SYNACKTIV

A study on Windows authentication & Prox-Ez

THCon 2023

Geoffrey Bertoli & Pierre Milioni

Who are we?

- **Geoffrey Bertoli (*@YofBalibump*) & Pierre Milioni (*@b1two_*)**
  - Pentesters at Synacktiv
- **Working for Synacktiv**
  - Offensive security
  - ~140 ninjas: pentest, reverse engineering, development, CSIRT
  - 4 locations: Paris, Rennes, Lyon, Toulouse, (soon at Lille) & remote
  - We are hiring! → *apply@synacktiv.com*

Introduction

- **A little bit of history**
  - NTLM introduced in 1993 with Windows NT 3.1
  - NTLMv2 since Windows NT 4.0 SP4 – 1998
  - But here comes the mighty Kerberos
    - Became a standard in 1993 (v5)
    - Introduced in Windows 2000
  - NTLM still widely used nowadays

Introduction

- **Multiple mitigations against relay**
  - NTLMv1 → v2 (not our focus today)
  - NTLM – MIC (not our focus today)
  - NTLM EPA (Extended Protection for Authentication)
    - Channel Binding
    - Service Binding
  - Kerberos
    - Whole new authentication mechanism
    - More complex than NTLM
    - Mutual authentication
    - Fix relay attack

Introduction

- **Still of interest today**
  - Lack of (proper) documentation of some topics
  - Not supported by all tools
  - Lack of tooling for these authentications over HTTP
- **Prox-Ez**
  - MitM proxy for Windows authentication over HTTP(s)
  - Single file, born to be patched

Agenda

- **Quick overview of NTLM**
- **NTLM and relaying**
- **NTLM-EPA (Extended Protection for Authentication)**
  - Channel Binding
  - Service Binding
- **What about Kerberos?**
  - Over HTTP
  - Security overview

NTLM

- **New Technology Lan Manager**
  - Windows authentication protocol
  - Single Sign-On
  - Based on challenge/response exchange
  - Authenticates a session (TCP connection in case of HTTP)
    - May cause issues/slowdowns with programs that creates new TCP connections for each request

      BurpSuite now supports TCP connection reuse

- **… over HTTP**

Client's browser

Web server

Client initial HTTP request
GET / HTTP

Server requires authentication
401 + *WWW-Authenticate: NTLM*

Client NTLM Negotiate message
Authorization: NTLM <negotiate>

Server NTLM Challenge message
WWW-Authenticate: NTLM <challenge>

Client NTLM Auth message
Authorization: NTLM <auth>

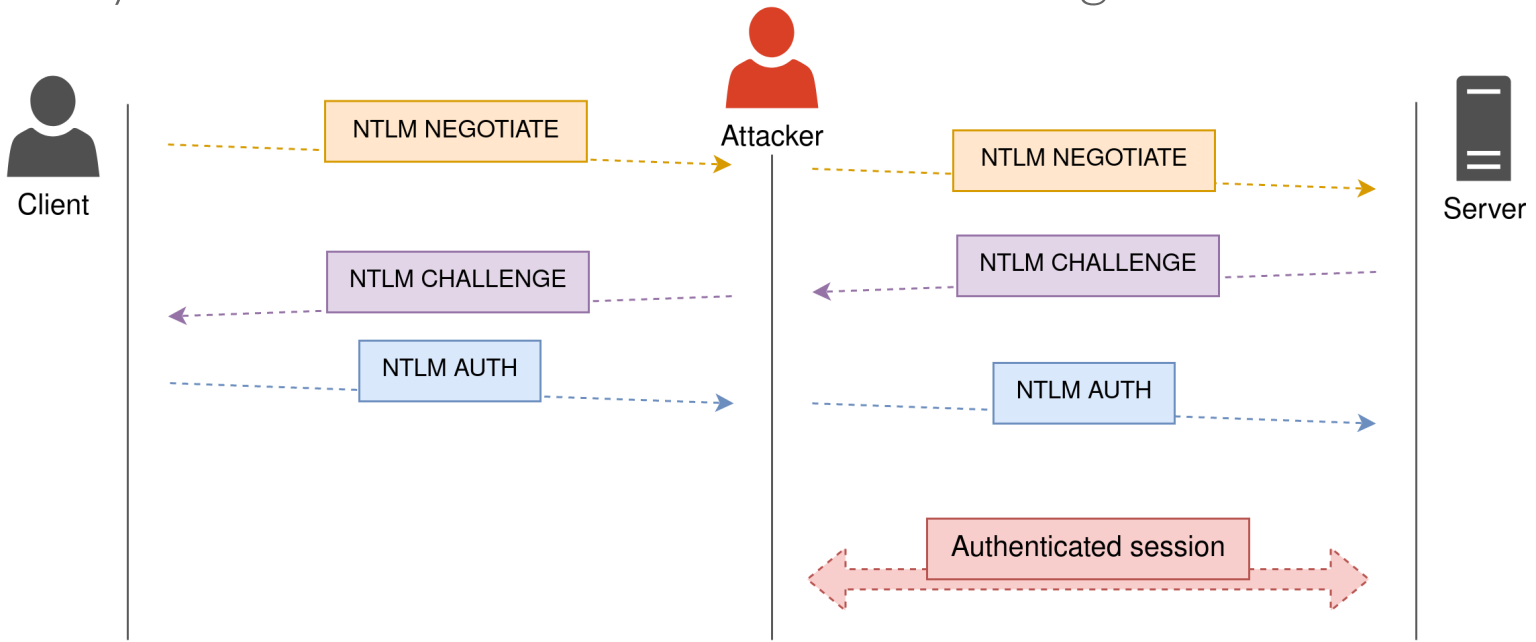
Server returns initially requested page
HTTP/1.1 200

More HTTP requests can be sent
in this authenticated TCP connection

NTLM

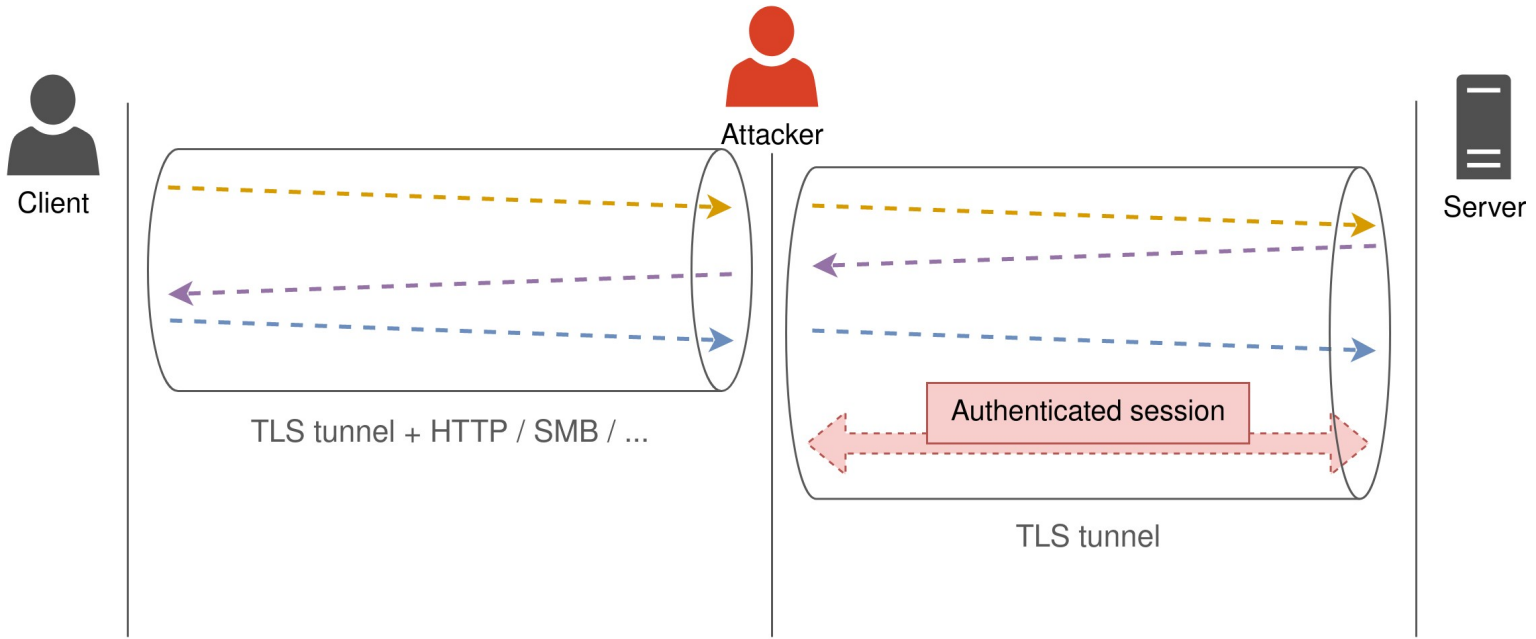
NTLM relaying

- Attacker in a relaying position (able to forward messages from a client)
- Relays the client's authentication to the targeted server



