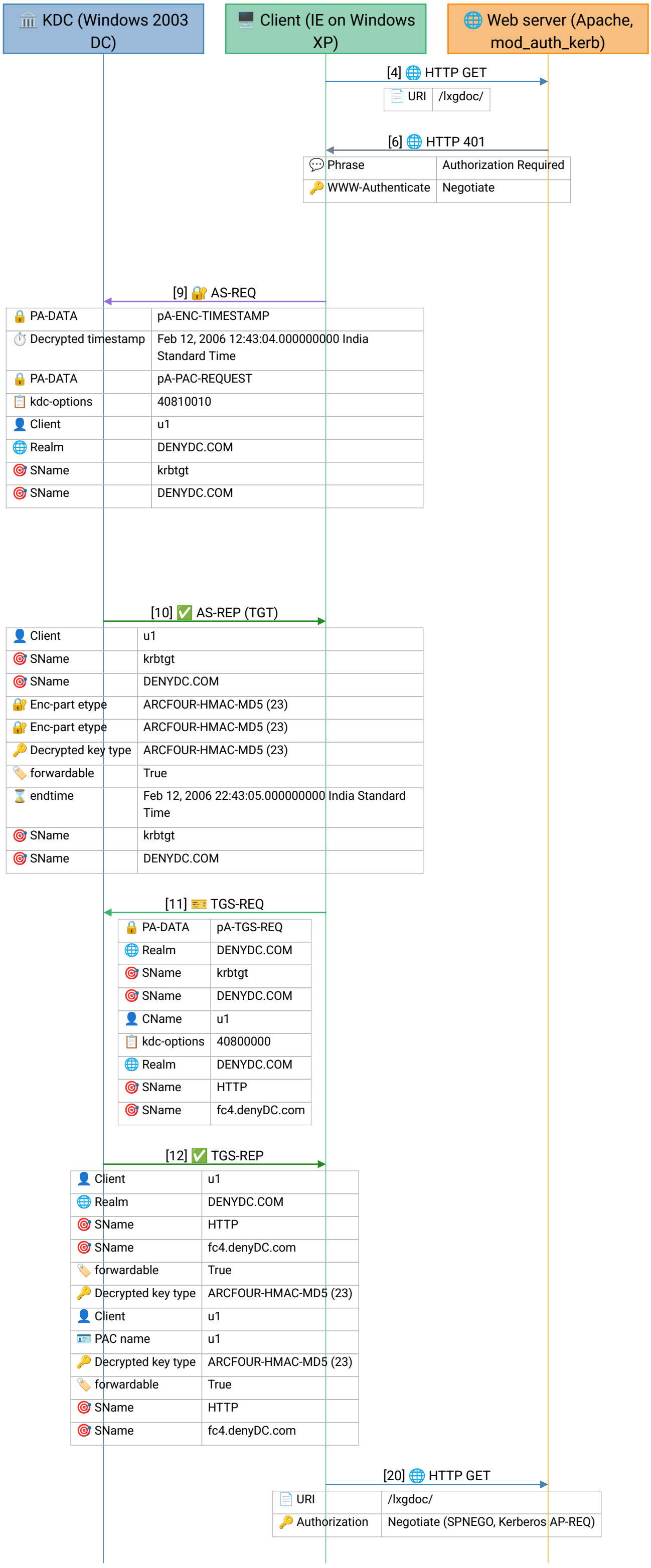
kerberos-ticket-exchange-http.annotated



[Frame 4] Phase 1: the web server asks for Kerberos. User `u1` on a Windows XP client opens `http://fc4/lxgdoc/` in Internet Explorer. The first request carries no credentials.

[Frame 6] The Apache server (with `mod_auth_kerb`) refuses with `401` and the bare header `WWW-Authenticate: Negotiate`. That invites the browser to authenticate with SPNEGO, which here means Kerberos (RFC 4559).

To log in to the web server, the client needs a service ticket for `HTTP/fc4.denyDC.com`. To get that, it first needs a TGT (ticket-granting ticket) from the KDC.

[Frame 9] Phase 2: the AS exchange gets a TGT. The client asks the KDC (Key Distribution Center, the Windows 2003 domain controller) for a TGT for `u1`: the request names the service `krbtgt/DENYDC.COM`.

It proves it knows u1's password without sending it. `PA-ENC-TIMESTAMP` is the current time encrypted with a key derived from the password (RFC 4120 5.2.7.2). Decrypted, it reads 12 Feb 2006 12:43:04. The KDC decrypts it with its copy of the key; a wrong password or a stale time fails here.

The capture file's own timestamps say 24 January: the sniffer's clock was wrong, not the client's. Kerberos only needs client and KDC to agree, within 5 minutes by default.

`PA-PAC-REQUEST` asks for a PAC (Privilege Attribute Certificate), Microsoft's list of the user's groups. The client offers only RC4 and DES encryption types, as Windows XP did.

[Frame 10] The KDC answers with two encrypted parts:

- **The TGT,** encrypted with the `krbtgt` account's key. Only KDCs can read it, so the client stores it without opening it (our keytab can't open it either).
- **The enc-part,** encrypted with u1's key. Decrypted, it holds the TGT session key (RC4), the ticket flags `forwardable, renewable, initial, pre-authent,` and a 10-hour lifetime (12:43 to 22:43), renewable for 7 days.

