RSA CONFERENCE EUROPE 2007
22-24 October | ExCel London | United Kingdom

Mechanics of User Identification and Authentication

Dobromir Todorov
BT Global Services/
ITCE

Overview

- Basics of User Identification and Authentication
- Security Servers and Databases
- User Authentication Protocols
- Authentication Federation
- Summary
- Questions and Answers

* Basics of User Identification and Authentication

- User Identification and Authentication
- Compliance Requirements
- Authentication Components
- Authentication Levels
- AAA
- Threats to User I&A
- Enterprise Challenges

User Identification

- The process of identifying a user (security principal) within an Information System
- Unique name for the user (ID)
- Uniqueness within the information system or within the Internet
- A single user object may have multiple associated IDs (however, each ID must point to exactly one user object)

Examples:

- NTLM: INS\TodorovD
- Kerberos/SMTP/SIP (RFC 733/822): dobromir.todorov@bt.com
- X.400 (X.509) Certificate: Subject/Alt Subject: CN=Dobromir Todorov, OU=INS, O=BT Global Services
- Biometric: Retina/Fingerprint (also provides Recognition)

User Authentication

Authentication is the **process** of providing **proof of identity:**

- A user applies for access and provides identity information
- The system requests proof of identity
- The user provides credentials
- Authentication succeeds or fails

Authentication always comes before authorisation

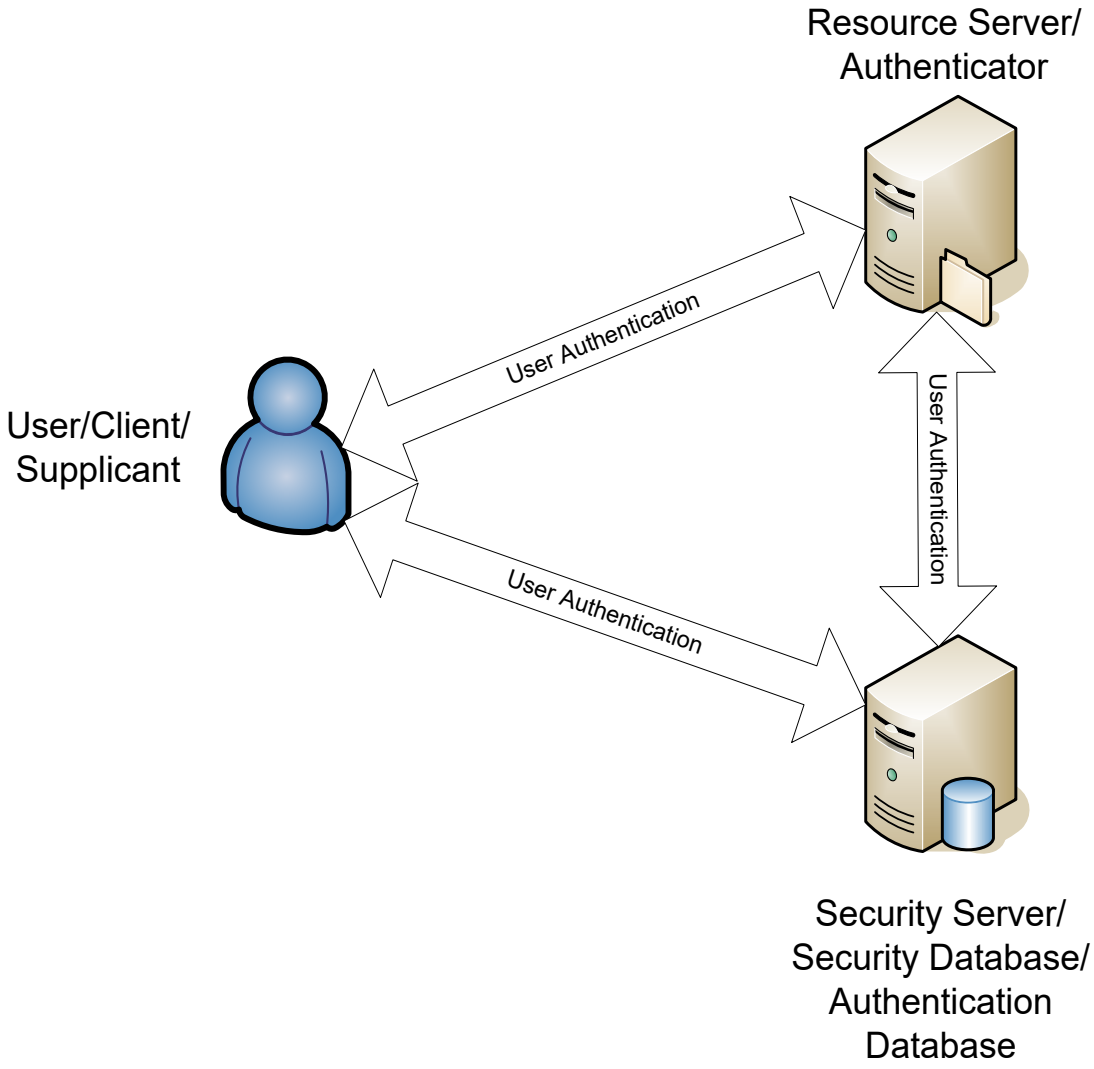
Compliance Requirements

- User Identification and Authentication is required:
  - EU Directives (1999/93/EC – Electronic Signatures & 2000/31/EC - Electronic Commerce)
  - SOX (Section 302 - Internal Controls)
  - Basel II – Operational Risk – *"the risk of loss resulting from inadequate or failed internal processes, people, and systems, or from external events"*
  - HIPAA §164.312(c)(2)