NTLM relaying – Over TLS

- Attacker in a relaying position
- Relays the client's authentication to the targeted server

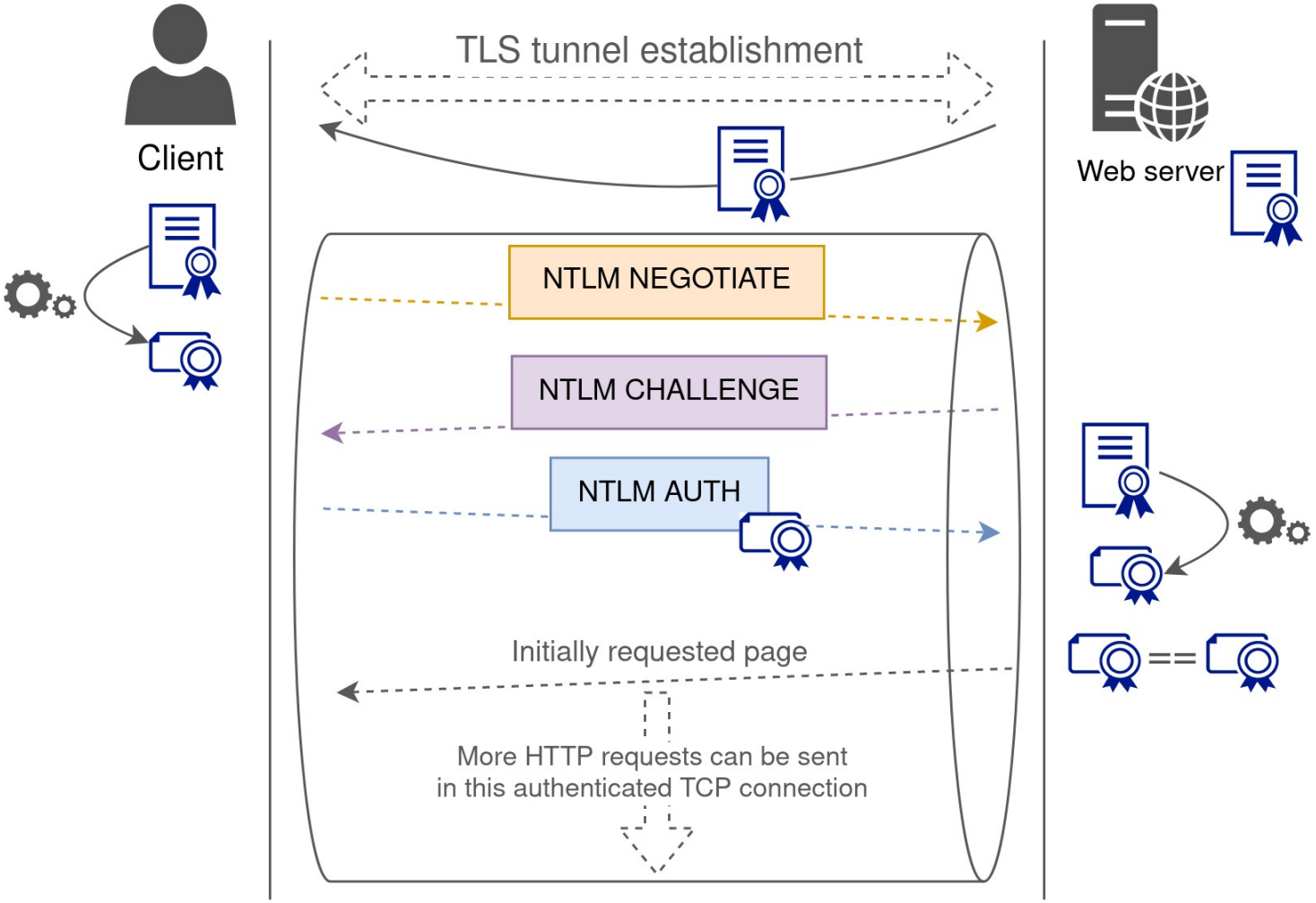


NTLM EPA – Channel Binding

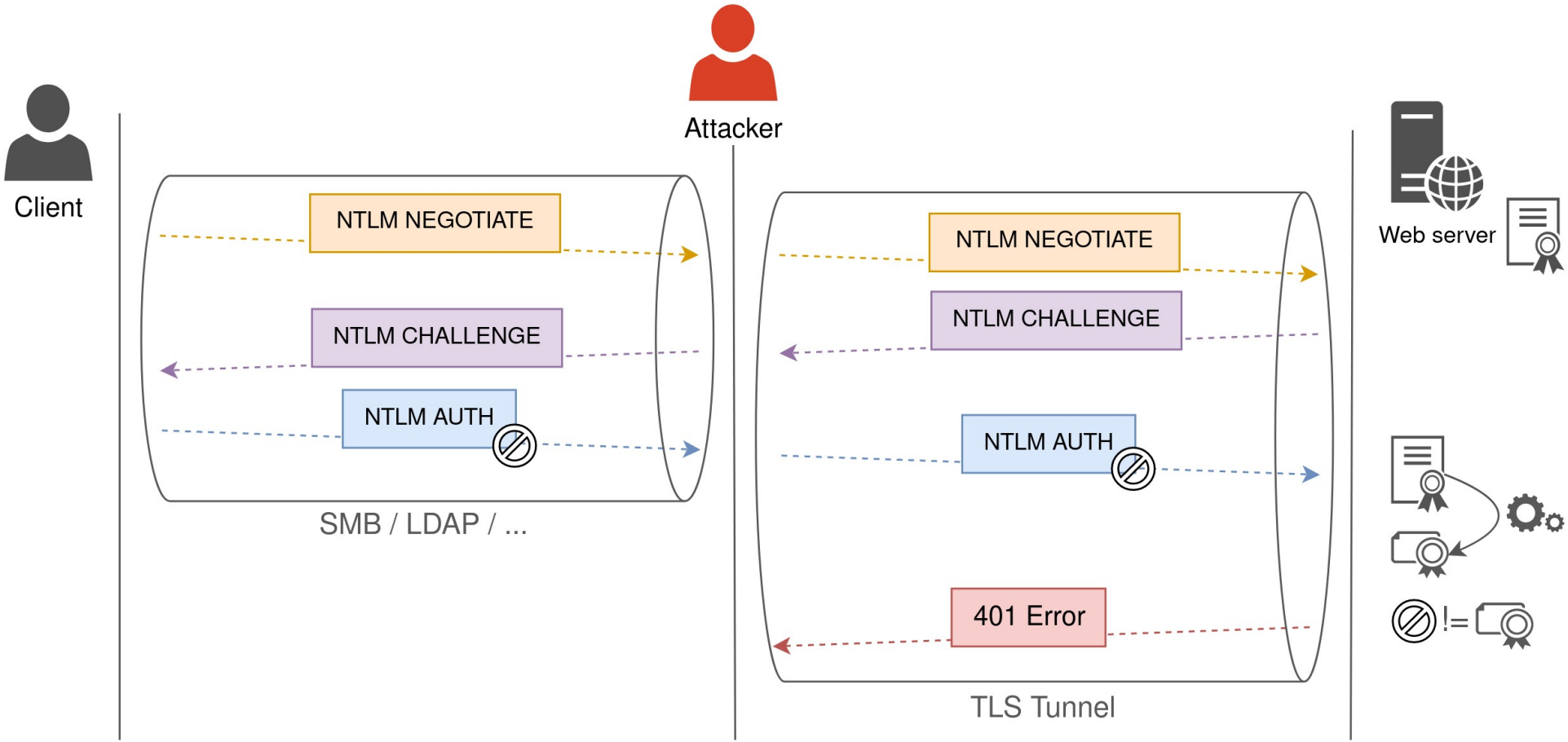
- **Channel Binding**
  - Microsoft's solution to protect against MitM attacks
  - Used on TLS based communications
  - "Binds" the authentication to the outer TLS channel

