AAD Joined Machines - The New Lateral Movement

Mor Rubin

Who am I?

- Mor Rubin (@rubin_mor)
- Senior security researcher at Microsoft
- Interested in networking, cloud and On-Prem attacks, mitigations and detections

Microsoft

Agenda

- Introduction to key terms
- NegoEx protocol
- Attacks
- Demo
- Hunting
- Takeaways

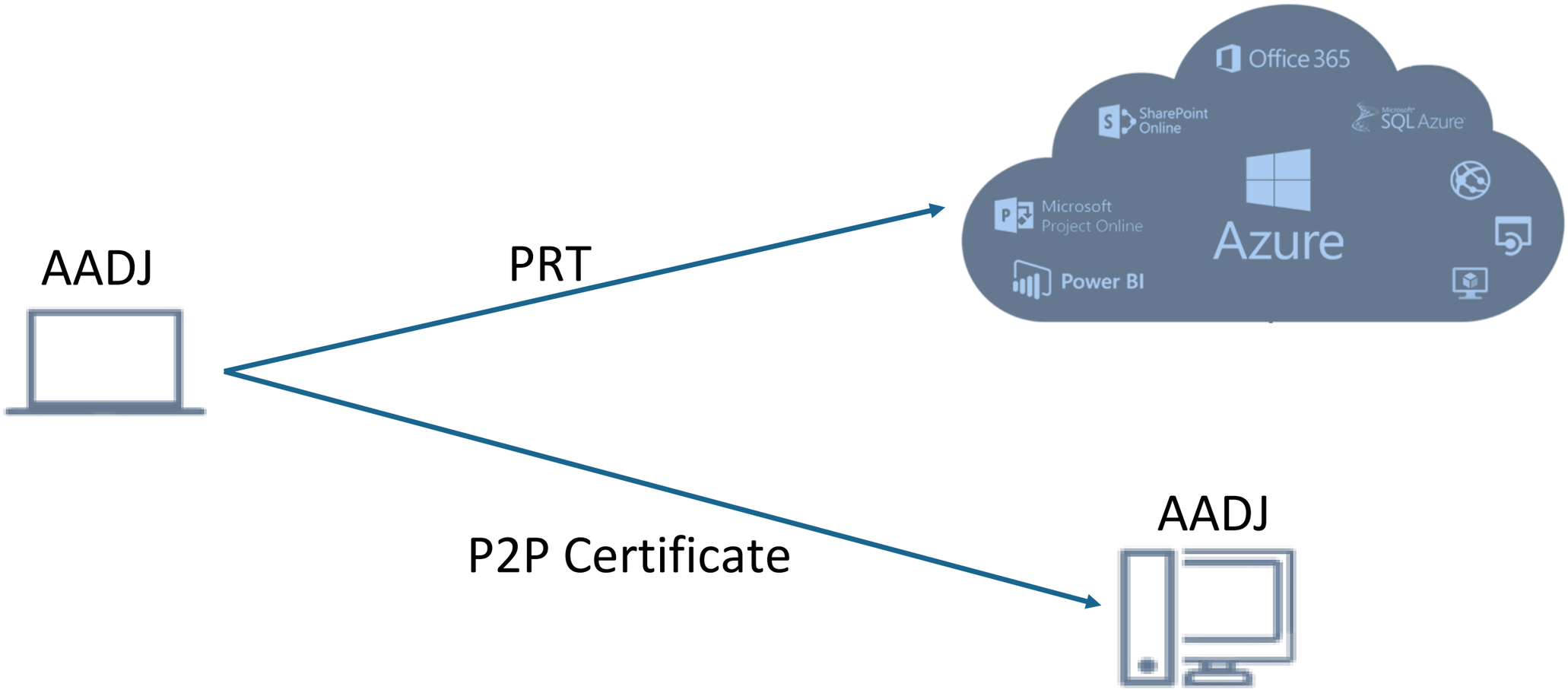
Technical background

Azure AD Joined device

(Windows Server) Active Directory	Azure Active Directory
LDAP	REST API's
NTLM/Kerberos	OAuth/SAML/OpenID/etc
Structured directory (OU tree)	Flat structure
GPO's	No GPO's
Super fine-tuned access controls	Predefined roles
Domain/forest	Tenant
Trusts	Guests

Ref: Dirk-Jan Mollema "I'm in your cloud… reading everyone's email"
Troopers '19 https://dirkjanm.io/assets/raw/TR19-Im%20in%20your%20cloud.pdf

AADJ Authenticated Connections

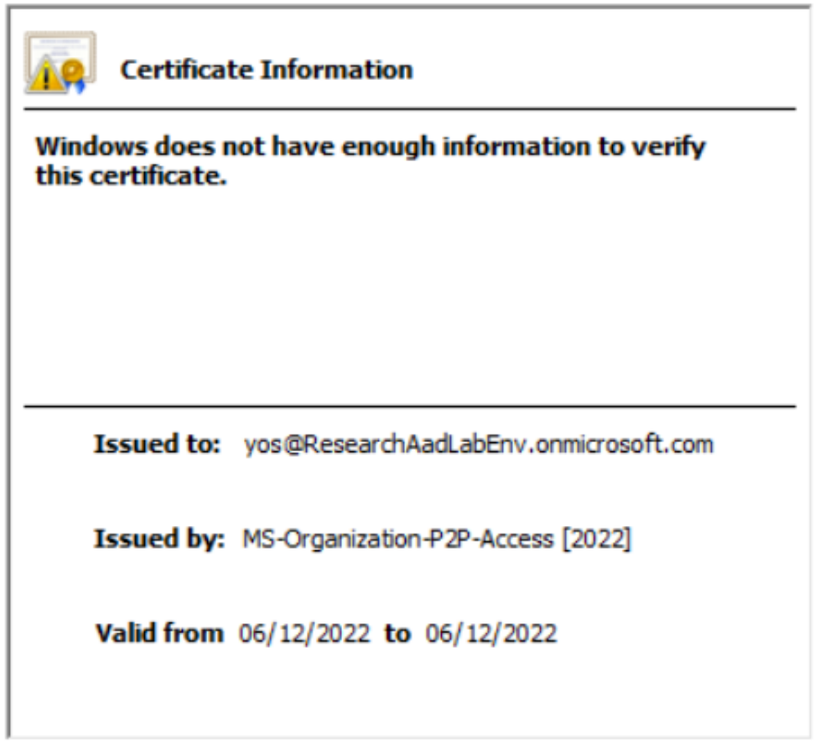


Primary Refresh Token - PRT

- A JSON Web Token (JWT) for the user and the device it was issued for
- Can be compared to Ticket Granting Ticket (TGT)
- Can be used to authenticate to any application