    "(d) Standard: Person or entity authentication. Implement procedures to verify that a person or entity seeking access to electronic protected health information is the one claimed."
- At the same time, privacy will also be required for public systems:
  - EU Privacy Directives (directive 95/46/EC, etc)

    "Whereas, the principles of protection must apply to any information concerning an identified or identifiable person; whereas, to determine whether a person is identifiable, account should be taken of all the means likely reasonably to be used either by the controller or by any other person to identify the said person…"
  - HIPAA
- Balance must be found

User Authentication - Components



Credentials (proof of identity)

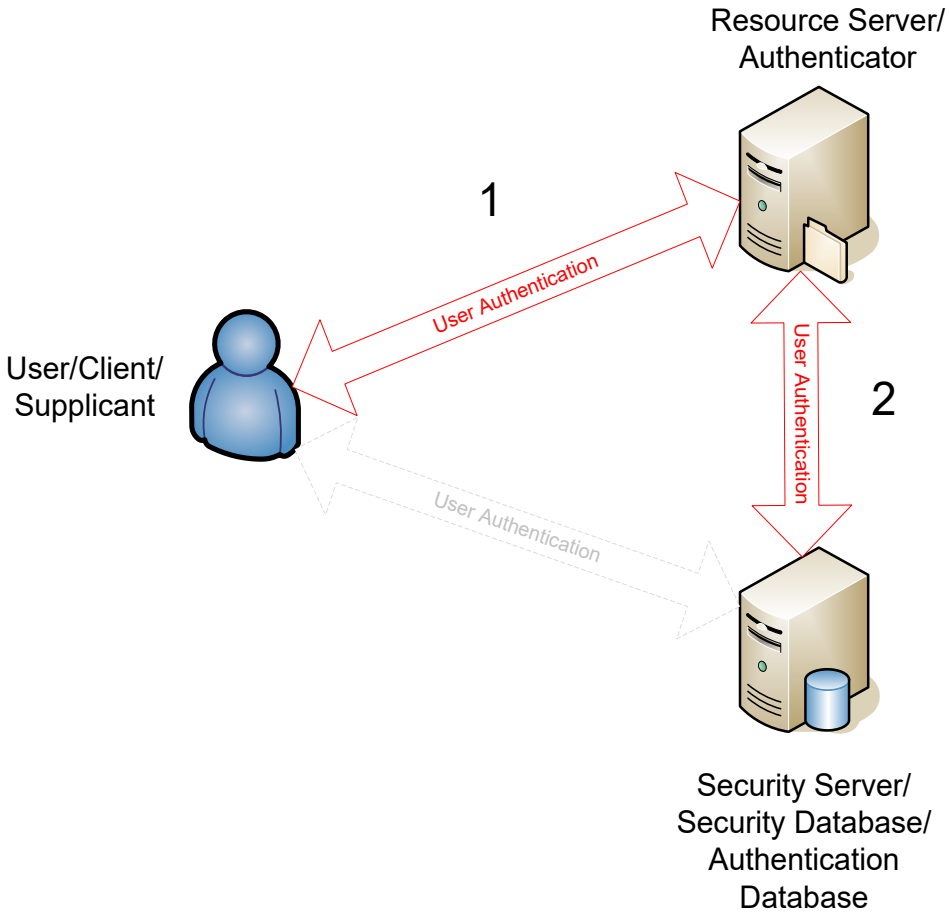
- Something you know
  - Static Password (including PIN Code)
  - Private Key
- Something you have
  - Authentication Token
  - Smart Card (with private key or other security information)
- Something you are - biometrics
  - Static (Physiological patterns)
  - Dynamic (Behavioral)
- *Multiple factors of authentication*
  - *Dual-factor authentication*
  - *Triple-factor authentication*

Credential Comparison

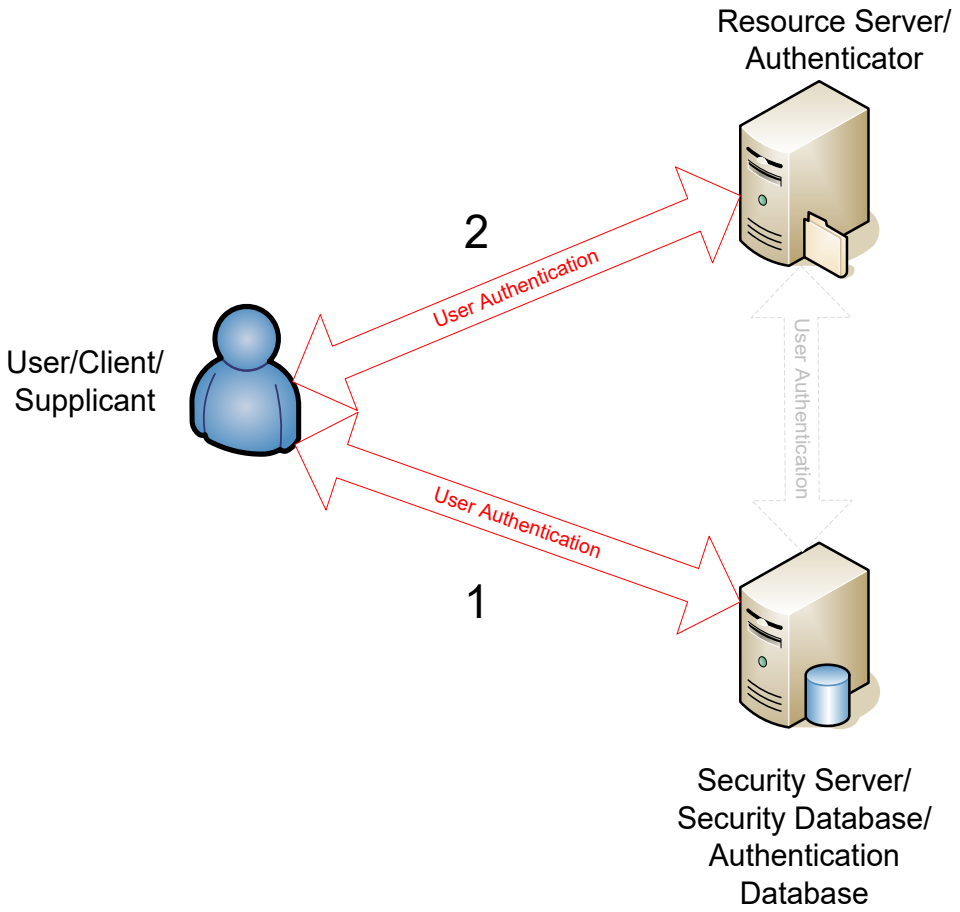
Credential Type	Strength *	Popularity *
Static Password	1	4
One Time Password (OTP)	2	2
Certificate Based Authentication	3	3
Biometric Authentication	4	1
Multiple Factor Authentication	4	2

