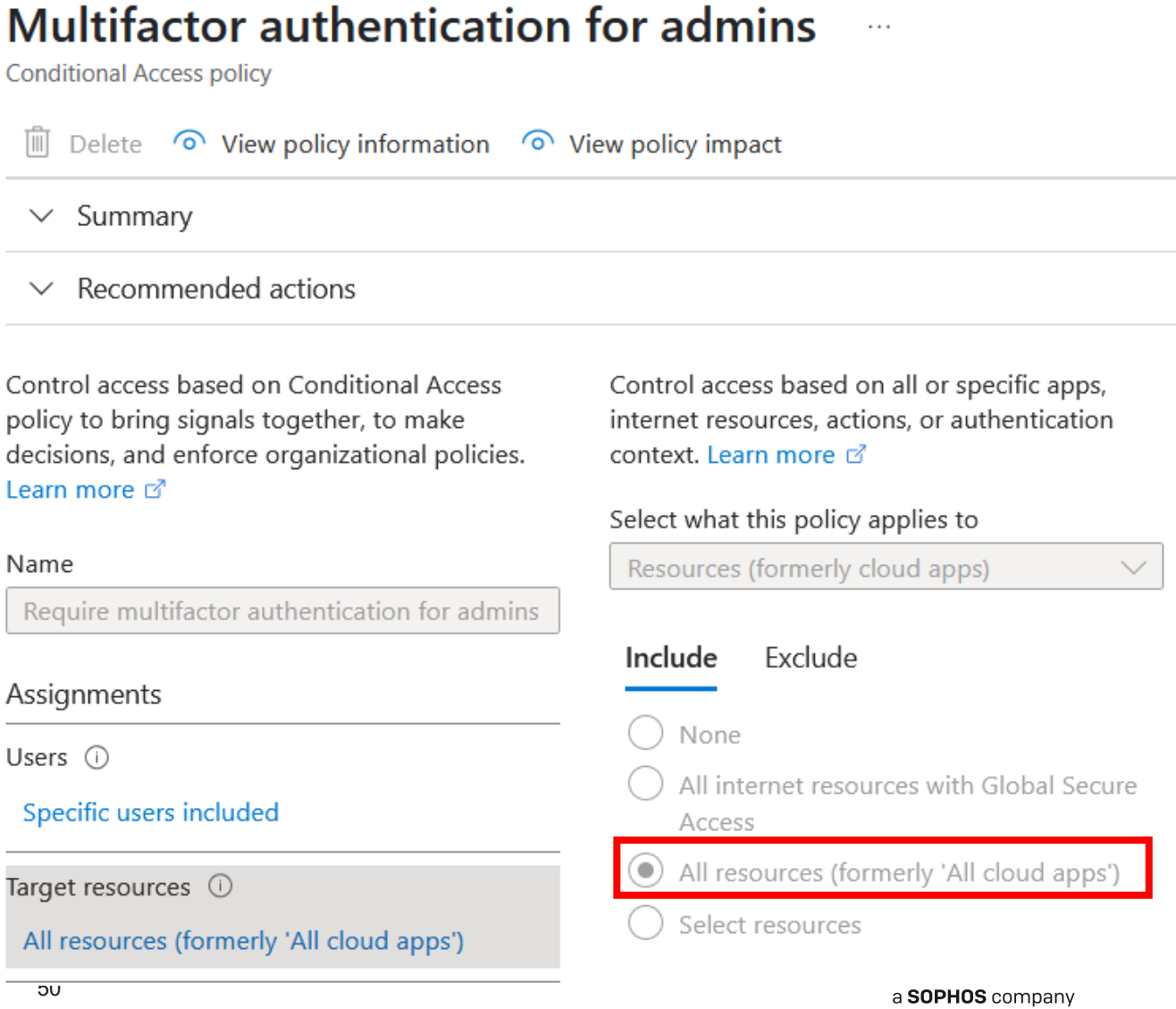
Prevent & Detect Pass-the-Certificate

Prevent

Attack Scenario	Strategies
Join a fake device 	Require MFA for device registration
Phishing for P2P cert 	Restrict device code flow
Living off the Land Key 	Implement network access control

Prevent

- Requiring MFA for **all resources** in Conditional Access is not perfect, except for Azure VM