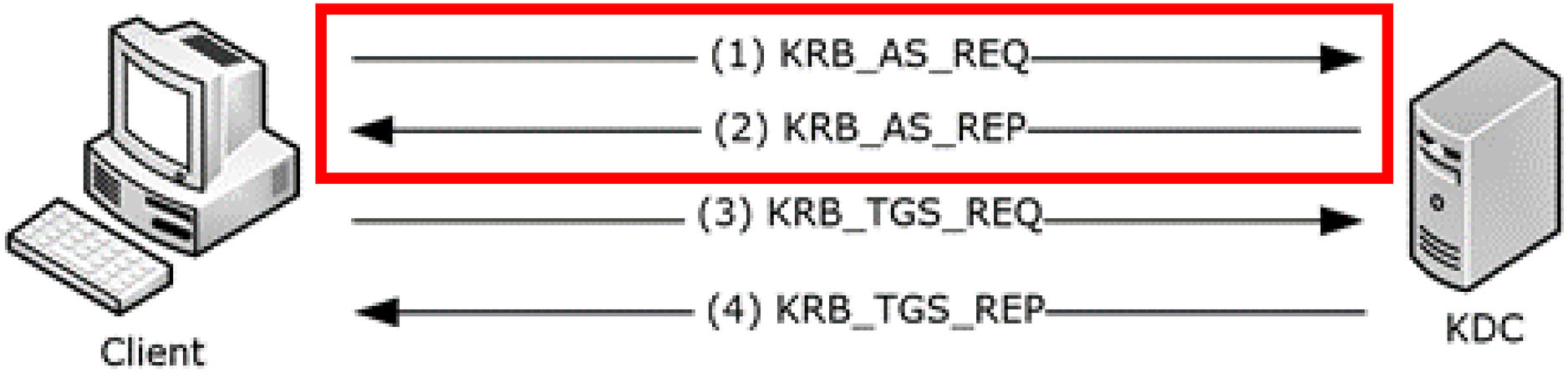
P2P Azure AD certificate

- Certificate that is used for peer-to-peer authentication between Azure AD joined devices
- Issued by Azure AD upon request and valid only for 1 hour



Kerberos PKINIT

Kerberos extension allows certificate authentication over Kerberos instead of hash (password)



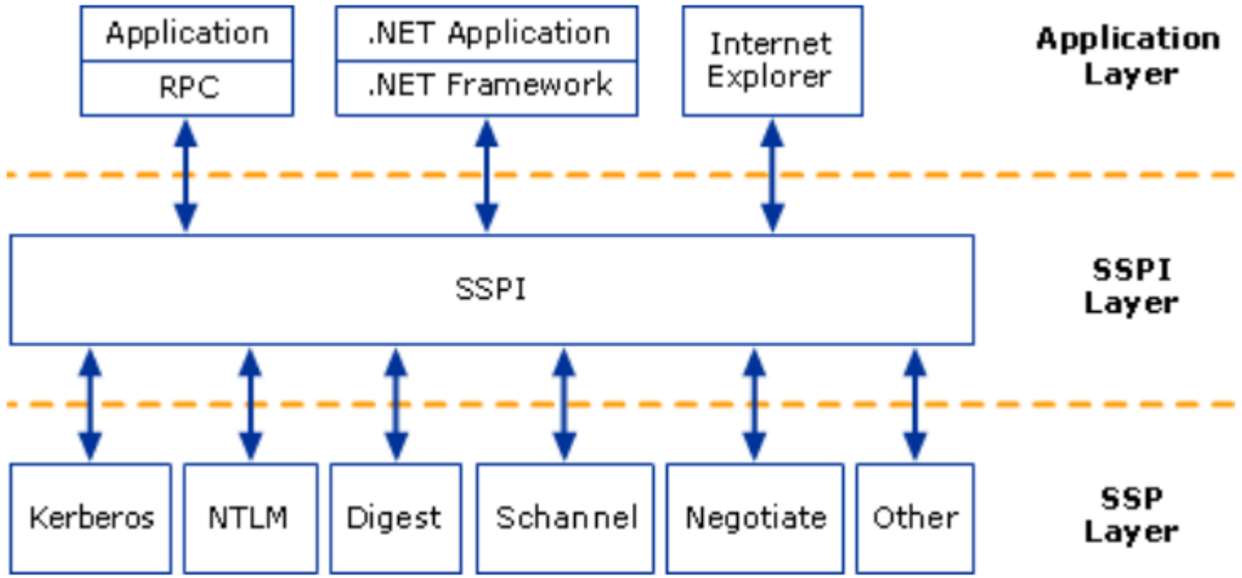
Ref: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/b4af186e-b2ff-43f9-b18e-eedb366abf13

PKU2U

- Based on Kerberos version 5 messages and the Kerberos GSS-API mechanism.
- Implemented as a Security Support Provider (SSP) enables peer-to-peer authentication
- Used for authentication in NegoEx when no KDC exists

GSSAPI \ SSPI

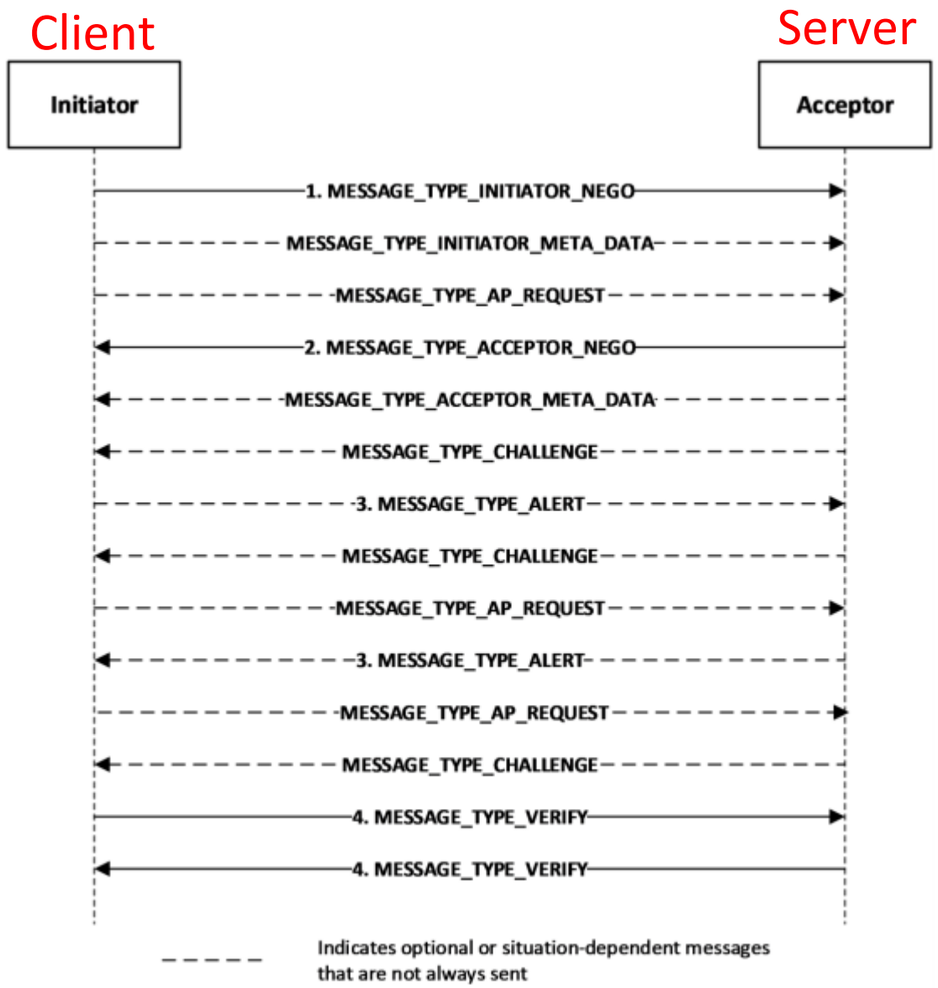
- SSPI is an API that allows application to add authenticity and privacy layer
- It is applicable to any application that allows 'Windows Authentication'



Ref: https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/security-support-provider-interface-architecture

