WINDOWS AUTHENTICATION DESIGN

SOLUTION BRIEF

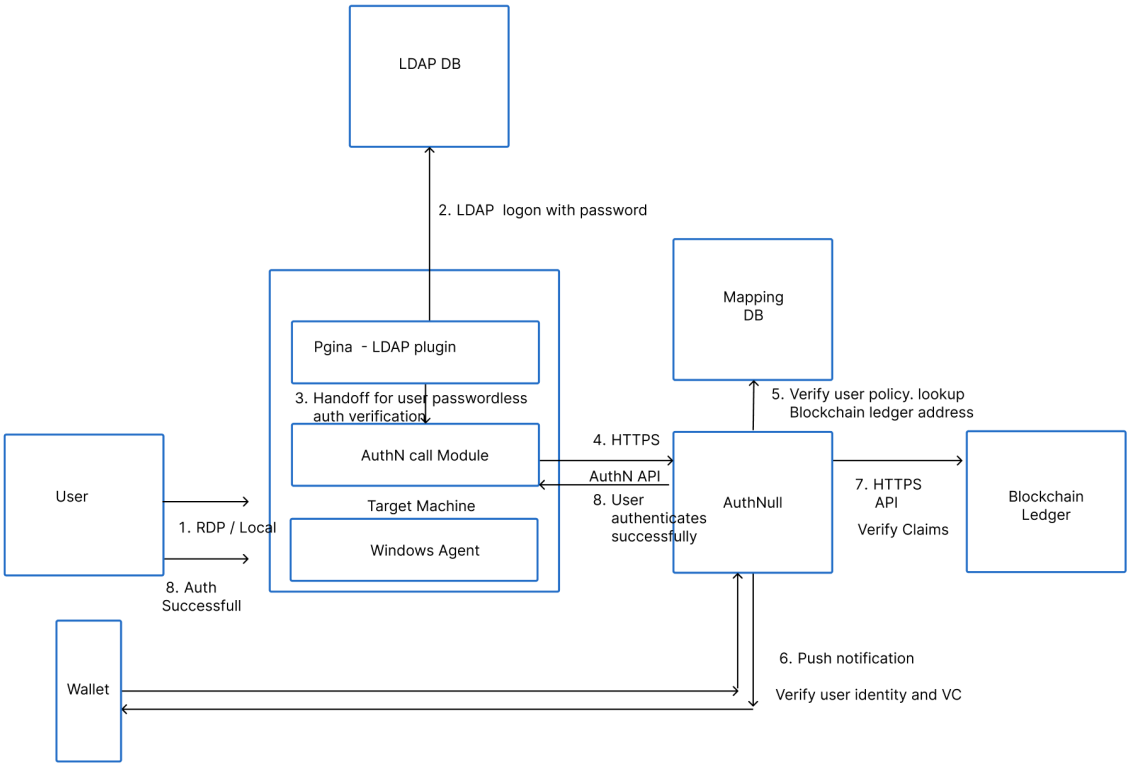
AuthNull

Overview

This document provides an overview of the AuthNull's authentication for interactive and non-interactive / m2m / service account authentication on Linux infrastructure