    → Adds a token that depends on the TLS tunnel into the NTLM authentication
  - Can be *required* by the server
    - Any client without the proper token are denied access

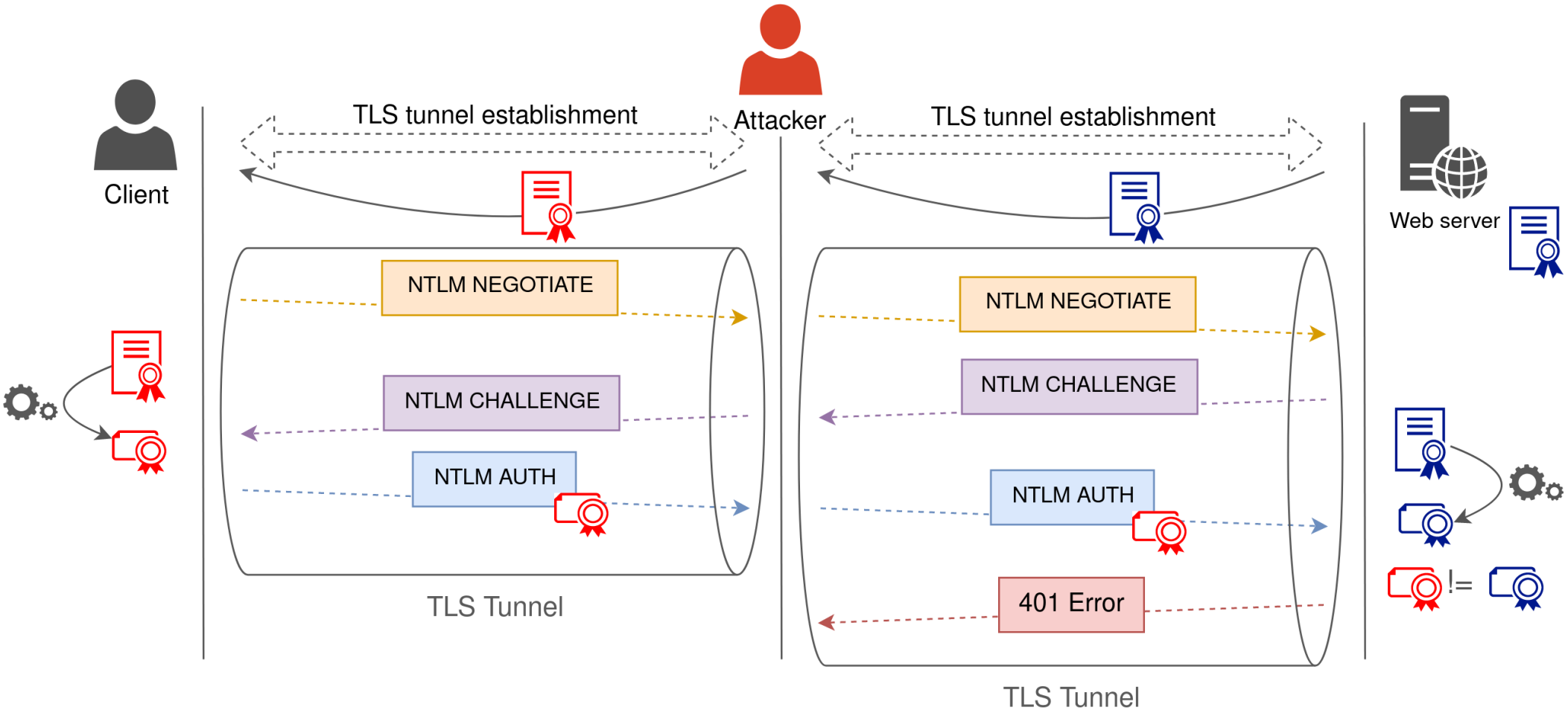
NTLM EPA – Channel Binding



NTLM EPA – Channel Binding



NTLM EPA – Channel Binding



NTLM EPA – Channel Binding

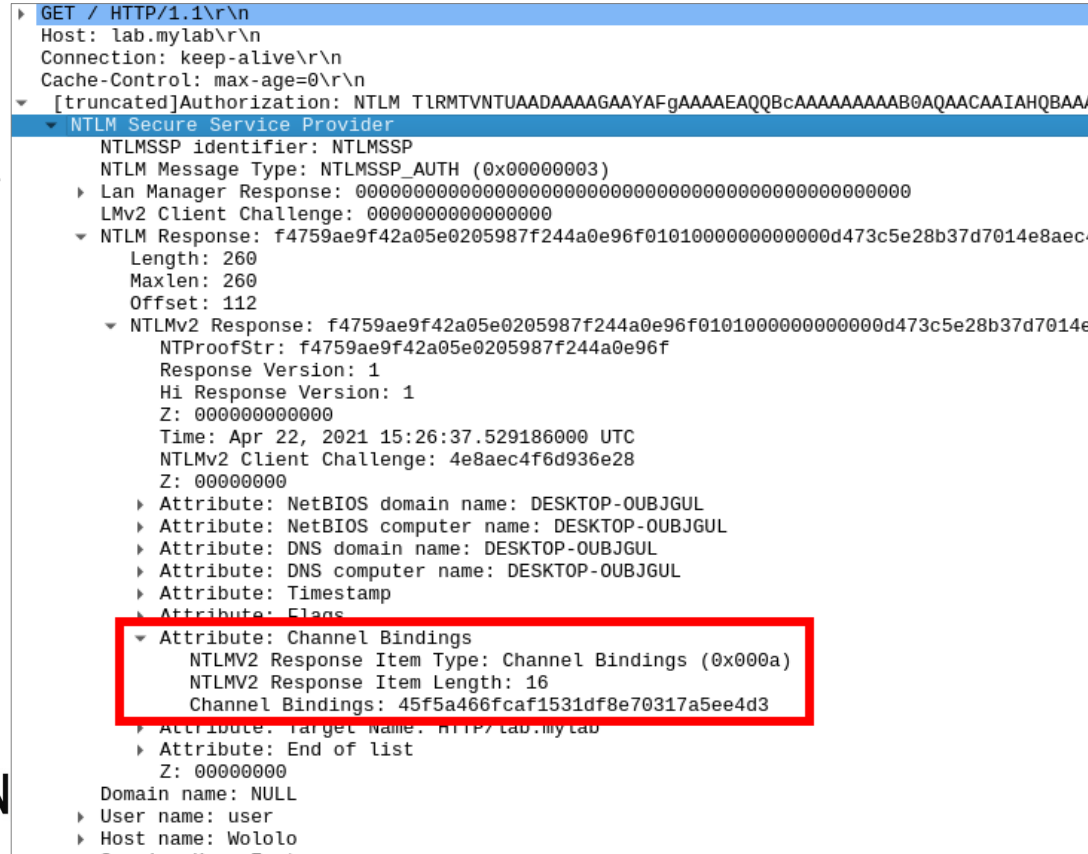
- **CBT: *Channel Binding Token***
  - Hash of the server's certificate
  - With the hash function used to compute the certificate's signature

Certificate signature's hash function	MD5 / SHA-1	Other hash function	No hash function / multiple hash functions
CBT's hash function	SHA-256	Signature's hash function	Undefined

NTLM EPA – Channel Binding

- ***Channel Bindings* attribute**

- Derived from the CBT
- Inserted in the NTLM *AUTH* message
- Cannot be modified
  - Protected by the MIC value (HMAC)
    → protected by the *Flags* attribute
    → protected by *NTProofStr*
    → protected by the client's secret



NTLM EPA – Channel Binding

- **Channel Binding**
  - Still not supported by many clients
    → no authentication possible if EPA is required
  - How to use our tools against EPA protected websites?