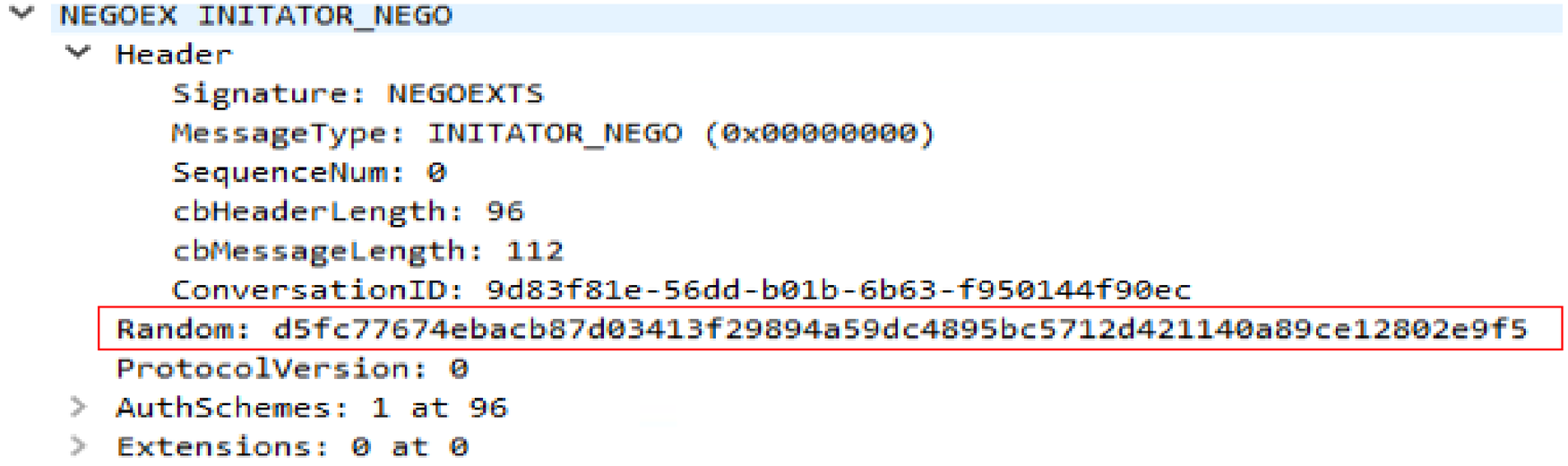
NegoEx protocol

NegoEx


Client
Server
Initiator
Acceptor
1. MESSAGE_TYPE_INITIATOR_NEGO
MESSAGE_TYPE_INITIATOR_META_DATA
MESSAGE_TYPE_AP_REQUEST
2. MESSAGE_TYPE_ACCEPTOR_NEGO
MESSAGE_TYPE_ACCEPTOR_META_DATA
MESSAGE_TYPE_CHALLENGE
3. MESSAGE_TYPE_ALERT
MESSAGE_TYPE_CHALLENGE
MESSAGE_TYPE_AP_REQUEST
3. MESSAGE_TYPE_ALERT
MESSAGE_TYPE_AP_REQUEST
MESSAGE_TYPE_CHALLENGE
4. MESSAGE_TYPE_VERIFY
4. MESSAGE_TYPE_VERIFY
Indicates optional or situation-dependent messages that are not always sent

Initiator nego

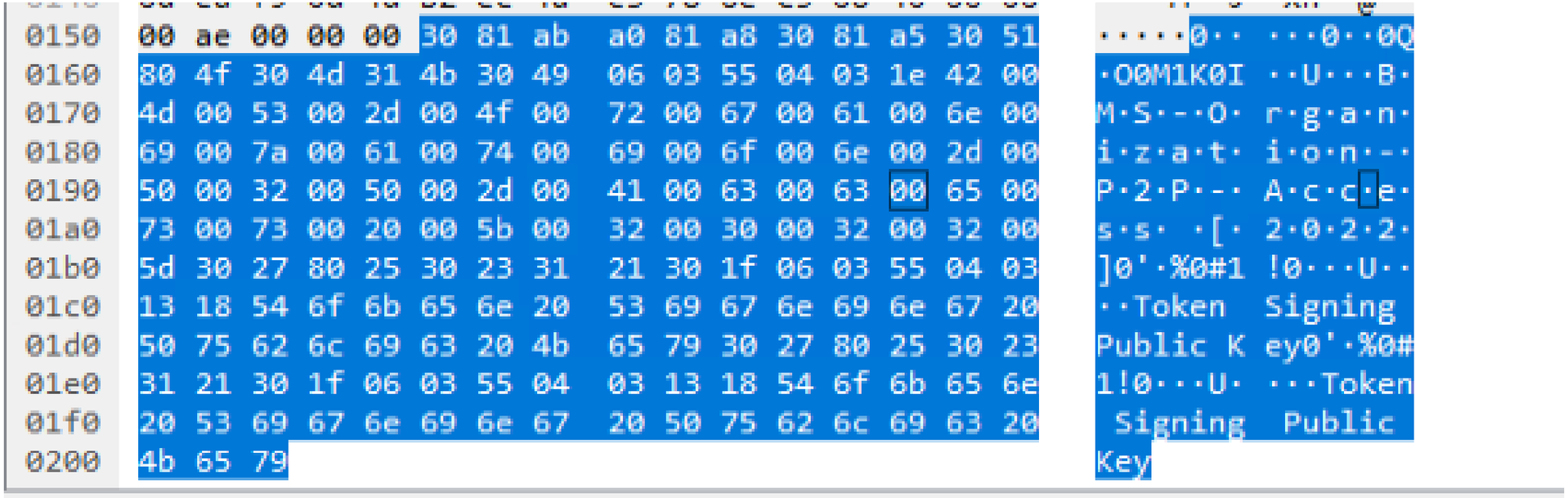
- Contains a random identifies the session



* Similar for the server side in Acceptor nego

Initiator metadata

Generated by GSS_Query_meta_data()