Use case #1: Interactive authentication

Logical architecture and flow

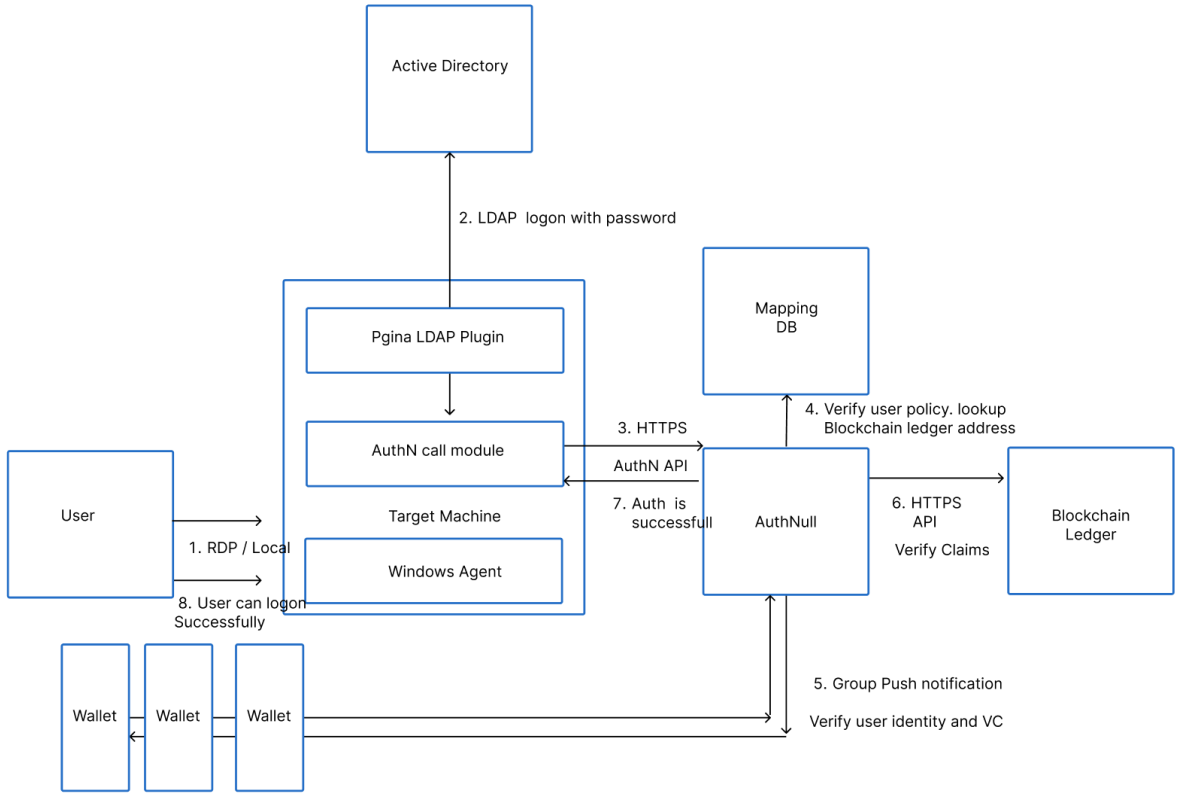


Steps	What happens
Step #1	User attempts to connect to target machine using RDP or local auth
Step #2	Pgina does windows authentication with a LDAP connector to verify the user identity on an LDAP directory (LDAP / TCP). When successful this moves to step #3. If not successful – authentication will be denied.
Step #3	Custom LDAP authenticator then makes AuthNull call to verify Passwordless claim

Step #4	SSP server gets called to verify interactive authentication policy from the mapping db.
Step #5	Mapping DB is used to look up authentication policy and verify - Does this user actually have access rights? Does an interactive policy exist? - Lookup Address of blockchain ledger.
Step #6	User / Owner of account gets a push notification on wallet - User submits presentation submission (credential signed by private key) if he accepts the authentication request, or deny the request if they think someone else is accessing the account. - This PS can be verified using users public key - Additionally, it is converted to a hash with current salt and random string for the day. - This can be considered as user Hash
Step #6	The blockchain ledger hash is looked up from the address from Mapping DB for this user authentication policy. This hash is compared to the computed hash retrieved from the wallet. If both the hashes match, it further verifies that the user has the correct credentials using blockchain
Step #7	User authenticates successfully.