* 1 – lowest | 4 - highest

Direct Network Authentication



Indirect (Ticket Based) Authentication



Authentication Layers

When accessing information resources, identification and authentication may take place at different levels independently:

- **Physical Access** – access cards, passes/badges, ID cards
- **Network Infrastructure Access** – local wired, local wireless, remote, VPN
- **Access to Applications and Services** – directory services, e-mail, ERP applications, etc

AAA

- **Authentication** (I&A) – Identify the user and check user credentials
- **Authorisation** – Once the user has been authenticated, provide granular access to specific information, if the user is granted access to this information; otherwise, deny access
- **Accounting/Auditing** – Log successful and unsuccessful access attempts in the system security log

Delegation and Impersonation on the Server

- **Anonymous access**: Do NOT authenticate/impersonate the user.
- **Identify** the user: Determine the identity and/or specific attributes of the user but do not change the security context. The user may or may not be authenticated.
- **Impersonate**: Identify and authenticate the user. Switch the security context to such using the privileges of the user. However, when accessing other resources on behalf of the user, use a server account.
- **Delegate**: Identify and authenticate the user. Switch the security context to such using the privileges of the user. Use the privileges of the user to access all types of resources.

Threats to User I&A

- Bypassing authentication
- Weak passwords
- Privilege escalation
- Password guessing
- Sniffing network traffic
- Replaying authentication
- Downgrading authentication strength and Man in the Middle
- Session Hijacking and Man in the Middle
- Shoulder Surfing
- Social Engineering
- Keyboard loggers, trojans, viruses, phishing
- Stealing password databases
- Dumpster diving and identity theft

Enterprise Authentication Challenges

- Centralised vs Decentralised Authentication
- Authentication Strength ("…you are as weak as your weakest link")
- Single Sign-on
- Impersonation and Delegation
- Identity Management
- Federated Access to Resources

* Security Server and Database

- Security Database
- User Authentication at the Security Server

Security Database

- Local
  - Delimited Plaintext – UNIX **/etc/passwd** file
  - Structured – Windows NT SAM
  - Structured - Active Directory NTDS DIT files (locally on DC)
- Network
  - NIS/NIS+ - download **/etc/passwd** content from a server
  - Active Directory – access using Kerberos/NTLM over the network (where local DB is stored on the DC)
  - LDAP – either query LDAP or attempt authentication with user credentials (impersonate)
  - Authentication Passthrough – impersonate user and attempt network authentication against a service provided by the security server
- None
  - Certificate Authority – only needs a secure store for its private key (can be stored on an HSM or Smart Card as well)
  - Some CAs may provide for key recovery and may have a secure archive storage
  - Trust established from leaves to the top of the PKI hierarchy

User Authentication at the Security Server - 1/4

- Plaintext passwords (or reversibly encrypted passwords)
  - Compare locally stored password with user provided password (plaintext or decrypted)
  - *Examples: PPP PAP, Interactive*

- One Way Function (OWF) – hash
  - Security Server doesn't have the plaintext password
  - Client calculates OWF from the user provided password and submits it to the Security Server
  - Security Server calculates OWF of the user password when the password is set/changed
  - The Security Server compares the locally calculated hash with the client provided hash
  - Marginally more secure than plaintext: OWF is equivalent to password in terms of protection
  - *Examples: NTLM v1/2, MS-CHAP*

User Authentication at the Security Server - 2/4

- Plaintext/OWF Hybrid
  - Security Server stores multiple pre-calculated OWF password hashes – one for each variation in the challenge-response algorithm (for example capital user name, lower case user name, etc)
  - Client calculates OWF from user provided plaintext password
  - Server compares client response with all the server stored OWF-transformed passwords
  - *Examples: PPP CHAP, SASL DIGEST*

User Authentication at the Security Server - 3/4

- One Time Password (OTP)
  - Calculate Response based on initial seed and other parameters (such as time)
  - Compare client response with server calculated response
  - *Example: S/Key, RSA SecurID*

- Encrypted Response/Ticket
  - Security Server has client password, or hash, or certificate
  - Client requests to receive a ticket and specifies protection suite and identity
  - Security Server returns ticket encrypted in the client credentials
  - Only a client that has valid credentials can decrypt the ticket and use it
  - *Example: Kerberos*

User Authentication at the Security Server - 4/4

- Biometrics
  - (Optional) Recognition – search database and identify user
  - Compare biometric data provided with data available in the local database
  - *Examples: Verid+*
- Certificate Authentication
  - Authentication only done once, at the time of enrollment
  - User provides proof of identity – may include driving license, passport, or authentication credentials, or a combination
  - *Examples: OpenSSL, Microsoft CA*