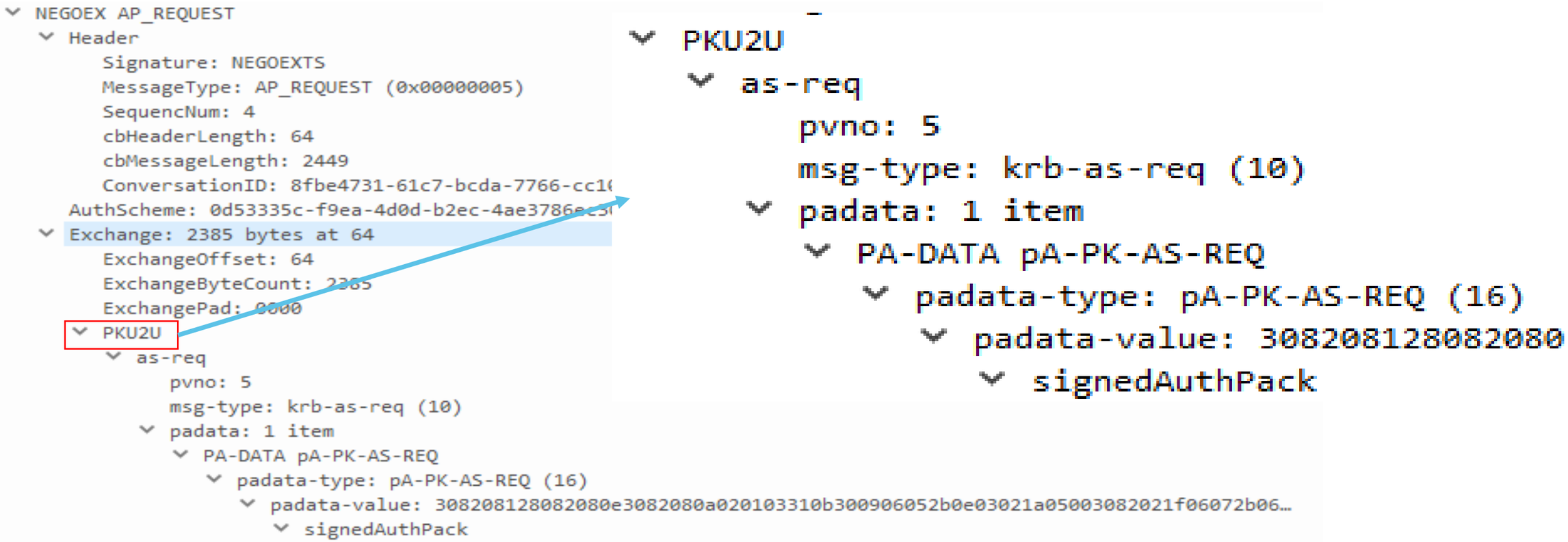


```
0150  00 ae 00 00 00 30 81 ab  a0 81 a8 30 81 a5 30 51  ·····0·· ···0··0Q
0160  80 4f 30 4d 31 4b 30 49  06 03 55 04 03 1e 42 00  ·O0M1K0I ··U···B·
0170  4d 00 53 00 2d 00 4f 00  72 00 67 00 61 00 6e 00  M·S·-·O· r·g·a·n·
0180  69 00 7a 00 61 00 74 00  69 00 6f 00 6e 00 2d 00  i·z·a·t· i·o·n·-·
0190  50 00 32 00 50 00 2d 00  41 00 63 00 63 00 65 00  P·2·P·-· A·c·c·e·
01a0  73 00 73 00 20 00 5b 00  32 00 30 00 32 00 32 00  s·s· ·[· 2·0·2·2·
01b0  5d 30 27 80 25 30 23 31  21 30 1f 06 03 55 04 03  ]0'·%0#1 !0···U··
01c0  13 18 54 6f 6b 65 6e 20  53 69 67 6e 69 6e 67 20  ··Token  Signing
01d0  50 75 62 6c 69 63 20 4b  65 79 30 27 80 25 30 23  Public K ey0'·%0#
01e0  31 21 30 1f 06 03 55 04  03 13 18 54 6f 6b 65 6e  1!0···U· ···Token
01f0  20 53 69 67 6e 69 6e 67  20 50 75 62 6c 69 63 20  Signing  Public
0200  4b 65 79                                          Key
```

* Similar for the server side in Acceptor metadata

AP_REQUEST

PKU2U part



CHALLENGE

PKU2U part

```
NEGOEX CHALLENGE
  > Header
    AuthScheme: 0d53335c-f9ea-4d0d-b2ec-4ae3786ec308
  v Exchange: 3143 bytes at 64
      ExchangeOffset: 64
      ExchangeByteCount: 3143
      ExchangePad: 0000
    v PKU2U
      v as-rep
          pvno: 5
          msg-type: krb-as-rep (11)
        v padata: 1 item
          v PA-DATA pA-PK-AS-REP
            v padata-type: pA-PK-AS-REP (17)
              v padata-value: a08206ab308206a78082067f3082
                > dhSignedData
          crealm: WELLKNOWN:PKU2U
        > cname
        v ticket
            tkt-vno: 5
            realm: WELLKNOWN:PKU2U
```

```
PKU2U
v as-rep
    pvno: 5
    msg-type: krb-as-rep (11)
  v padata: 1 item
    v PA-DATA pA-PK-AS-REP
      v padata-type: pA-PK-AS-REP (17)
        v padata-value: a08206ab30820(
          > dhSignedData
    crealm: WELLKNOWN:PKU2U
  > cname
  v ticket
      tkt-vno: 5
      realm: WELLKNOWN:PKU2U
```

VERIFY

- NegoEx validator
- Checksum all previous messages

```
NEGOEX VERIFY
  Header
      Signature: NEGOEXTS
      MessageType: VERIFY (0x00000006)
      SequencNum: 7
      cbHeaderLength: 80
      cbMessageLength: 92
      ConversationID: 8fbe4731-61c7-bcda-7766-cc10e83bb9b2
    AuthScheme: 0d53335c-f9ea-4d0d-b2ec-4ae3786ec308
  Checksum
      cbHeaderLength: 20
      ChecksumScheme: rfc3961 (1)
      ChecksumType: 16
    Checksum Vector: 12 at 80
        ChecksumOffset: 80
        ChecksumCount: 12
        ChecksumPad: 0000
        Checksum: bcf15036e12a6754e3a0fb50
```