In this lab the keytab's one key, for `HTTP/fc4,` also opens u1's enc-part, so u1 and the web service share a key.

[Frame 11] Phase 3: the TGS exchange gets a service ticket. The client sends the TGT back to the KDC, now in its TGS (ticket-granting service) role, asking for `HTTP/fc4.denyDC.com`.

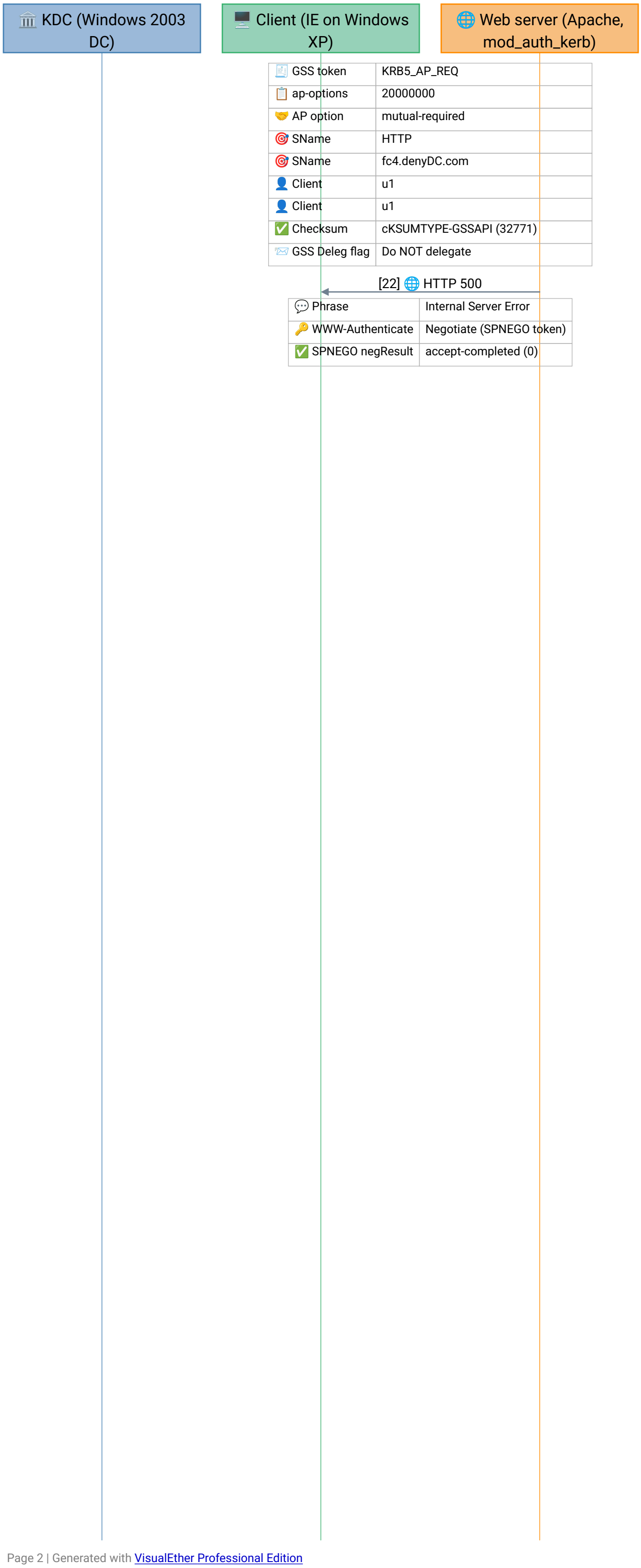
The TGT rides in `PA-TGS-REQ` as an AP-REQ: the TGT plus an authenticator (u1's name and the time) encrypted with the TGT session key from frame 10. No password is involved from here on.

[Frame 12] The KDC returns the service ticket, encrypted with the web service's key, so our keytab opens it. Inside:

- client `u1,` the session key for talking to the web server, the same end time as the TGT,
- flags `forwardable, renewable, pre-authent.` There is no `ok-as-delegate`: the KDC is not telling the client this service may receive delegated credentials (compare the delegation capture),
- the PAC: user RID 1111 and groups 513 (Domain Users) and 512 (Domain Admins).

The outer enc-part, under the TGT session key, gives the client its own copy of the new session key.

[Frame 20] Phase 4: the AP exchange logs in to the web server. The browser retries the GET with `Authorization: Negotiate` and a SPNEGO token holding a Kerberos AP-REQ: the service ticket from frame 12 plus a fresh authenticator.



The web server decrypts the ticket with its own key, takes out the session key, and decrypts the authenticator. A matching name and a current time prove the client holds the session key, so it is u1. The server never contacts the KDC.

The authenticator's checksum is the GSS-API type (0x8003, RFC 4121 4.1.1). Its flags request `Mutual` authentication and set `Deleg` to 0: the client forwards no credentials. `mutual-required` is also set in `ap-options`.

[Frame 22] SPNEGO says `accept-completed`: Kerberos authentication succeeded. The `500 Internal Server Error` comes from the page itself, after authentication; it is not a Kerberos failure. The delegation capture ends the same way.

This SPNEGO answer carries no AP-REP token, so the client gets no proof of the server's identity in this exchange despite asking for it.