* User Authentication Protocols

- Evolution
- User Authentication at the Security Server
- Ticket-based Authentication Methods
- Trust

The Evolution of User Authentication Protocols

- Clear Text Static Passwords
- One Time Passwords
- Encrypted Credentials
- Challenge-Response
- Ticket Based Authentication
- Federated Identity

Case Study: Clear Text Authentication Protocols - 1/2

- **Problem:** User authentication is required
- **Solution:** Plain text dialogue:
  - Server prompts for authentication (interactive or within protocol data)
  - Server requests username
  - Client sends username
  - Server requests password
  - Client sends clear text (or encoded BUT NOT encrypted) password
- **Examples:** Terminal Line Interactive, PAP, SASL Login/Basic, HTTP Basic, Telnet, SSH Interactive*

Case Study: Clear Text Authentication Protocols - 2/2

- **Attack:**
  - Sniff on the wire between the client and the server
  - Capture the username
  - Capture the password
  - Authenticate to the server (until the password is changed, or account invalidated/disabled)

Case Study: Encrypted Credentials Authentication – 1/2

- **Problem**: Clear Text Authentication provides plaintext usernames and passwords. These need to be hidden from the eye of the attacker
- **Solution**: Encrypt authentication information including credentials with a key that rarely (or never) changes
- **Examples**: WEP (802.11)

Case Study: Encrypted Credentials Authentication – 2/2

- **Attack**:
  - Capture the encrypted authentication information
  - Replay the authentication information to the server
  - No need to know decrypted password as long as encrypted information is available

Case Study: Challenge/Response Authentication – 1/2

- **Problem**: Encrypted credentials still allow an attacker to replay authentication; the attacker does not even need to decrypt the credentials
- **Solution**: Make every authentication session unique by generating a session specific string (challenge) which may be random, or time based but it must be unpredictable and virtually unique. Send the challenge to the client and request the client to encrypt (or generate a hash of) the challenge with the client credentials
- **Examples**: CHAP/MS-CHAPv.1&2, SASL CRAM & Digest

Case Study: Challenge/Response Authentication – 2/2

- Attack 1: Man in the Middle
  - S is the server, C is the client, M is the attacker
  - M connects to S and receives a challenge A
  - M waits for a connection from C to S and intercepts it
  - M sends the challenge A to C pretending to be S
  - C generates the response R based on user credentials and returns it to M (who C thinks is S)
  - M sends the response R to S
  - S grants access for M with the privileges of C
- Attack 2: Known Plaintext attack
  - Attacker is M, client is C
  - M tricks C into accessing a resource on M
  - M sends a challenge to C (M can choose the challenge carefully)
  - C generates a response using the password (and potentially other parameters)
  - M tries to recover C's credentials using cryptanalysis techniques

Authentication Protocol Negotiation

- If client and server support multiple authentication methods, they can negotiate which one they will use
- Client and server must find a common language (authentication protocol)
- SASL provides an abstraction layer for applications to negotiate authentication methods, authenticate clients and protect the communication channel
- PPP Link Control Protocol can facilitate authentication protocol selection
- IKE can negotiate authentication methods
- SPNEGO (MS Negotiate SSP) can provide for authentication protocol selection (typically selection between Kerberos and NTLM)
- SSH provides for different authentication protocols

Case Study: Downgrading Authentication Strength

- Authentication Negotiation is convenient when client and server are running different applications and products
- However, an attacker in a Man in the Middle attack may potentially intercept the channel between the client and the server and request that the client (or the server) be authenticated using a weak authentication method
- Strong authentication methods may not be used at all (bypassed)
- To protect from such attacks, do either of the following:
  - Do not allow authentication using a weak authentication method; only allow authentication using a number of equally strong authentication methods
  - Protect the integrity of the communication channel between the client and the server

Case Study: Protecting the Authentication Channel

- **Problems**:
  - Clients must be able to distinguish between legitimate servers and imposters: server authentication
  - The channel between the client and the server must be encrypted
  - The integrity of the channel between the client and the server must be guaranteed

- **Solution**:
  - Provide a protected channel between the client and the server
  - Provide for peer authentication