MitM Proxy – Prox-Ez ("prox easy")

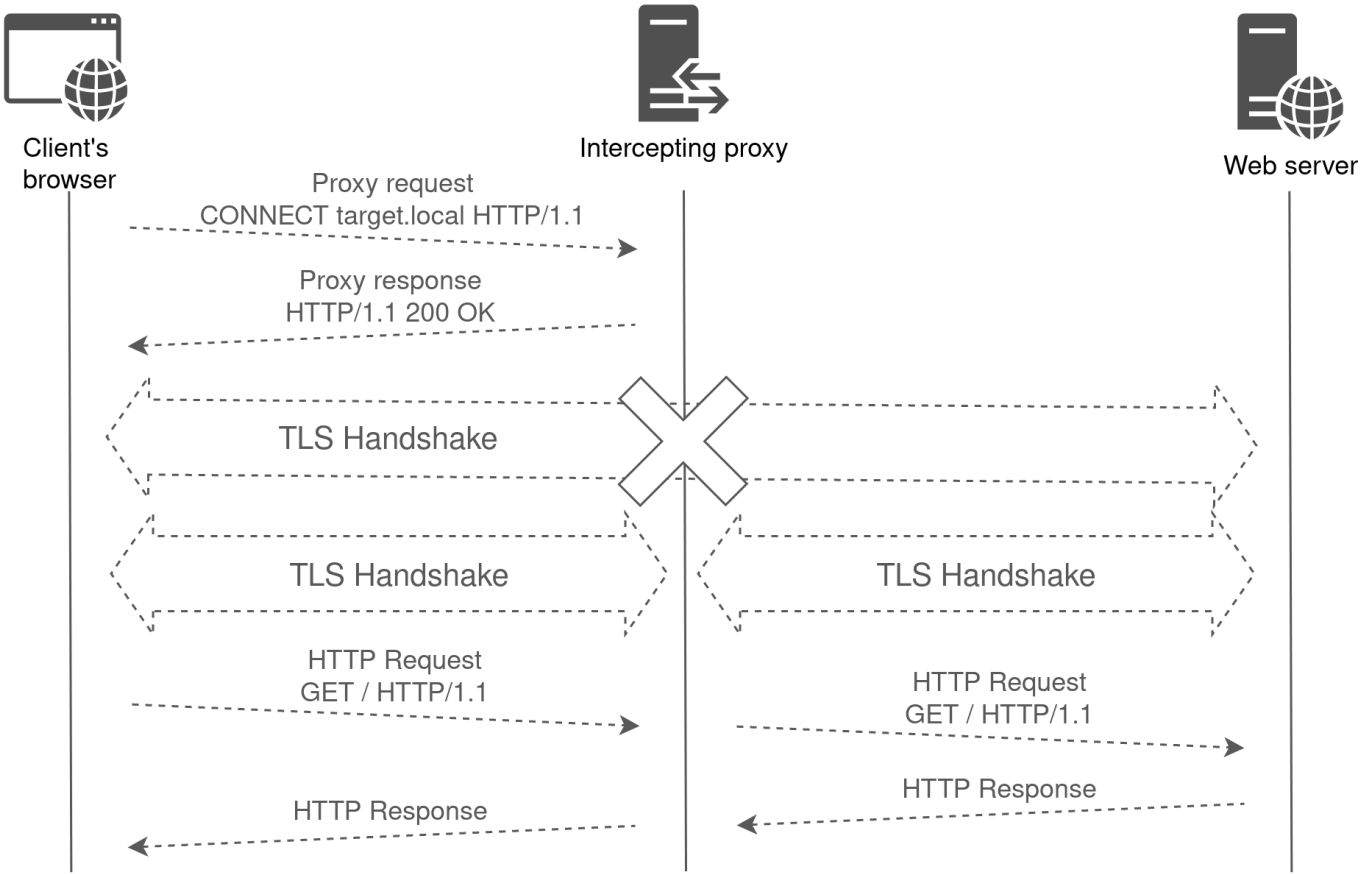
- **Why?**
  - Be able to use any tool against HTTP(s) servers using
    - NTLM
    - NTLM-EPA
    - *Kerberos*
  - Be able to control the authentication

MitM Proxy – Prox-Ez ("prox easy")

- **How?**
  - Has to work with TLS
    → TLS interception
    → Register a custom certificate authority on the client
    → Generate on-the-fly certificates
  - Good documentation on *mitmproxy* website

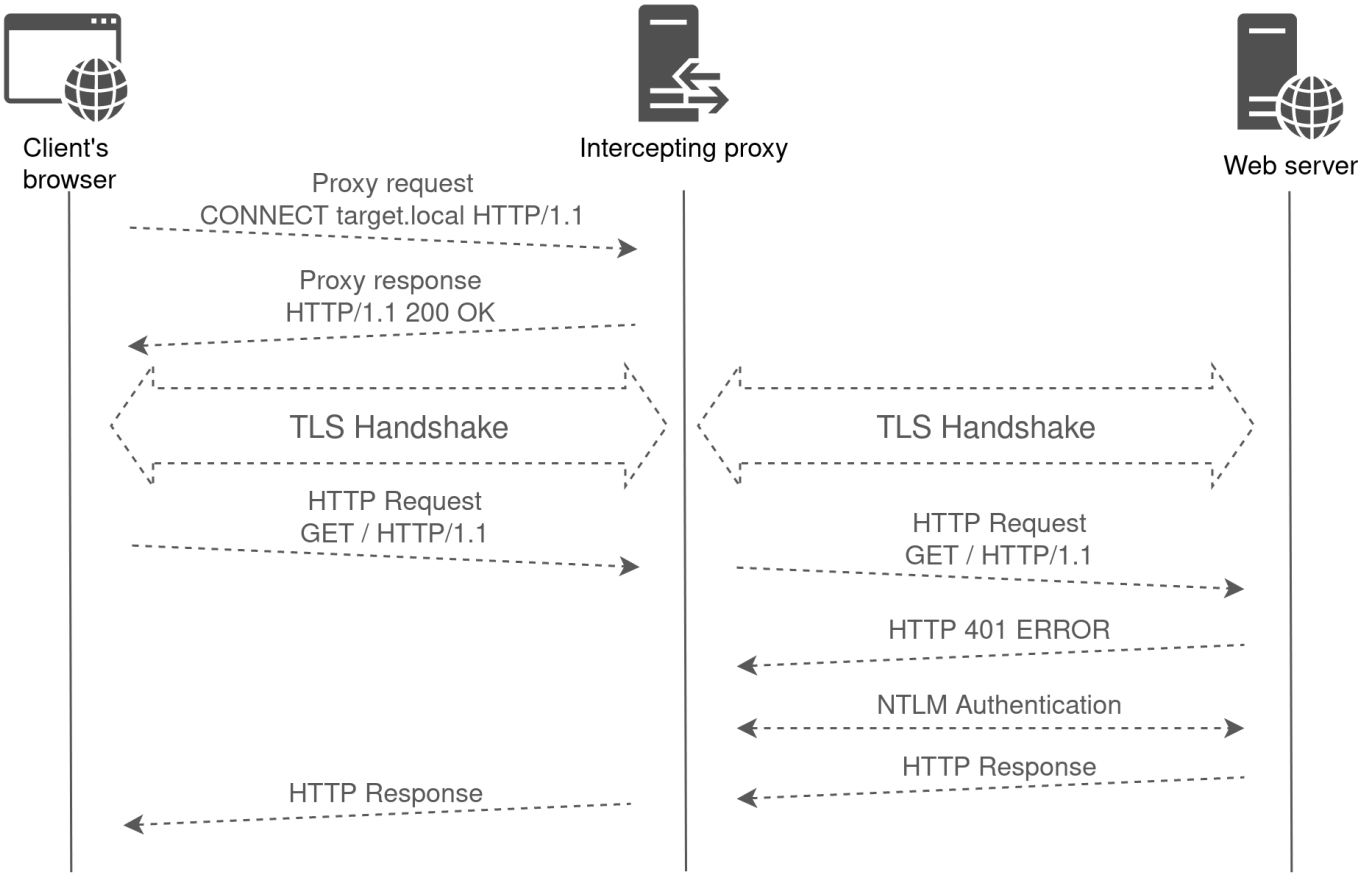
MitM Proxy – Prox-Ez ("prox easy")