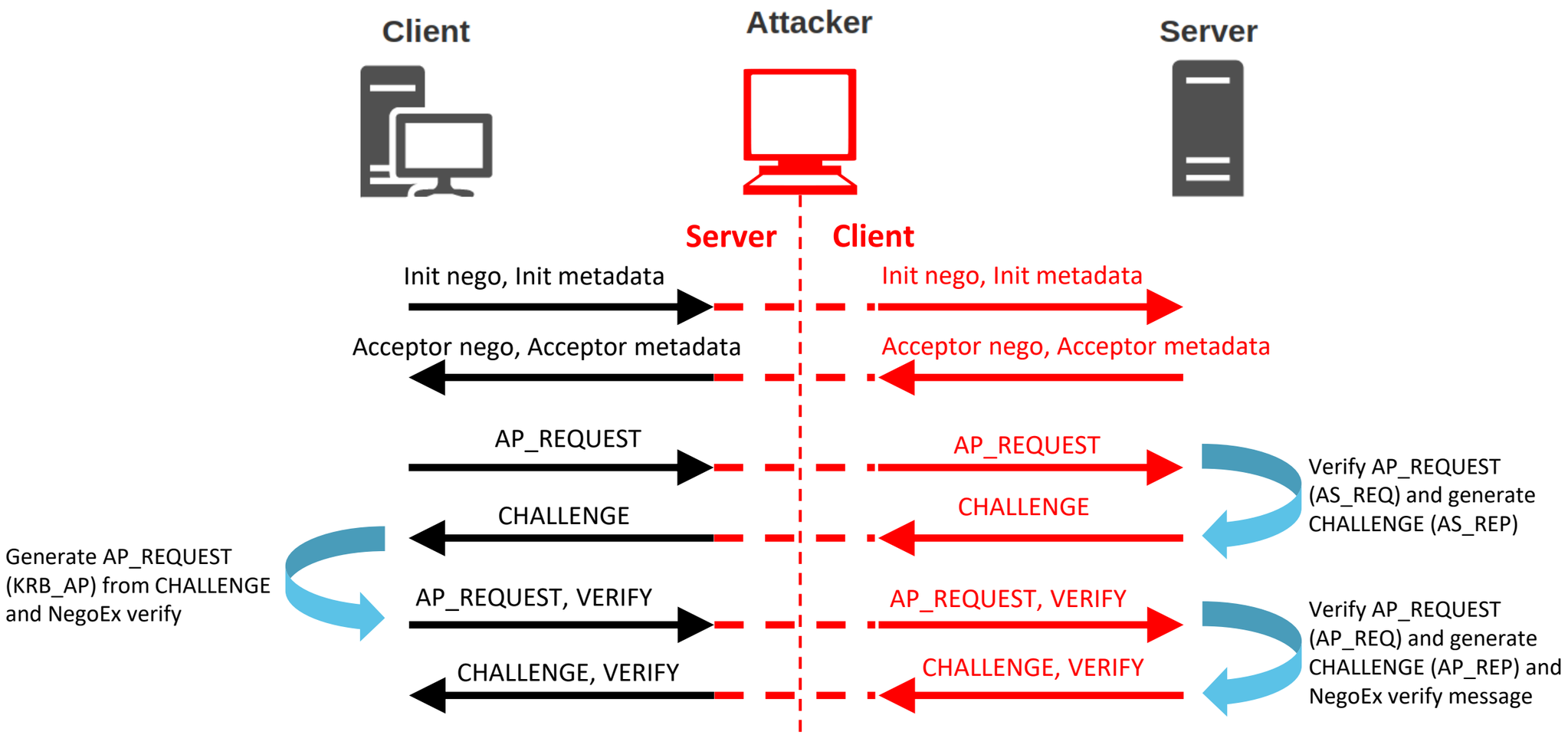
Attacks

NegoEx Relay

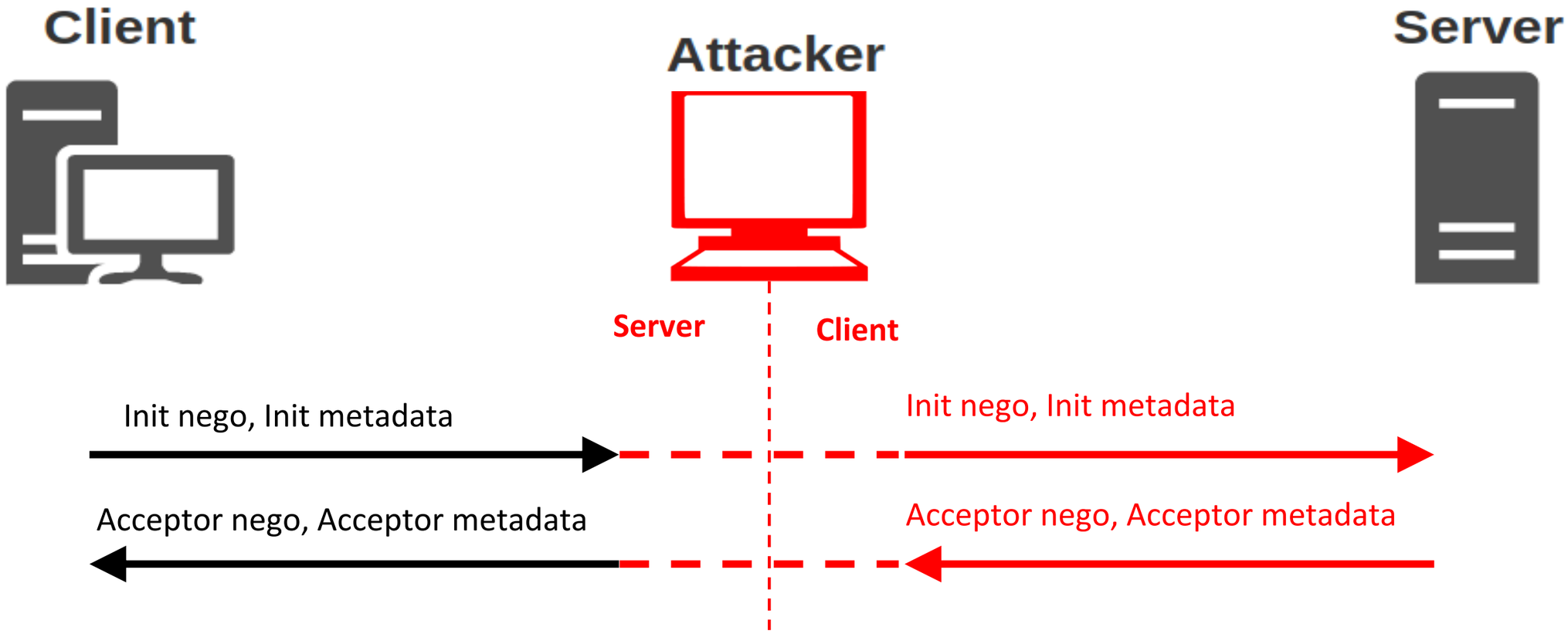
- Data is not shared between computers to find the “real” source
- Authentication process is easily relayed when not signed