- **Examples**: SSL/TLS, SSH, IPSec, NTLM Secure Channel

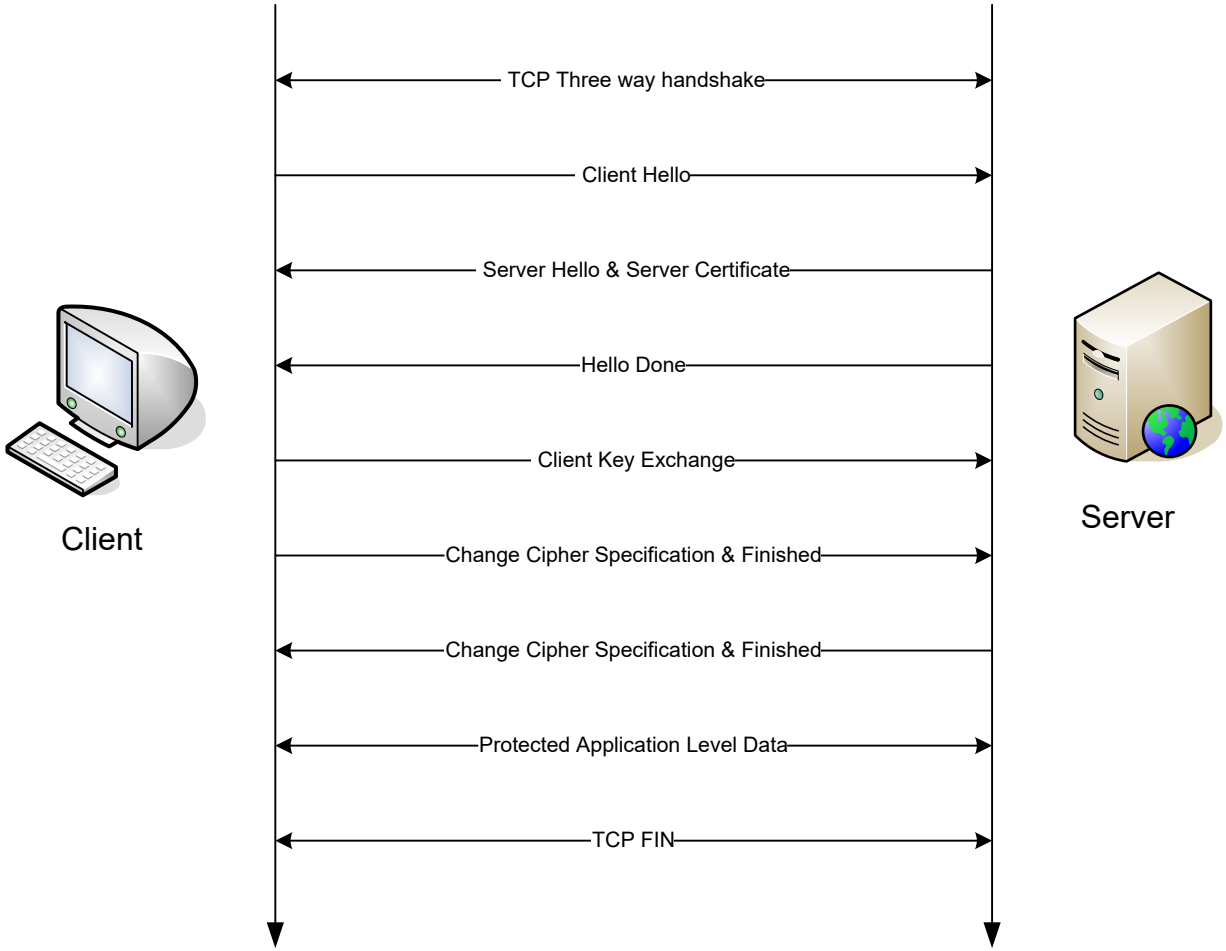
SSL/TLS Channel Protection - 1/2

- SSL = Secure Sockets Layer
- TLS = Transport Layer Security

Both provide:

- Peer authentication (typically based on certificates and associated RSA key pair; other options less popular)
- Peer authentication can identify the server only OR client and server
- Data Integrity by using HMACs – MD5 or SHA-1
- Data Encryption – DES/3DES, RC4, AES

SSL/TLS Channel Protection - 2/2



SSH Channel Protection

- SSH v.1 – Host Keys (pair) and temporary Server Key (pairs)
- SSH v.2 – Host Keys (pair) and session keys

Both Provide:

- Peer authentication (typically uses key pairs but certificates are an option)
- Client authenticates server identity
- Data Integrity by using HMACs – MD5 or SHA-1
- Data Encryption – DES/3DES, RC4, AES, Blowfish, etc

ISAKMP/IKE (IPSec)

- Peer authentication (typically based on certificates and associated RSA key pair, pre-shared keys or Kerberos tickets (Microsoft))
- Peer authentication always identifies and authenticates the identity of both peers
- Data Integrity by using HMACs – MD5 or SHA-1
- Data Encryption – DES/3DES, AES
- Can be used to run clear text authentication mechanisms on top

NTLM Secure Channel

- Used by Windows NT/2000/2003 Domain Members
- Required for NTLM domain authentication (not used for Kerberos authentication)
- Supported for compatibility with older versions
- Domain members establish a Secure Channel with a domain controller upon startup
- Domain member authenticates to domain controller using workstation (or server) account and associated password
- Secure Channel can be encrypted and the integrity can be authenticated
- Subsequent user authentication attempts are sent to domain controller over Secure Channel using MS RPC

Ticket Based Authentication Methods – 1/2

Indirect authentication approaches:

- Kerberos
- Certificate Mapping
- SAML/WS-Security

Ticket Based Authentication Methods – 2/2

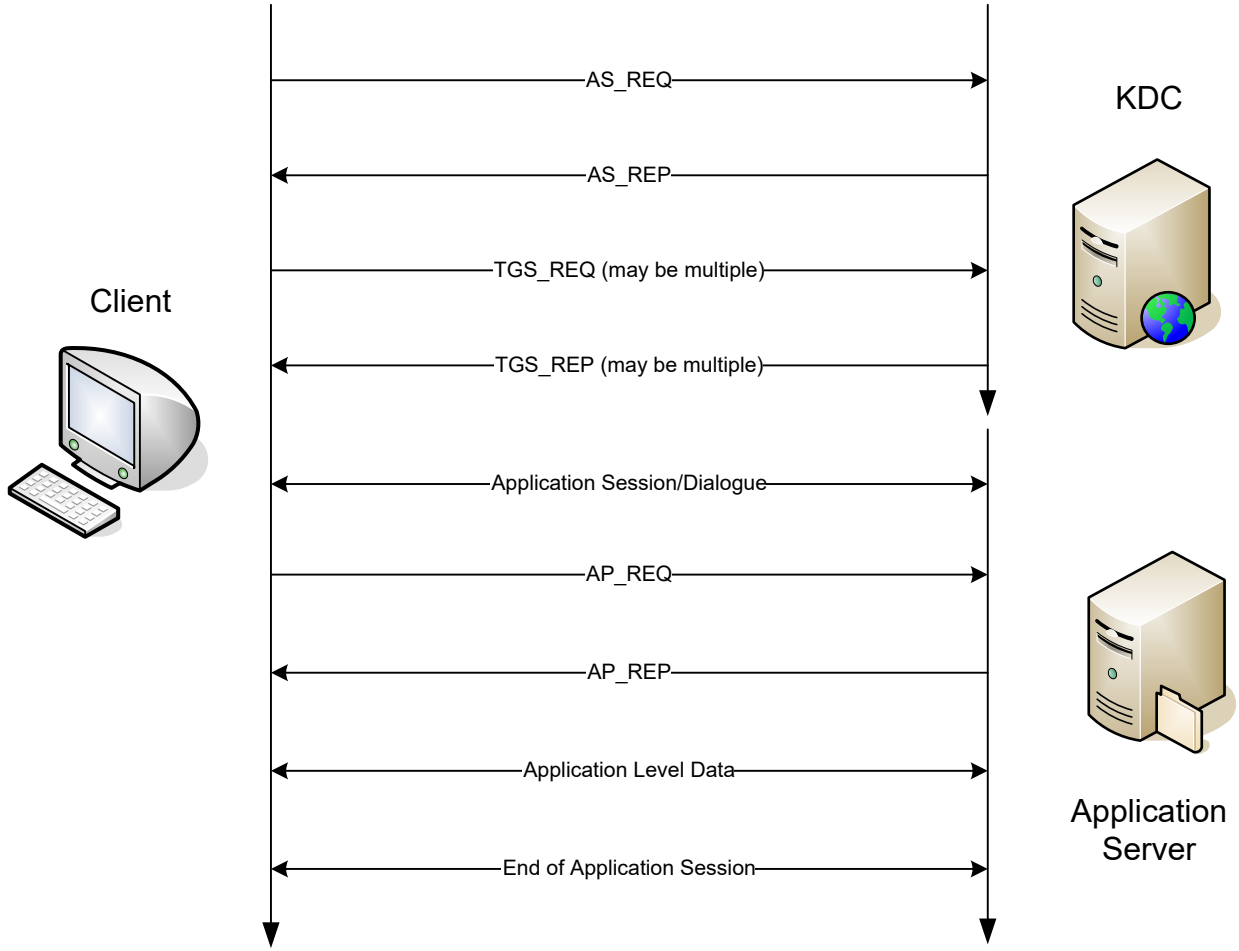
- A trusted authentication provider is required
- Client authenticates to trusted authentication provider first; the client may use any supported set of credentials
- Trusted Authentication Provider issues a ticket to the client
- Client can access resources using the ticket by sending the ticket to a server; client credentials are no longer required to be provided upon access to resources
- Tickets have a Time to Live and then expire
- Tickets may have intended purposes
- Tickets contain client identification and potentially authorisation information

Ticket Based Authentication – Kerberos - 1/2

- Trusted Authentication Provider – Kerberos Distribution Centre (KDC)
- Kerberos client
- Kerberos server (Kerberized service)

- Client Authenticates to KDC – Authentication Service (aka Ticket Granting Ticket – TGT)
- Client requests a ticket for a service from the KDC – Ticket Granting Service (TGS)
- Client authenticates to server and provides ticket (identification and authorisation information) – Application Protocol – AP
- The Server does NOT need to contact the KDC to authenticate the user; all information is contained within the signed ticket

Ticket Based Authentication – Kerberos – 2/2



Ticket Based Authentication – Certificates

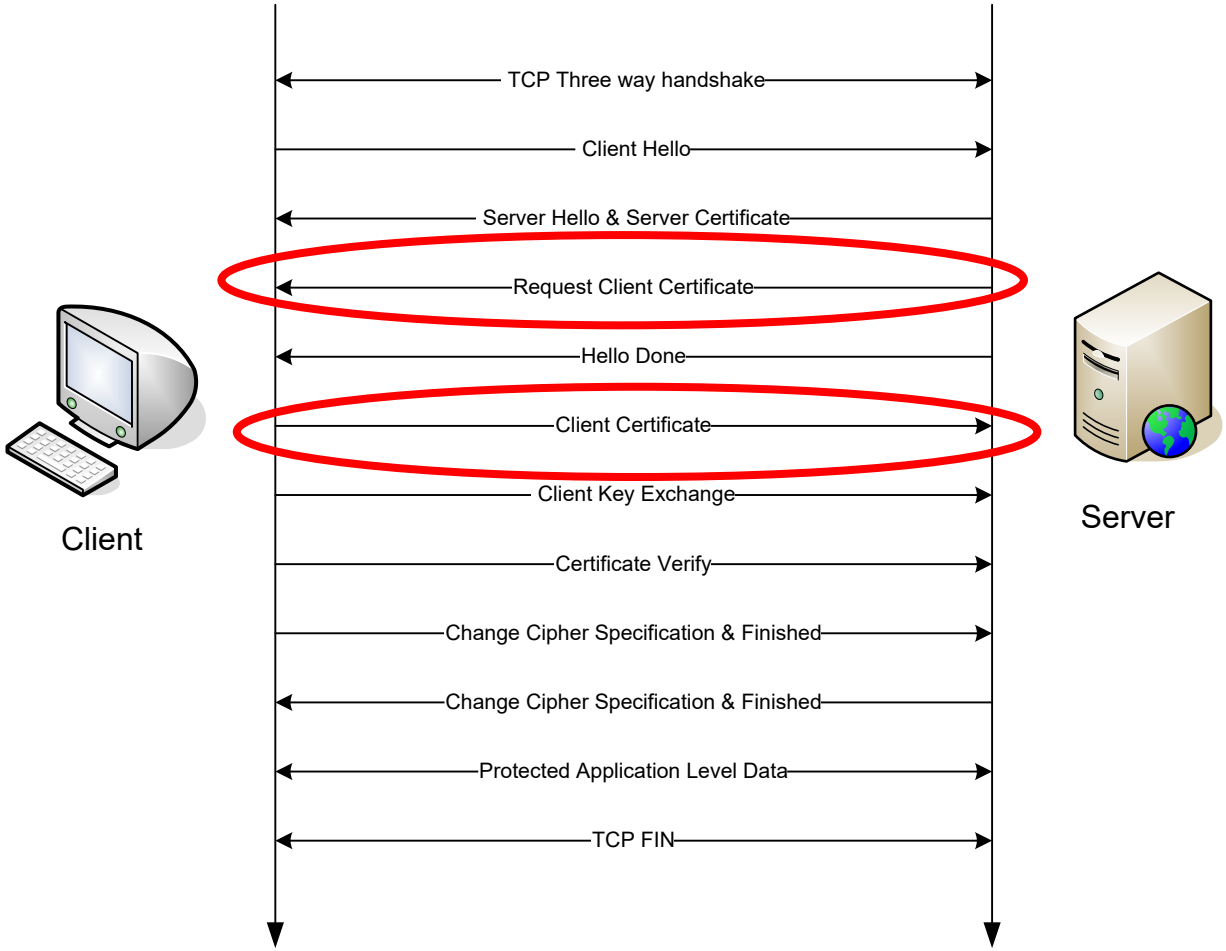
- Trusted Authentication Provider – Certificate Authority
- Client that has a certificate signed by the CA
- Server that has a certificate signed by the CA
- Both client and server trust the CA (directly or indirectly)

