The workstation account, netlogon schannel and credentials

SambaXP 2018
Göttingen

Volker Lendecke

Samba Team / SerNet

2018-06-06

Why this talk?

- To me, NETLOGON and schannel were a big mystery
- I could never remember what kind of key is used when, what can be shared where, what needs to be locked how.
- In 2017, a customer asked me to optimize the winbind NETLOGON client for a cluster
- A deep-dive into cli_netlogon.c and netlogon_creds_cli.c was due
- The results:
  - Some serious optimizations for a clustered environment
  - Me understanding the data structures ⇒ this talk
- For all the fine crypto details, ask Metze :-)

Active Directory Membership

- Active Directory is Microsoft's central user database
  - Successor to NT4-based Security Account Manager (SAM)
- Member workstations and servers delegate authentication and authorization to the domain
- Multiple requirements for crytpgraphy
- Workstations need to trust the Domain Controllers (DCs)
  - Rogue DC could fake local root/Administrator to members
- User details need to be encrypted
  - Privacy requirements, offline attacks
  - Authentication yields user session key material

Symmetric Cryptography for Membership

- Trust between domain and members based on a shared secret
- Every member holds a workstation account
- Account password used as a shared secret
  - Existing protocols for user password changes can be re-used
- Kerberos
  - Strong authentication protocol based on symmetric crypto
  - Workstation account ⇔ service principal
  - Based on tickets with lifetimes
- NTLM
  - Challenge-Response protocol
  - No tickets, direct queries to the domain
- NETLOGON
  - Wrapper for pass-through NTLM queries

NETLOGON

- Microsoft RPC interface described in [MS-NRPC]
  - https://msdn.microsoft.com/en-us/library/cc237008.aspx
- Specifies the Netlogon Remote Protocol, an RPC interface that is used for user and machine authentication on domain-based networks; to replicate the user account database for operating systems earlier than Windows 2000 backup domain controllers; to maintain domain relationships from the members of a domain to the domain controller, among domain controllers for a domain, and between domain controllers across domains; and to discover and manage these relationships.

NETLOGON Secure Channel Setup

- Having a workstation account enables the trusting workstation to establish a NETLOGON secure channel to DCs of the trusted domain
- RPC-calls used to establish a secure channel:
  - ServerReqChallenge() and ServerAuthenticate() are used for challenge/response authentication
- Both calls are used on an unauthenticated, plain-text RPC connection
- Result from a successful ServerAuthenticate:
  - struct netlogon_creds_CredentialState
  - librpc/idl/schannel.idl
  - Stored in netlogon_creds_cli.tdb (client) and schannel.tdb (server)
- netlogon_creds_cli_auth_send/recv() in libcli/auth/netlogon_creds_cli.c do the ReqChallenge/Authenticate steps.

Using netlogon_creds_CredentialState

- Credentials for encrypted bind to NETLOGON rpc service
  - Custom DCERPC authentication type (auth_type=68, schannel)
  - DCERPC bind only passes domain and computer name
  - Session key from netlogon_creds_CredentialState used like a temporary password and sign/seal crypto seed
  - Once "logged in" to SCHANNEL, netlogon_creds_CredentialState is no longer used for this purpose
- Functions using netr_Authenticator
  - netr_LogonSamLogon[WithFlags](), netr_ServerPasswordSet[2](), netr_LogonGetDomainInfo(), netr_GetForestTrustInformation() and others.
  - The netr_Authenticator implies a global (!) sequence and ordering, thus an exclusive lock on netlogon_creds_CredentialState required

Scaling authentication

- netr_LogonSamLogon() and netr_LogonSamLogonWithFlags() use the netr_Authenticator
  - Limited to one request in-flight globally
  - Exclusive lock on netlogon_creds_CredentialState across SamLogon
  - Called the netlogon credential chain in lkcl's book
  - Designed to prevent session highjacking
- With a secure (signed and encrypted) transport, this is no longer necessary
- Schannel-protected Netlogon RPC can use netr_LogonSamLogonEx(), which avoids the netr_Authenticator
- Multiple connections to a DC have multiple SamLogonEx in flight

Implementation issues

- Requirement: Clustered exclusive and shared locks
- dbwrap only implements exclusive locks
- g_lock on top of dbwrap implements shared and exclusive locks
    - netlogon_creds_cli.tdb entries are protected by a g_lock
    - Two tdb files involved
- g_lock can now store data
    - We could implement netlogon_creds_cli.tdb directly using g_lock code
- Let's look at some code

Questions?

vl@samba.org / vl@sernet.de

http://www.sambaxp.org/