* Relaying back to victim does not work

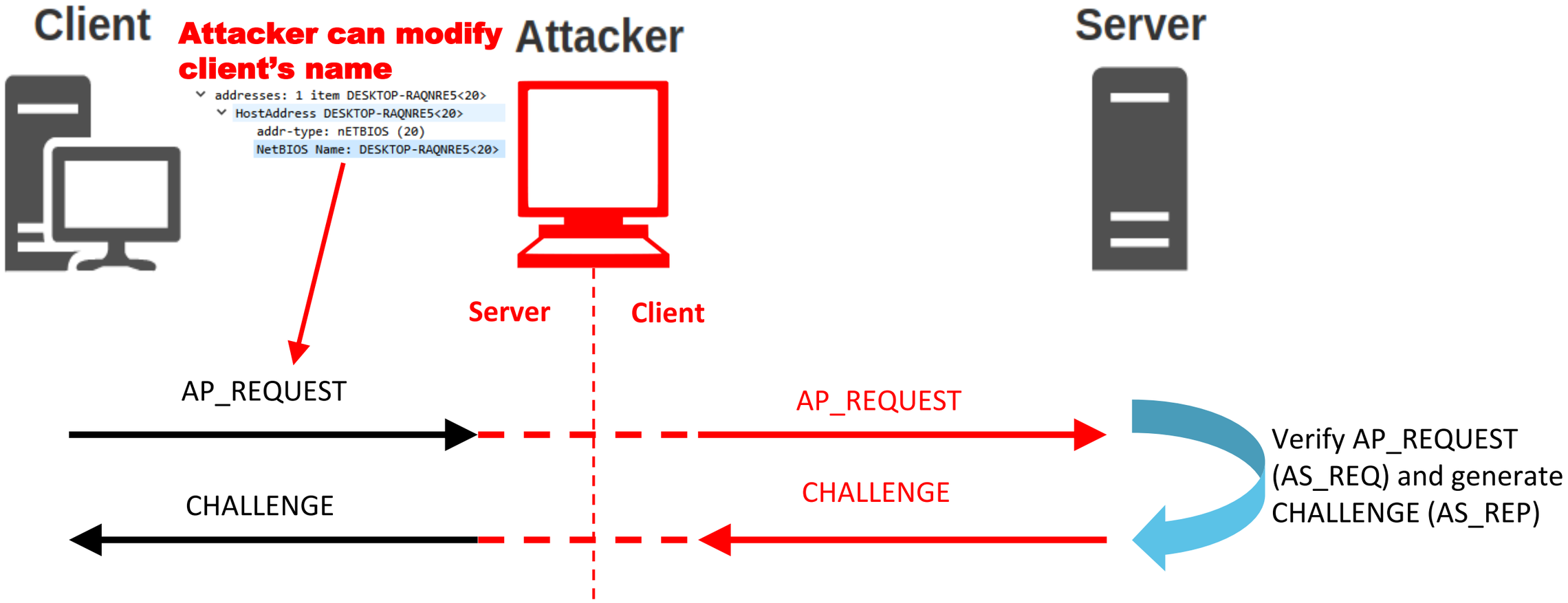
NegoEx Relay



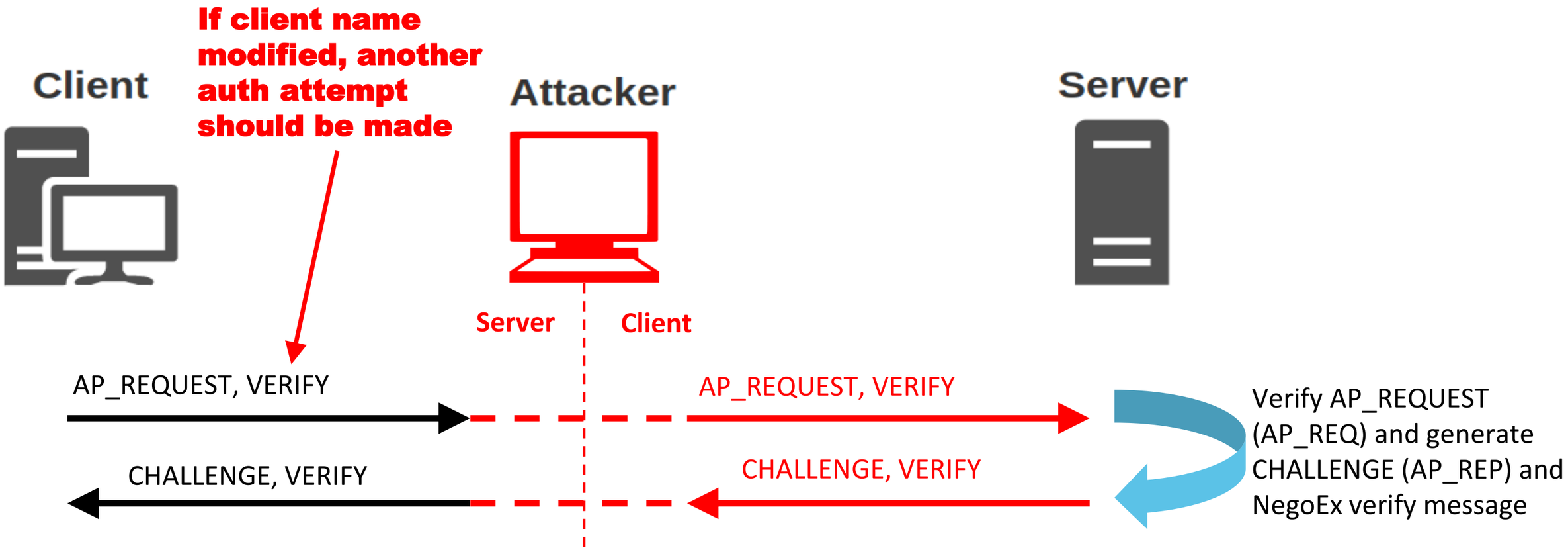
NegoEx Relay



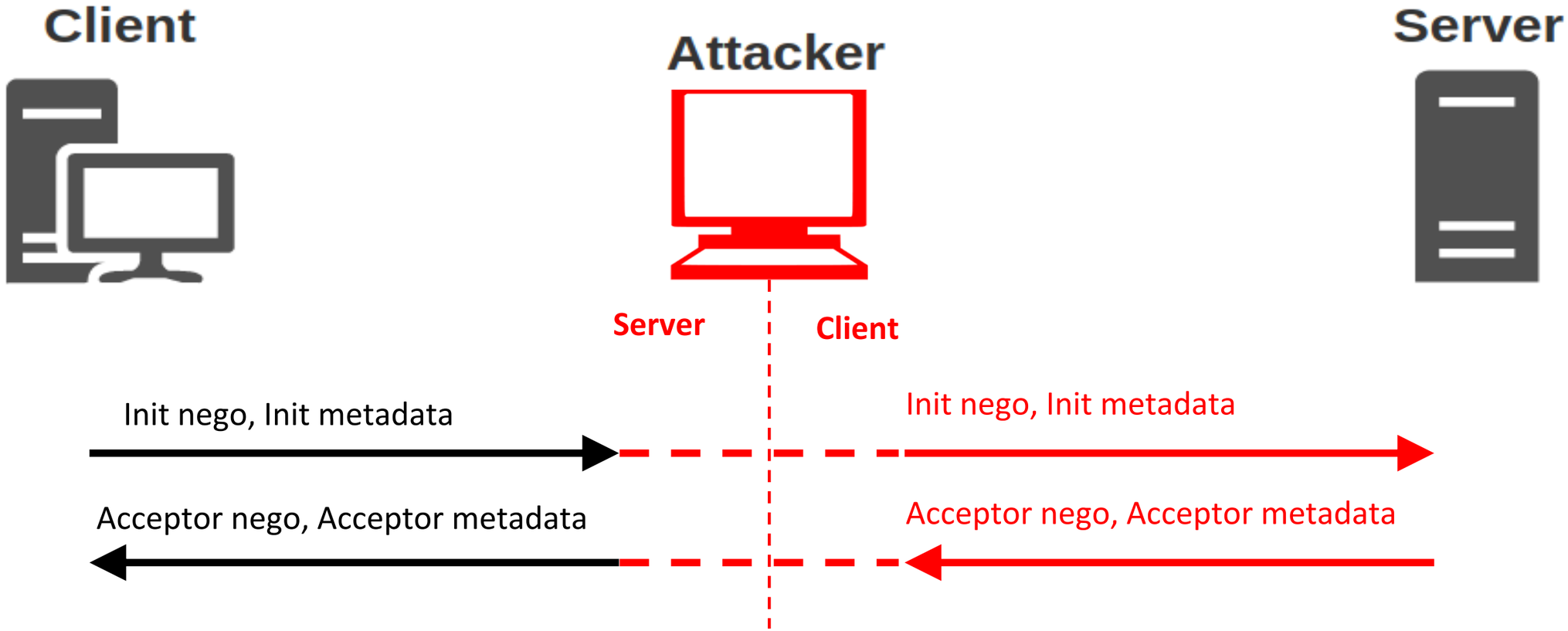
NegoEx Relay



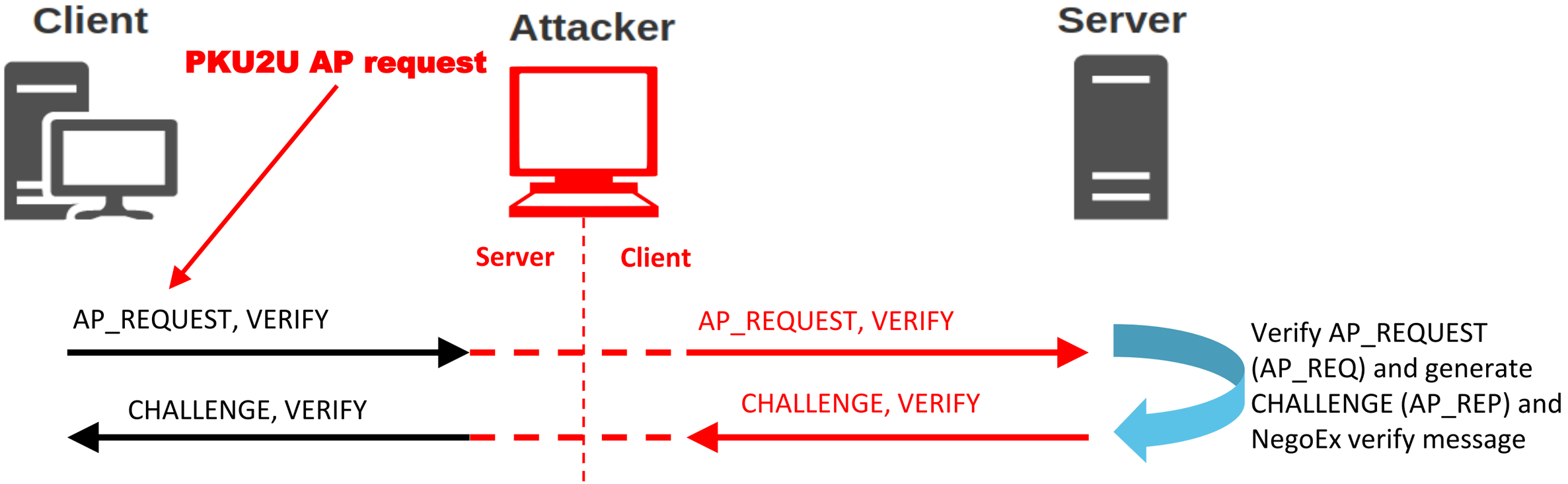
NegoEx Relay



NegoEx Relay – Changed client name



NegoEx Relay - Changed client name



Demo

PRT To Cert


```
{
  "alg": "HS256",
  "ctx": "XLYoL3j04fP8CzB6940G2LlKEgYuhRZ6"
}.{
  "aud": "login.microsoftonline.com",
  "cert_token_use": "user_cert",
  "client_id": "38aa3b87-a06d-4817-b275-7a316988d93b",
  "csr":
"MIICdTCCAV0CAQAwMDEuMCwGA1UEAwwldGplZmZvcmRzQGF0cHJlc2VhcmNoLm9ubWljcm9zb2Z0LmNvbTCCAS
HePQC83gOD5JZzbdtR1riGWi6X1CUn1mmt8/e5Vf3N7HE2xqz4mJzZlqMeGhy50ZaAIOLaB0uzw7FfS5coLPO2r
Ig3MoEXbWCclipp7y4rwPpNg4vhLdrshaM3EoTXHsakr5MitGF0AU5nHyNrvPFnYxS/gcaP9HQRSQbTBIHFpU4c
RY0RbDWzkUSqYbCBwpgnsndGdrF73K1mLPThu9rlIuqRmMcfIzelqRkiGwVA0pTEo8CAwEAAaAAMA0GCSqGSIb3
7UfQSM9AfK0GsT+FxdIH0HO8Pc/d5pYHSsVH5AISyjW7gvLWhZyl1BbNZHwe1GusfGFCaq7BI2y3dZgw3UY+hSFJ
UsQu+bNu0j45GGcXbYa3kGjufFgiLCTkmyFyxwpCSRhn37kNd0RUTZuGZAdxLhu3XIBBGEbSoIZGfYT7CCQO4Gc
KH90+Xrv+R+v5sqvKp/25rYJIgHKCBukyseyM+KGb",
  "csr_type": "http://schemas.microsoft.com/windows/pki/2009/01/enrollment#PKCS10",
  "grant_type": "refresh_token",
  "iss": "aad:brokerplugin",
  "refresh_token": "0.ATkAY9dbAZUjoUiJxnig3rfkOIc7qjhtoBdIsnV6MWmI2Ts5ABo.AgABAAAAAAD--
5ArU_aSiX8965g2mjzvDPK44gV406v3dz_7iAlmCKHNptk2dMtum7U8j7FT99Bz_3K6tPpEOtkT_29vrobffibm
qFIXYw_2uLiGrmJHjS4cJQ-lR7VY136fZ33Qe-G8tP_ePxk9wmCEGlBB6Ld9y9JhHYF8v5-
R9KmVW5WggRoBVmTCuTNs5Lqm18B71IxFWlWv4ZAhyi7Nkb1gg6AbQ2NO_6UoMs8KcZvx49BqF3WF3NvhDbf2yk
umZ7sm_spLZb4sqI15bvFr4L3_5Kw59JUy8hHEBTAnaR3J3FNuVfEa1eP36PKTWVSqioDBaOzMlTF-