- Client prepares a certificate request with client attributes and sends it to the CA
- The CA requests identification information from the client – may include passport, driving license, or Kerberos ticket (within the Enterprise); CA validates the attributes, requested by the client
- CA issues a certificate to the client
- Client can now authenticate to servers by successfully completing an SSL/TLS handshake which shows that the client has a keypair that corresponds to the trusted certificate
- Server maps client to a specific user/priviliges by static mapping tables or by using the Subject or Alternate Subject fields in the certificate
- The server does not need to contact the CA to authenticate the user – all information is contained in the signed certificate

Case Study: Certificate Mapping/Authentication – 1/2

- SSL/TLS can be used as a security layer between communicating parties
- SSL/TLS encrypt the channel between the client and the server
- By default the client authenticates the server (typically using server certificates)
- The server can authenticate the client as well – each client (user) has a unique personal certificate
- Certificate Mapping: Client certificates can be mapped to specific user accounts
- No need to run an authentication protocol on top of SSL/TLS. Simply map client certificates to user accounts!
- PKI is required to establish trust and mapping

Case Study: Certificate Mapping/Authentication – 2/2



Trust

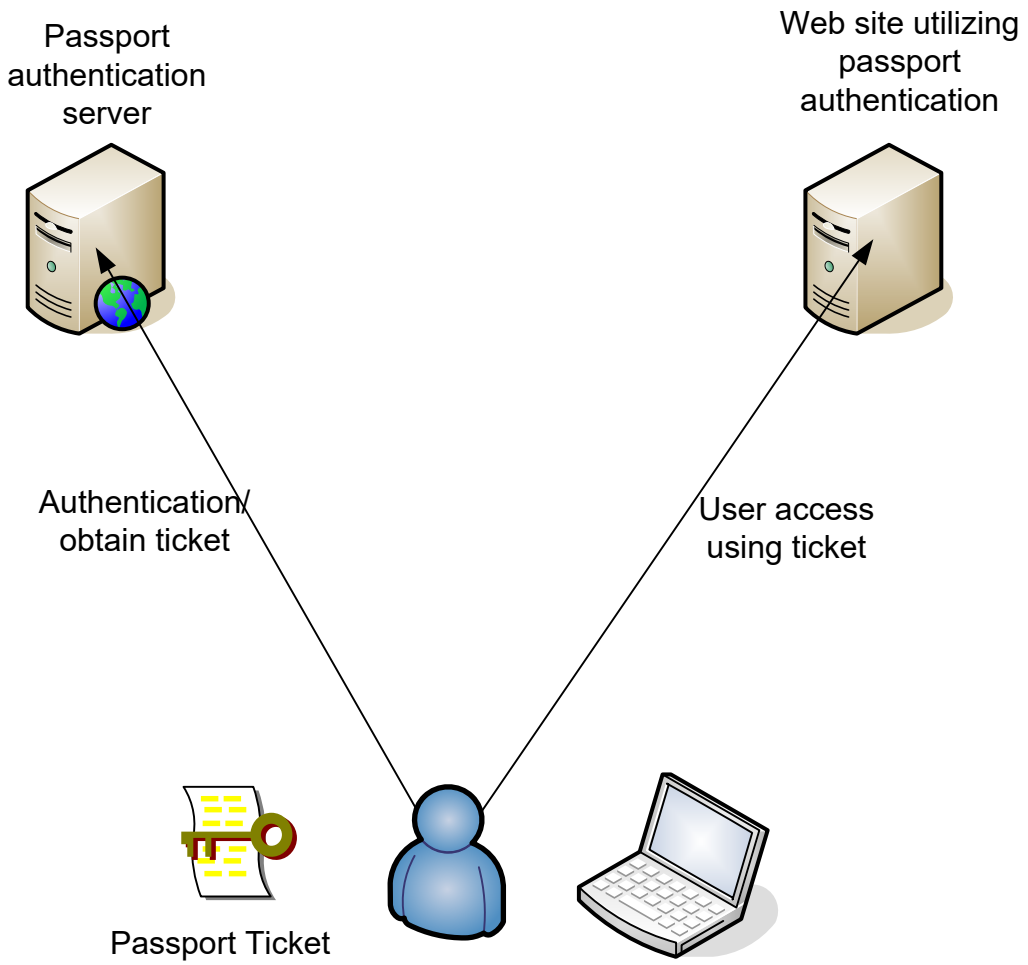
- Provides for authentication between different security domains
- May be transitive or non-transitive