- **How?**
  - TLS interception



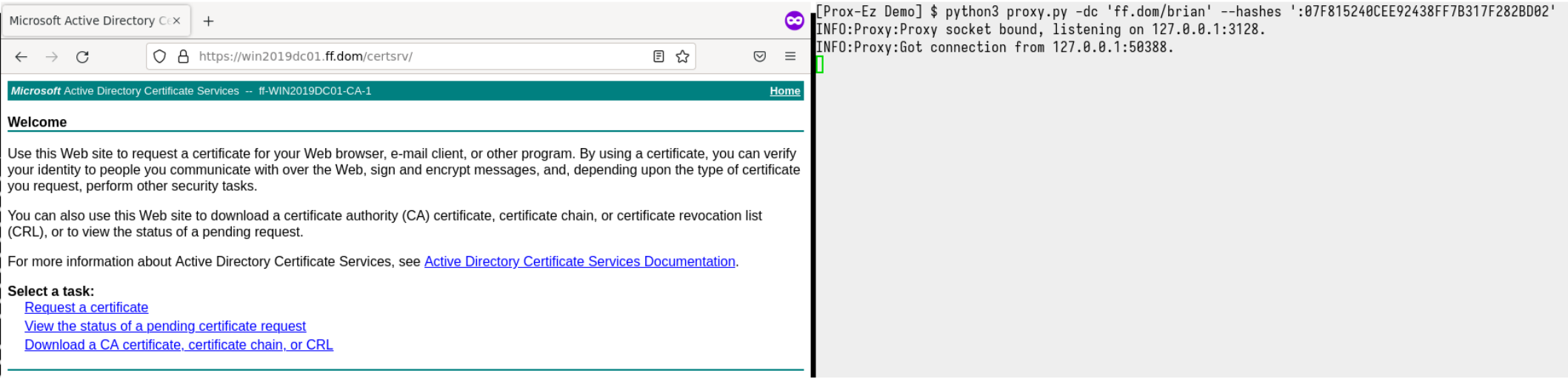
MitM Proxy – Prox-Ez ("prox easy")

- **How?**
  - TLS interception



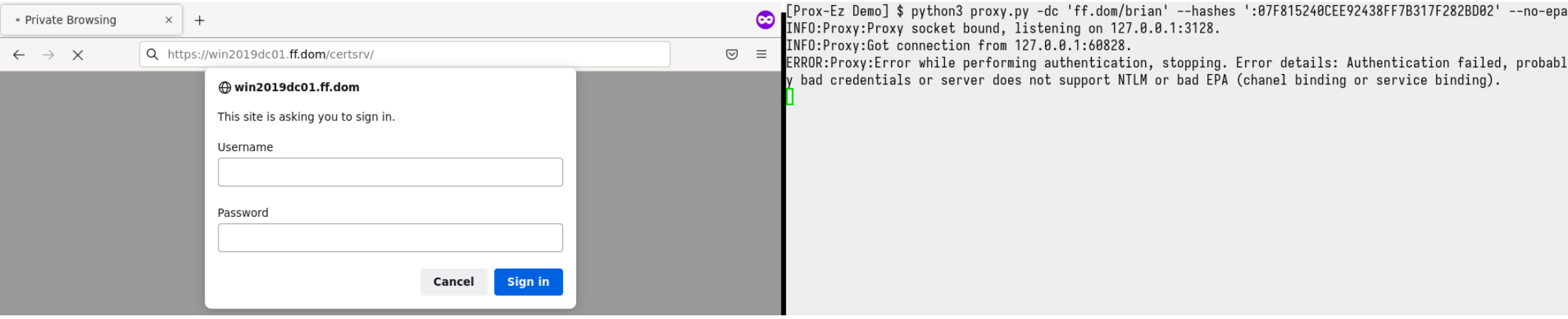
MitM Proxy – Prox-Ez ("prox easy")

- Demo



MitM Proxy – Prox-Ez ("prox easy")

- Demo

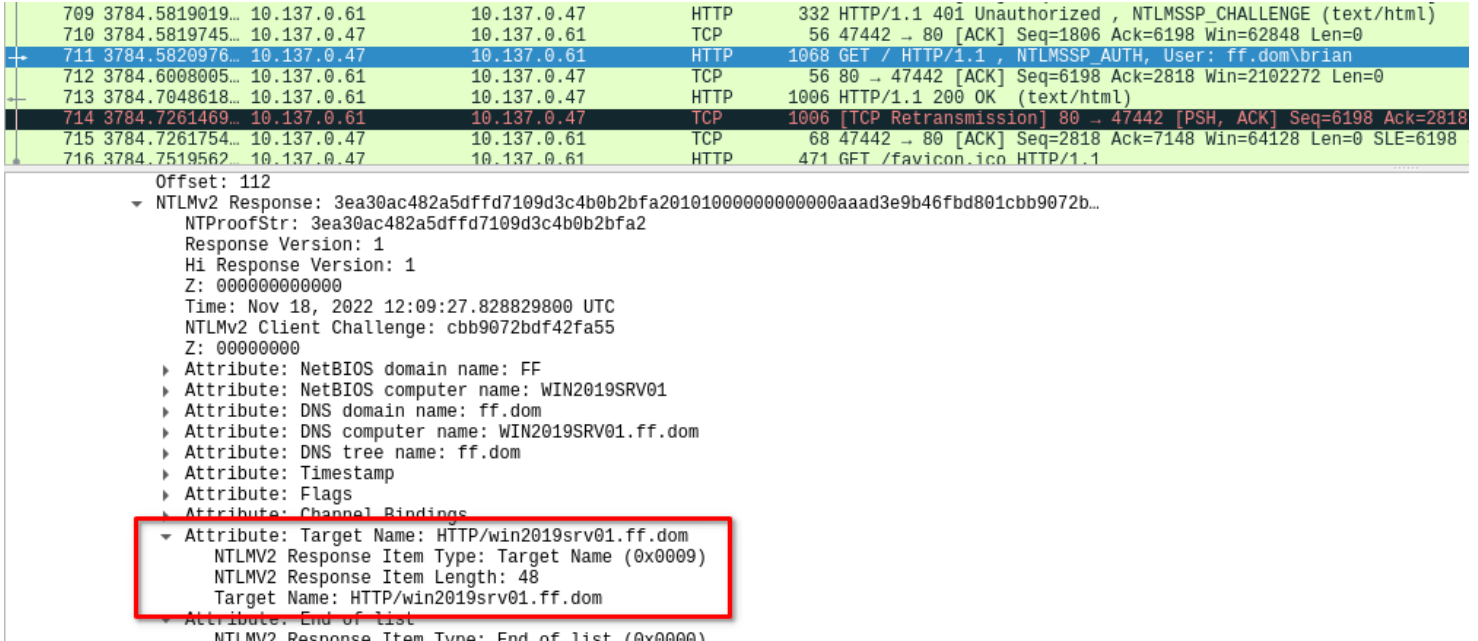


NTLM EPA – Service Binding

- **Service Binding**
  - Channel Binding requires TLS, now what about plain HTTP?
    - (Don't do plain HTTP)
  - Microsoft implemented a new protection
    - Service Binding
  - Same objective as Channel Binding → Prevent MitM attacks