Use case #2: Shared user interactive authentication

Logical architecture and flow

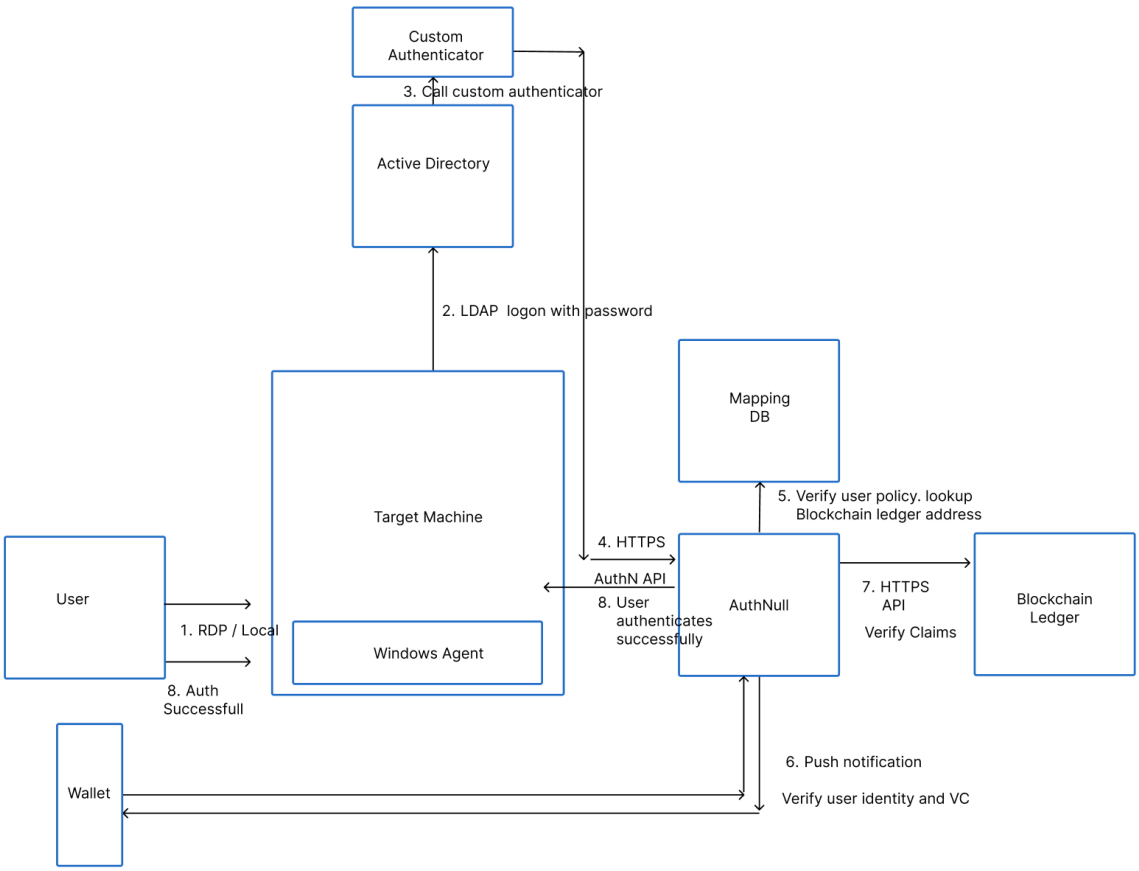


Steps	What happens
Step #1	User attempts to connect to target machine using RDP or local auth
Step #2	Pgina does windows authentication with a LDAP authentication to verify the user identity on an LDAP directory (LDAP / TCP) along with password validation. When successful this moves to step #3. If not successful – authentication will be denied.
Step #3	AuthN module (part of LDAP plugin) calls AuthNull service to validate Passwordless claim
Step #4	SSP /AuthNull server mapping db lookup Mapping DB is used to look up authentication policy and verify

	- Does this user actually have access rights? Does an interactive policy exist? - Lookup Address of blockchain ledger.
Step #5	User / Owner (s) of account gets a push notification on wallet - Any single user / wallet owner can submit presentation submission (credential signed by private key) if he accepts the authentication request, or deny the request if they think someone else is accessing the account. - This PR can be verified using users public key - Additionally, it is converted to a hash with current salt and random string for the day.
Step #6	The blockchain ledger hash is looked up from the address from Mapping DB for this user authentication policy. This hash is compared to the computed hash retrieved from the wallet. If both the hashes match, it further verifies that the user has the correct credentials using blockchain -
Step #7	User is able to authenticate successfully.
Step #8	User is logged on to the target machine successfully.

Use case #3: “Agentless” authentication on client machines

Logical architecture and flow



Steps	What happens
Prerequisite	Admin (who owns m2m credential) first delegates credentials to a blockchain ledger and writes the hash in mapping db, along with address of ledger. Each target authentication machine and its unique IP also have a machine key, and a policy stored in the mapping db.
Step #1	Privileged User initiates RDP connection to target machine

Step #3	Active directory calls custom authenticator Custom authenticator calls Authnull
Step #4	Custom Authenticator calls AuthNull control plane to verify the policies
Step #5	SSP server is called to verify the user authentication. 1. SSP server checks to find if there's an M2M policy in Mapping db – yes. This is a M2M policy and a hash, with a address of the blockchain. 2. SSP checks Mapping db for active **interactive sessions** from source machine The mapping is done using source ip, source remote port, source username, and these values are typically also found on the destination machine using pam_exec, during the course of the session. 3. If this mapping session db has a session hash, along with an interactive user then this session is considered an interactive user. 4. SSP finds an active user logged on who initiated this authentication and reverts to evaluating this request as an interactive authentication.
Step #8	Owner of the wallet is notified with a push notification to accept or deny the authentication request. If the owner of the user denies the request, authentication will fail.
Step #7	The blockchain ledger hash is looked up from the address from Mapping DB for this user authentication policy. This hash is compared to the computed hash retrieved from the db. If both the hashes match, it further verifies that the m2m authentication is correct has the correct credentials using blockchain
Step #8	User is able to authenticate successfully.

Other Pertinent use cases

1. Rotate local root password and share to one or more wallets - Linux
2. Root bypass authentication (without MFA) - Linux

Contact Us to Learn More

Get in touch with us to learn more at sales@authnull.com or visit https://authnull.com/agentless.