33IQbSwFyvIqfy4mcFQU3sAOmQQ8t99GB6A1fPXCqdSor2uBKaP13HAYyKnX_QaFW4AXEDHVsL276lv4abzwrh4
_LjllTI1mDY1WG5cuu5xbI92uLRw1Cx713IjrPFEmKmWMran7qnTMGVtaSKUoW0Ybc7K6L0yePihKXsuJjSmDUS
zXb2EfdT_C4yZDRuwYr0SaD54B1yORcsdkYcsq0z8OuPHxHitfYhxRDKsXWRwWDSRG0Ns4bUbdlcJhliDGZ8_jW
Bei2bR0aoZZOpdvCIixVZSsndCYyhUas2tfGc_xyUTmNZyzdNKoPgTiNj9-dHPze0VFzcKNokYmlkQD17PETMwm
QL9HnsGsOEC_85JMTQt0pFnVn2hbf0NNmB8PJF7nZp3R_BQytg2e1pe8VCac3ds-
3BDN0h9hj20OkdXz4izdP98hfUJ5dF5Oly1BkXDHQaGDvOIzPk7PnOs9LIy_1g6iehIB3gI7D35z9IXUvGDbHLv
guYDtL0GwLd5ZwEymRj8yIU",
  "request_nonce": "AwABAAAAAAACAOz_BAD0_zOQzto8WLOcVlXqAjnuuHhAoRyff_t5sEhPPU1OMU2xlK
  "scope": "openid aza ugs"
}.[Signature]
```

PRT To Cert

Body

Name	Value
grant_type	urn:ietf:params:oauth:grant-type:jwt-bearer
request	eyJhbGciOiJIUzI1NiIsImN0eCI6IlhMWW9MM2owN(

Transformer | Headers | TextView | SyntaxView | ImageView | HexView | WebView | Auth | Caching | Cookies | R

JSON
- cert_token_use=user_cert
- expires_in=3599
- expires_on=1613896405
- ext_expires_in=0
- id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJhdWQiOiIzOGFhM2I4Ny1hMDZkLTQ4MTctYjI3NS03YTMxNjk4OGQ5M2IiLCJpc
- refresh_token=0.ATkAY9dbAZUjoUiJxnig3rfkOIc7qjhtoBdIsnV6MWmI2Ts5ABo.AgABAAAAAAD--DLA3VO7QrddgJg7WevrAgDs_wQ
- resource=urn:p2p_cert
- session_key_jwe=eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiUlNBLU9BRVAifQ.nrrvl9-iSgJAwtWDN0JgQlH1wvEr__XzzwTUoe0vnojMICbfr
- token_type=x509
- x5c=MIIEVTCCAz2gAwIBAgIQJOEcwa1CJ/ucbzsc7aj4LzANBgkqhkiG9w0BAQsFADBNMUswSQYDVQQDHkIATQBTAC0ATwByAGcAY(
- x5c_ca=MIIDRTCCAi2gAwIBAgIQG8IX37s9HpRBMbsfAj5ThDANBgkqhkiG9w0BAQsFADBNMUswSQYDVQQDHkIATQBTAC0ATwByAG

Pass the certificate

- Possible by requesting new certificate from PRT or by stealing a certificate from the certificate store on computer