NTLM EPA – Service Binding

- **Service Binding**
  - New attribute in the NTLM *AUTH* message
    - Identifies the targeted resource



NTLM EPA – Service Binding

- **Service Binding**
  - New attribute in the NTLM *AUTH* message
    - Identifies the targeted resource
    - Taken from the browser URL

NTLM EPA – Service Binding

- **Service Binding**
  - New attribute in the NTLM *AUTH* message
    - Identifies the targeted resource
    - Taken from the browser URL
  - If the authentication targets another server than the one receiving the authentication → denied access

NTLM EPA – Service Binding

- **Service Binding**
  - The web server needs to be configured with the proper SPNs
    - No implicit SPN
    - All the alternative DNS records
  - Bad integration in IIS
    - No graphical option
    - Manual modification of *C:\Windows\System32\inetsrv\Config*

NTLM EPA – Service Binding

Service Binding

```
# C:\Windows\System32\inetsrv\Config
<location path="Default Web Site">
  <system.webServer>
    <security>
      <authentication>
        <windowsAuthentication enabled="true" useKernelMode="false">
          <providers>
            <clear />
            <add value="NTLM" />
          </providers>
          <extendedProtection tokenChecking="Require" flags="Proxy,ProxyCohosting">
            <spn name="HTTP/win2019srv01.ff.dom" />
          </extendedProtection>
        </windowsAuthentication>
        <anonymousAuthentication enabled="false" />
      </authentication>
    </security>
  </system.webServer>
</location>
```

NTLM EPA – Service Binding

Service Binding

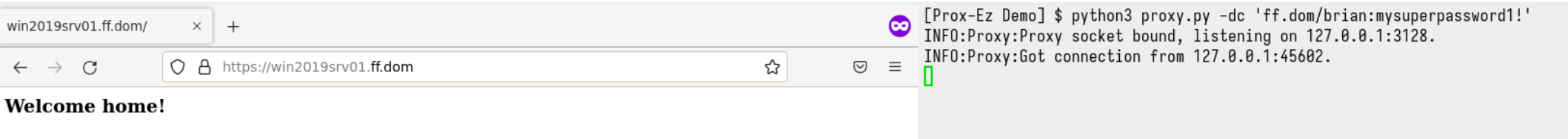
flags options	Behavior	Remark
Empty / None	Only verify CBT	HTTP is not protected ; HTTPs is protected
Proxy	Only verify SPN	HTTP is not working (no authentication possible) ; HTTPs is protected
Proxy,ProxyCohosting	Only verify SPN	Both HTTP and HTTPs are protected and work
Proxy,NoServiceNameCheck	Does not verify anything	HTTP is not working (no authentication possible) ; HTTPs is not protected but a SPN has to be provided (any value)
Proxy,ProxyCohosting,NoServiceNameCheck	Does not verify anything	Both HTTP and HTTPs are not protected (no SPN required)

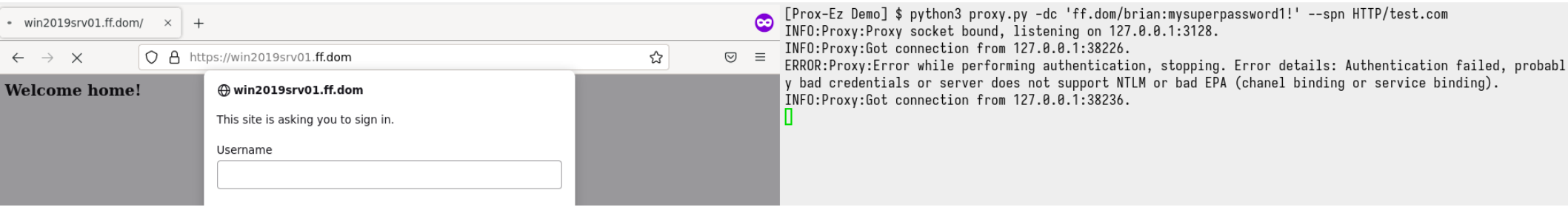
NTLM EPA – Service Binding

- **Service Binding**
  - Service Binding configuration is cumbersome
  - Default EPA configuration → Service Binding not enforced
  - Enforced EPA but plain HTTP available → vulnerable to MitM attacks

MitM Proxy – Prox-Ez ("prox easy")

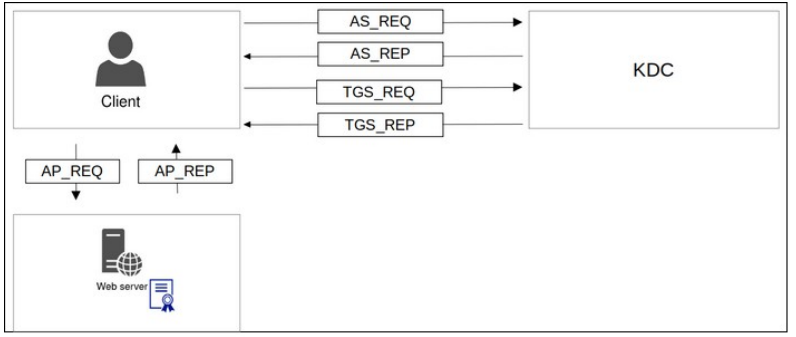
- **Even if not widely used**
  - Prox-Ez implements EPA-Service binding





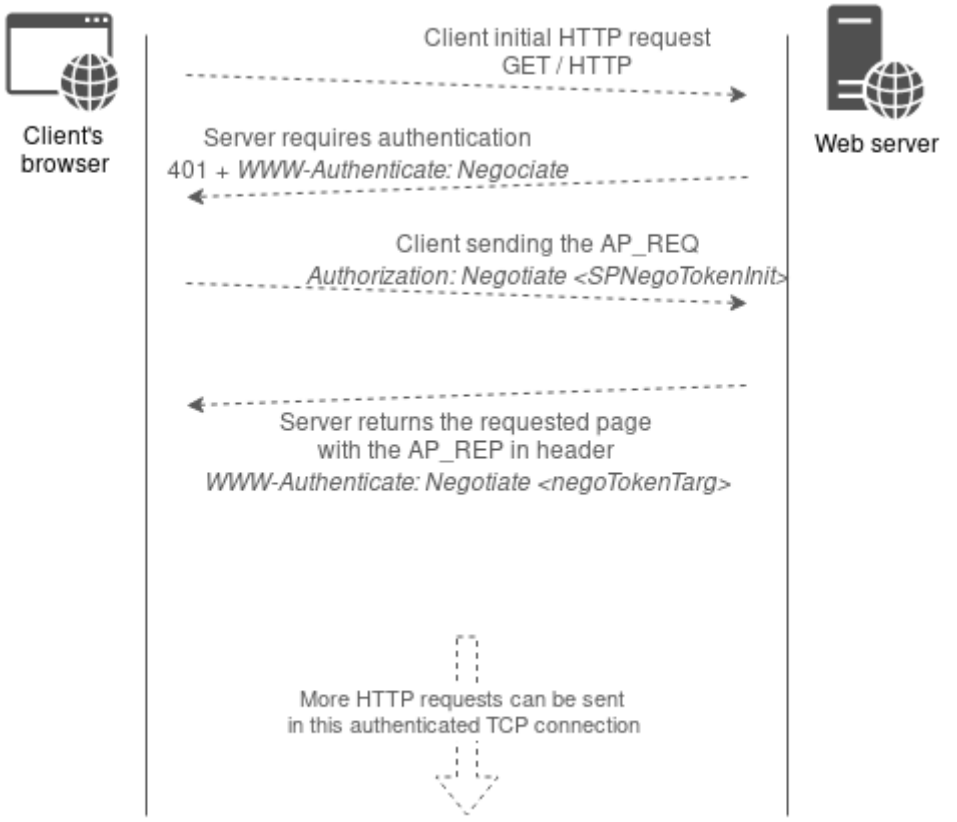
Kerberos

- **Why Kerberos ?**
  - Microsoft recommend enabling EPA as primary mitigation against relay attack (such as PetitPotam)
  - In addition, disable NTLM and replace it by Kerberos
  - Kerberos feature “Mutual Authentication”