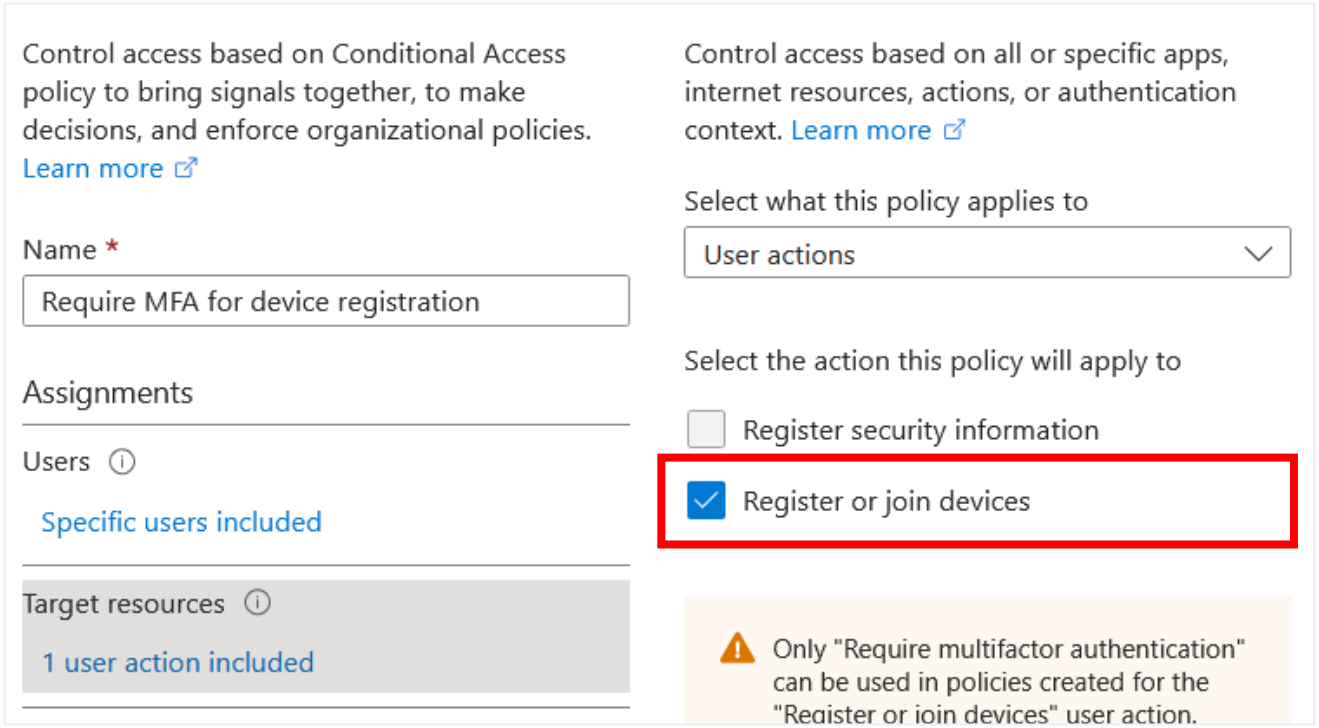
Prevent

- MFA is not required when acquiring tokens for device registration service

```
┌──(kali㉿kali)-[~]
└─$ roadtx gettokens -u globaladmin@vl4ly.onmicrosoft.com -p $PASSWORD -c 29d9ed98-a469-4536-ade2-f981bc1d605e -r msgraph
Requesting token for resource https://graph.microsoft.com/
Error during authentication: AADSTS50076: Due to a configuration change made by your administrator, or because you moved to a new location, you must use multi-factor authentication to access '00000003-0000-0000-c000-000000000000'. Trace ID: 28758381-744b-4494-b7c4-6363ae0a0300 Correlation ID: 405e5af0-378f-48ed-a0ec-e5ab8cc2d430 Timestamp: 2025-05-10 00:33:11Z

┌──(kali㉿kali)-[~]
└─$ roadtx gettokens -u globaladmin@vl4ly.onmicrosoft.com -p $PASSWORD -c 29d9ed98-a469-4536-ade2-f981bc1d605e -r urn:ms-drs:enterpriseregistration.windows.net
Requesting token for resource urn:ms-drs:enterpriseregistration.windows.net
Tokens were written to .roadtools_auth
```