160	5.659343	172.23.1.176	172.23.1.63	TCP	1514 63006 → 445 [ACK] Seq=300 Ack=1163 Win=2100992 Len=1460 [TCP segment of
161	5.659343	172.23.1.176	172.23.1.63	SMB2	1502 Session Setup Request, INITATOR_NEGO, INITIATOR_META_DATA, AP_REQUEST
162	5.659413	172.23.1.63	172.23.1.176	TCP	54 445 → 63006 [ACK] Seq=1163 Ack=3208 Win=2102272 Len=0
163	5.691521	172.23.1.63	172.23.1.176	SMB2	3372 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, CHALLENG
164	5.691868	172.23.1.176	172.23.1.63	TCP	54 63006 → 445 [ACK] Seq=3208 Ack=4481 Win=2102272 Len=0
165	5.693763	172.23.1.176	172.23.1.63	TCP	1514 63006 → 445 [ACK] Seq=3208 Ack=4481 Win=2102272 Len=1460 [TCP segment of
166	5.693763	172.23.1.176	172.23.1.63	SMB2	317 Session Setup Request, AP_REQUEST, VERIFY
167	5.693802	172.23.1.63	172.23.1.176	TCP	54 445 → 63006 [ACK] Seq=4481 Ack=4931 Win=2102272 Len=0
173	5.703085	172.23.1.63	172.23.1.176	SMB2	458 Session Setup Response, CHALLENGE, VERIFY
174	5.703638	172.23.1.176	172.23.1.63	SMB2	158 Tree Connect Request Tree: \\client7\IPC$

Hunting

Windows events

Windows event 4624 can be leveraged for hunting of NegoEx logins

```
New Logon:
        Security ID:            AzureAD\yos
        Account Name:           AzureAD\yos@ResearchAadLabEnv.onmicrosoft.com
        Account Domain:         -
        Logon ID:               0x622EF72

Network Information:
        Workstation Name:       DESKTOP-RAQNRE5
        Source Network Address: 172.18.255.51
        Source Port:            57137

Detailed Authentication Information:
        Logon Process:          Pku2uSsp
        Authentication Package: NegoExtender
        Transited Services:     -
        Package Name (NTLM only):       -
        Key Length:             0
```

Traffic analysis

Traffic analysis tools like Zeek can parse NegoEx and extract the relevant data

```
SMB2      1502 Session Setup Request, INITATOR_NEGO, INITIATOR_META_DATA, AP_REQ…

   version: v3 (2)
   serialNumber: 0x5c6cffa328ee476316b18f684136cc03
 ▾ signature (sha256WithRSAEncryption)
      Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
 ▾ issuer: rdnSequence (0)
   ▾ rdnSequence: 1 item (id-at-commonName=\000M\000S\000-\000O\000r\000g\000a\000n\000i\)
     ▾ RDNSequence item: 1 item (id-at-commonName=\000M\000S\000-\000O\000r\000g\000a\000
       ▾ RelativeDistinguishedName item (id-at-commonName=\000M\000S\000-\000O\000r\000g'
            Id: 2.5.4.3 (id-at-commonName)
         ▾ DirectoryString: bmpString (3)
              bmpString: MS-Organization-P2P-Access [2022]
 ▾ validity
   ▸ notBefore: utcTime (0)
   ▸ notAfter: utcTime (0)
 ▾ subject: rdnSequence (0)
   ▾ rdnSequence: 3 items (id-at-commonName=yos@ResearchAadLabEnv.onmicrosoft.com,id-at-co
```

Serial Number

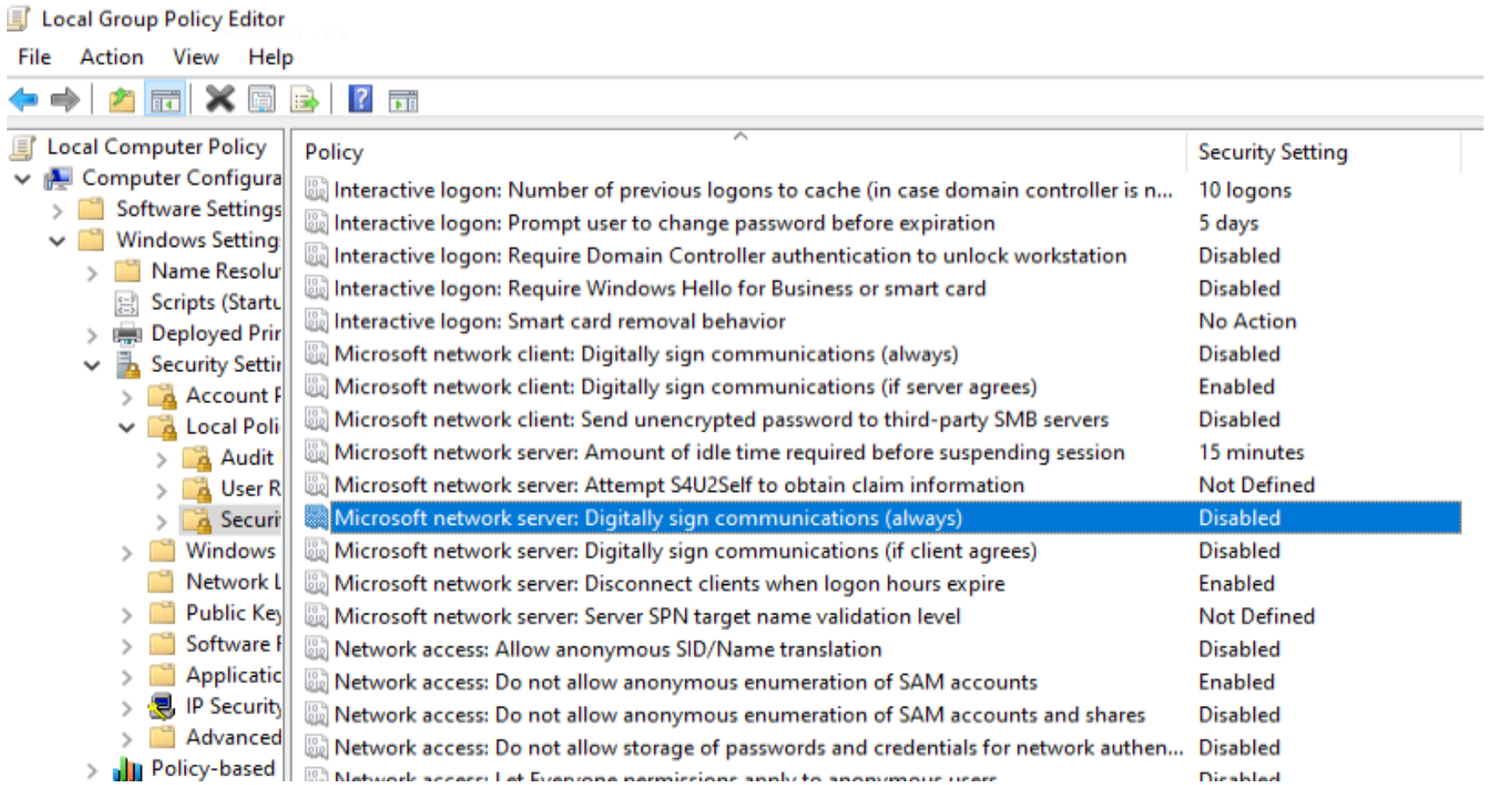
5C6CFFA328EE476316B18F684136CC03

Subject

CN=yos@ResearchAadLabEnv.onmicrosoft.com,CN=S-1-12-1-

Mitigations

- SMB signing



Tools

- NegoEx relay tool
- Azure AD certificate requestor
- Authentication tool with Azure AD certificate
- Wireshark dissector for NegoEx
- Zeek build that dissect NegoEx Kerberos packets

Takeaways

- Patching is not enough
- Usage of SMB signing is a must
- Hunt hunt hunt

Thank you
Questions?

@Rubin_mor