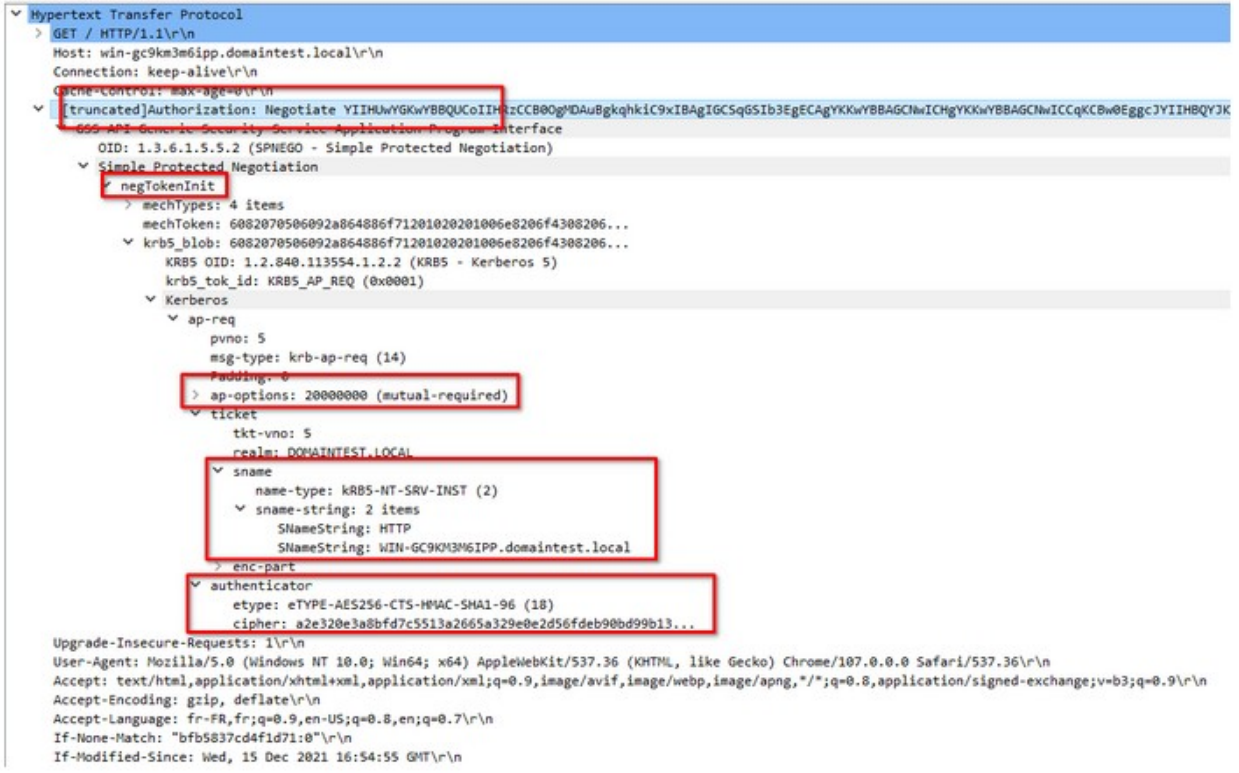
Kerberos

- **...over HTTP?**
  - Similar to NTLM
  - The client sent the AP_REQ in a specific header