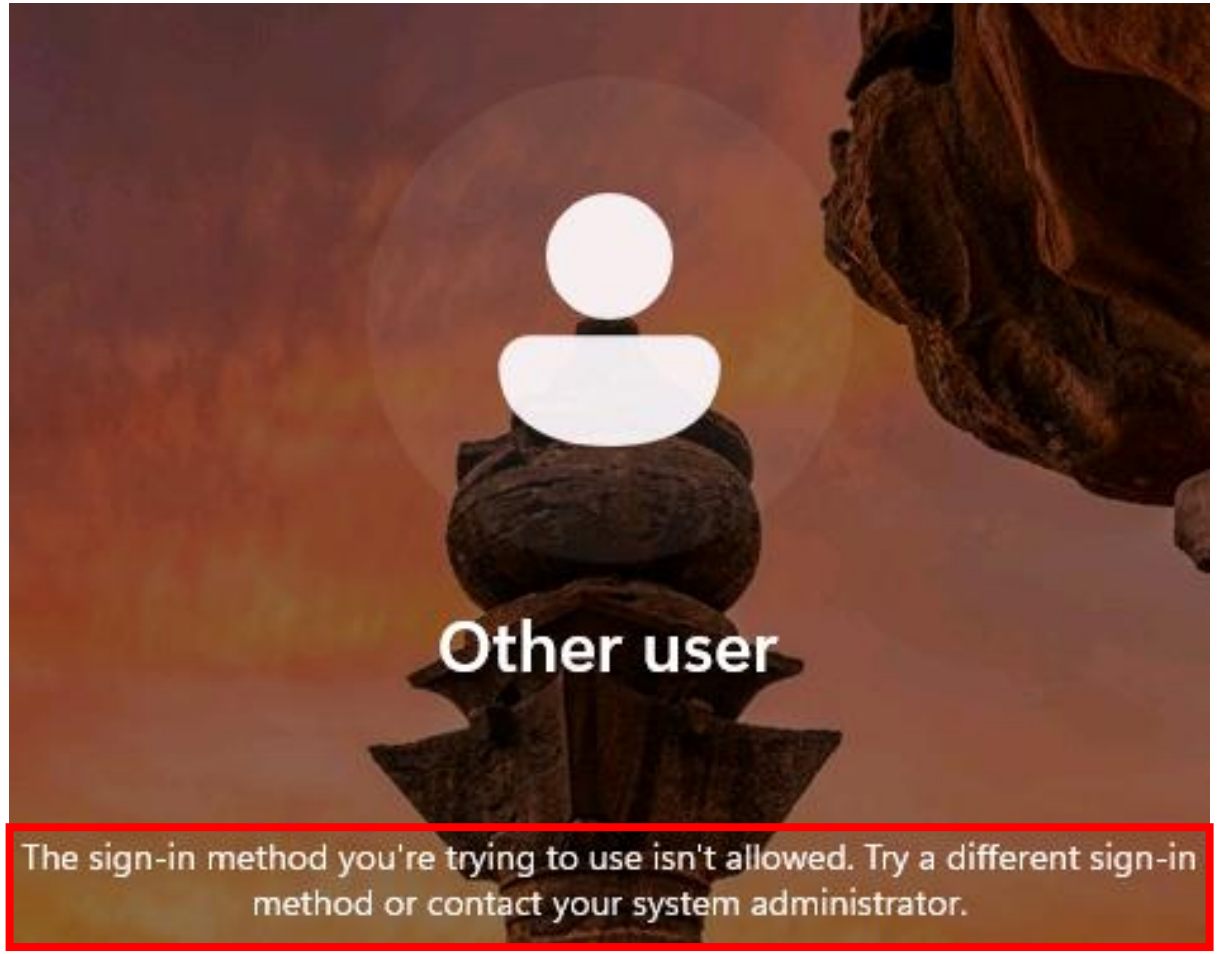
Prevent

- Require MFA for device registration is necessary
  - Block device code auth and implement network access control as well



Prevent

- Just requiring MFA for all resources works **for Azure VMs** which are joined to Entra through “Login with Entra ID” option



Detect

- Look for logon events (ID:4624) authenticated via PKU2U, as Mor Rubin recommended
- Watch suspicious token request logs for device registration in Sign-in logs or device registration activity in Audit logs

Conclusion

Conclusion

- Pass-the-Certificate helps attackers move laterally across Microsoft Entra-based environment with only credentials
- Reviewing Conditional Access policies and network segmentation is recommended

Secureworks
a SOPHOS company

@TEMP43487580

@優也-中堂-2601a596

EntraPassTheCert

BAADTokenBroker

Reference

- https://github.com/dirkjanm/ROADtools
- https://github.com/Gerenios/AADInternals
- https://github.com/diyan/pywinrm
- https://github.com/skelsec/aardwolf
- https://manage-the.cloud/2023/06/02/windows-remote-management-winrm-on-azure-ad-joined-devices/
- https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-negoex/0ad7a003-ab56-4839-a204-b555ca6759a2