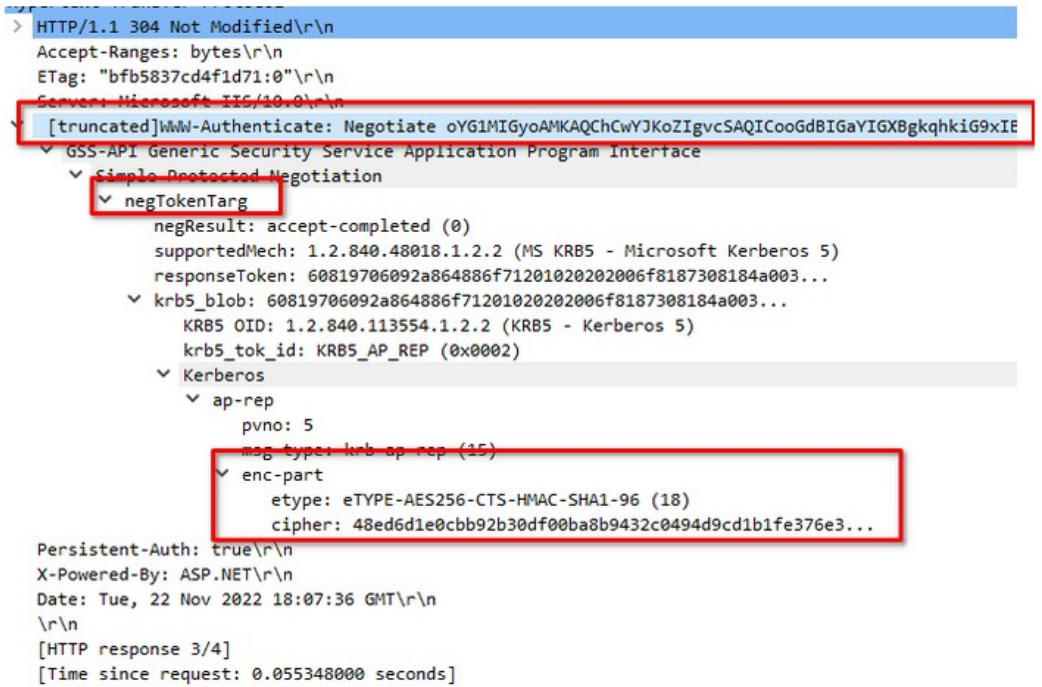
Kerberos

- Let's have a closer look



Kerberos

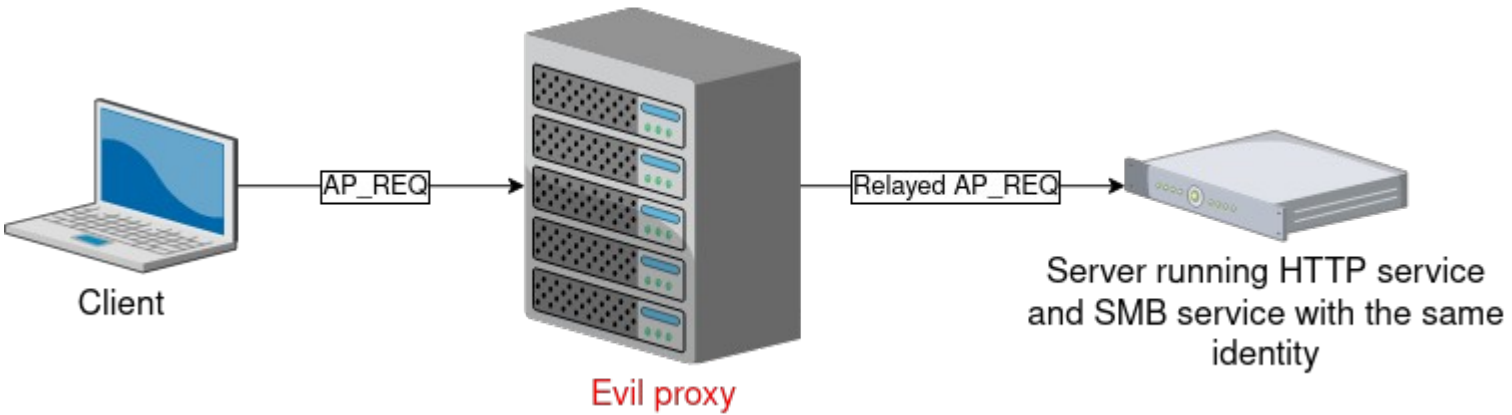
- Let’s have a closer look



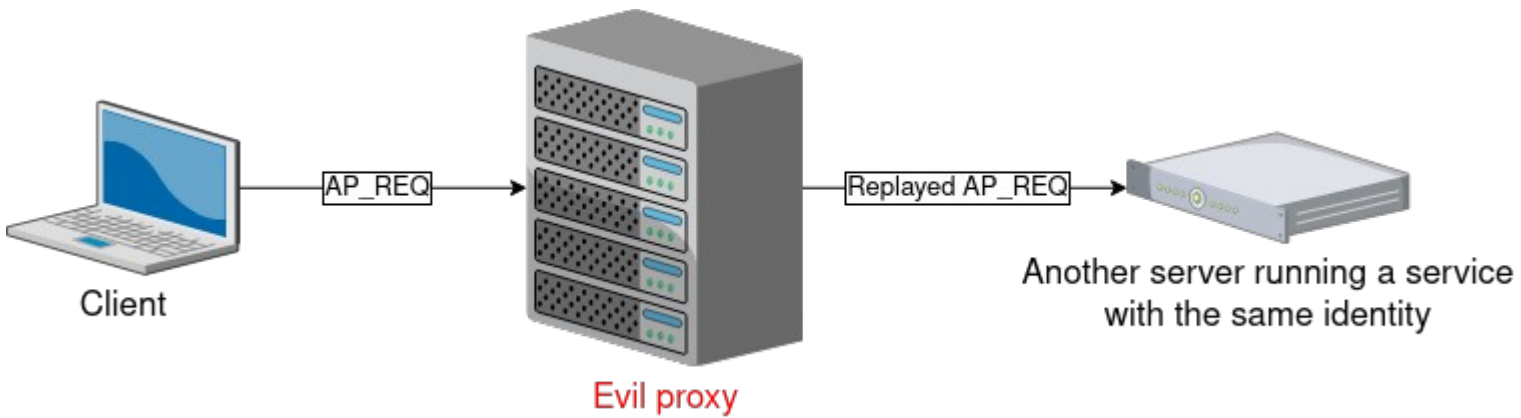
Kerberos

- **Security overview**
  - Two security measures to prevent replay attack
    - AP_REQ contain a timestamp : <5min
    - Host stores a MD5 hash of each AP_REQ : KRB_AP_ERR_REPEAT
    - AP_REQ contains SPN of the service : Not verified

- **Security overview**
  - Relay on a server using the same identity



- **Security overview**
  - Replay on another server using the same identity



Kerberos

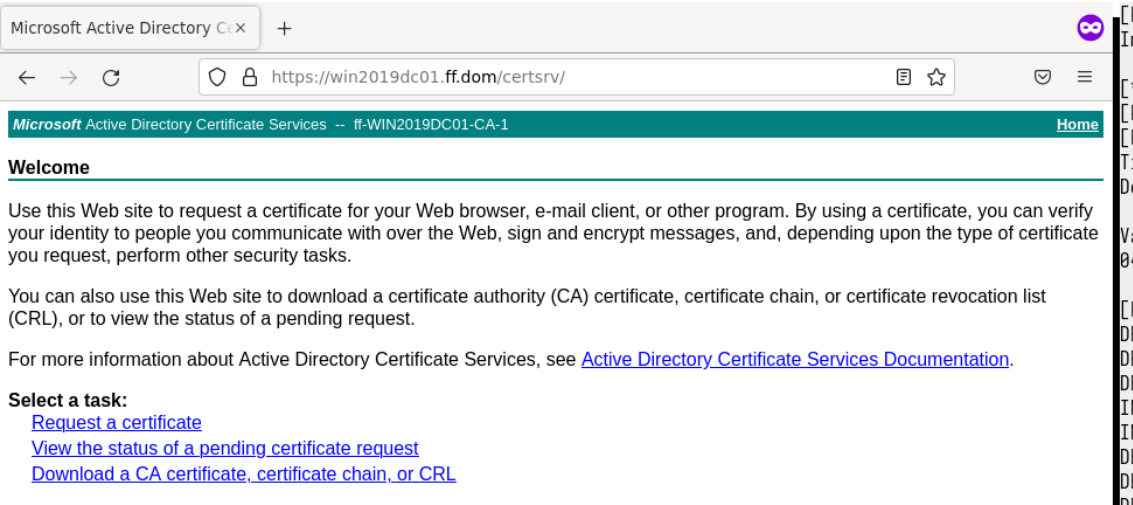
- **Why do we need a proxy**
  - Still not supported by many clients (Firefox, …)
    → No authentication possible if Kerberos is enforced
  - How to use our tools against Kerberos protected websites?
    - BurpSuite
    - Certipy
    - …

MitM Proxy – Prox-Ez ("prox easy")

- **Prox-EZ implement Kerberos authentication**
  - Standard user/password capabilities
  - Pass-the-ticket capabilities (from TGT or ST)
  - Overpass-the-hash capabilities (from the NT hash)

MitM Proxy – Prox-Ez ("prox easy")

- Demo




```
[Prox-Ez Demo] $ getTGT.py 'ff.dom/brian:mysuperpassword1!'
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Saving ticket in brian.ccache
[Prox-Ez Demo] $ export KRB5CCNAME=brian.ccache
[Prox-Ez Demo] $ klist
Ticket cache: FILE:brian.ccache
Default principal: brian@FF.DOM

Valid starting       Expires              Service principal
04/19/2023 16:27:42  04/20/2023 02:27:42  krbtgt/FF.DOM@FF.DOM
        renew until 04/20/2023 16:27:42
[Prox-Ez Demo] $ python3 proxy.py -dc 'ff.dom/brian' -k -d
DEBUG:Proxy:Entered proxy, creating sockets.
DEBUG:Proxy:Proxy socket created.
DEBUG:Proxy:Binding proxy socket.
INFO:Proxy:Proxy socket bound, listening on 127.0.0.1:3128.
INFO:Proxy:Got connection from 127.0.0.1:43272.
DEBUG:Proxy:Creating new process.
DEBUG:Proxy:Handling connection.
DEBUG:Proxy.Client<->ProxyHelper:Creating new ClientToProxyHelper
DEBUG:Proxy.Client<->ProxyHelper:Our state: IDLE; their state: IDLE
DEBUG:Proxy.Client<->ProxyHelper:Received

        CONNECT win2019dc01.ff.dom:443 HTTP/1.1
        user-agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
        proxy-connection: keep-alive
        connection: keep-alive
        host: win2019dc01.ff.dom:443
```

MitM Proxy – Prox-Ez ("prox easy")

- **Available on GitHub:**
  - *https://github.com/synacktiv/Prox-Ez*
  - PR & issues are welcome

README.md

Prox-Ez: The Swiss Army Knife of HTTP auth

This HTTP proxy handles all HTTP authentications on your behalf.

It supports NTLM EPA (channel binding and service binding), kerberos, pass-the-hash, overpass-the-hash (pass-the-key) and pass-the-ticket (TGT and TGS).

SYNACKTIV

Any question?

Linked articles:
https://www.synacktiv.com/publications/dissecting-ntlm-epa-with-love-building-a-mitm-proxy.html
https://www.synacktiv.com/publications/a-study-on-windows-http-authentication-part-ii.html

https://www.linkedin.com/company/synacktiv
https://twitter.com/synacktiv
Our publications: https://synacktiv.com