- Kerberos trust: Referral TGTs, transitive
- NTLM trust: Non-transitive, MS RPC Remote authentication
- X.509 Certificates: Trusted Root CAs, Certificate Trust List
- PGP Certificates: Web of Trust
- RADIUS & TACACS+ Trust: Shared Secrets & Relay
- Federated Identity: Encrypted Ticket

* Authentication Federation

- Case Study: Microsoft Passport/Live
- SAML
- WS-Security

Case Study: Microsoft Passport/Windows Live ID - 1/2



Case Study: Microsoft Passport/Windows Live ID - 2/2

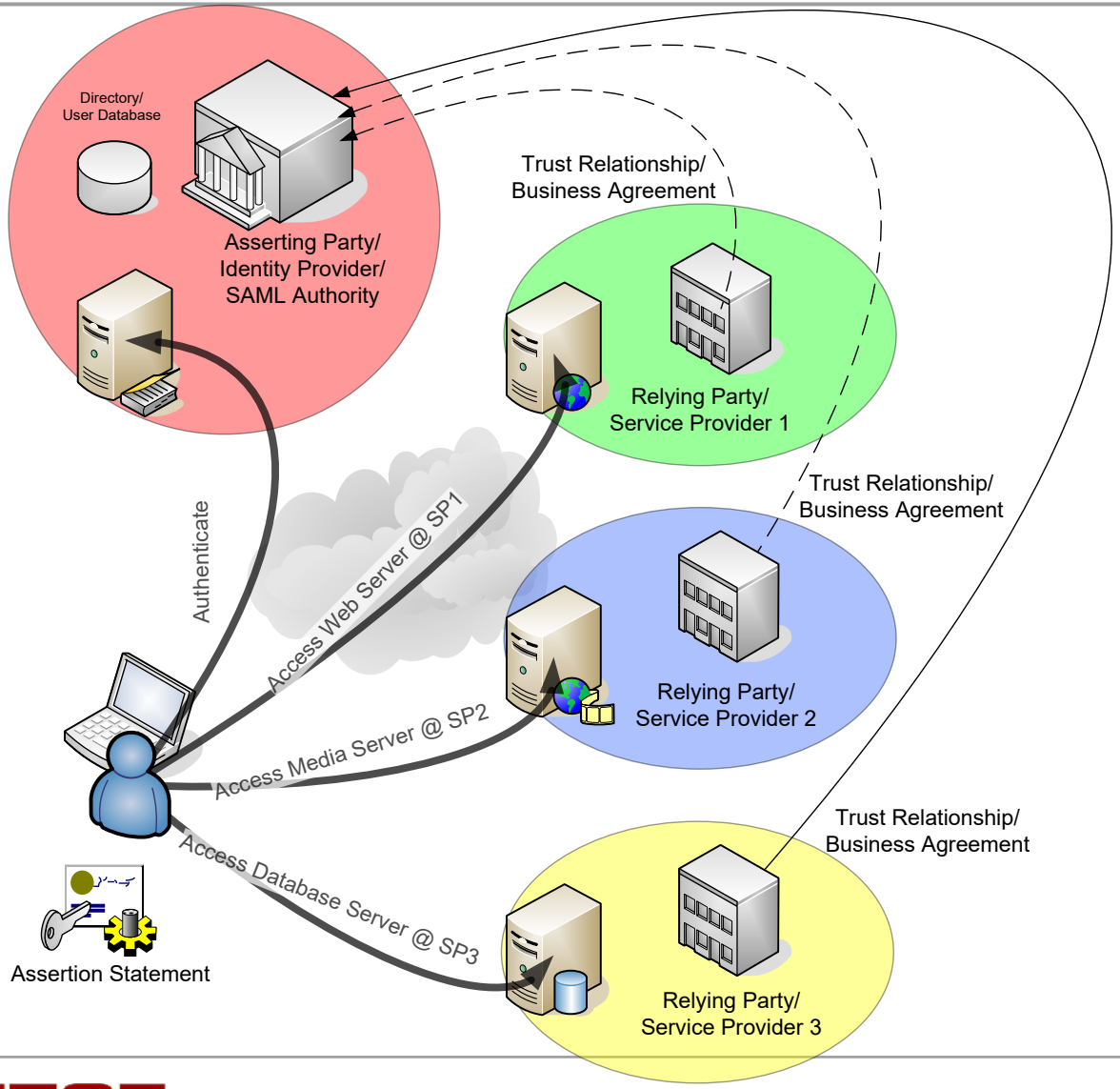
- Client Connects to a Passport Network Web Site using a browser
- Site checks for an authentication cookie (ticket)
- If a cookie is available, the user is authenticated
- If a cookie is not available, the client receives HTTP 302 “HTTP Redirect – Temporarily Moved” to passport.net
- Client authenticates to Passport/Windows Live ID and receives an authentication cookie (ticket=token)
- Passport.net returns HTTP 302 “HTTP Redirect” to the referring Web site
- Client re-connects to the Passport Network Web Site

Ticket-based Authentication - SAML/WS-Security

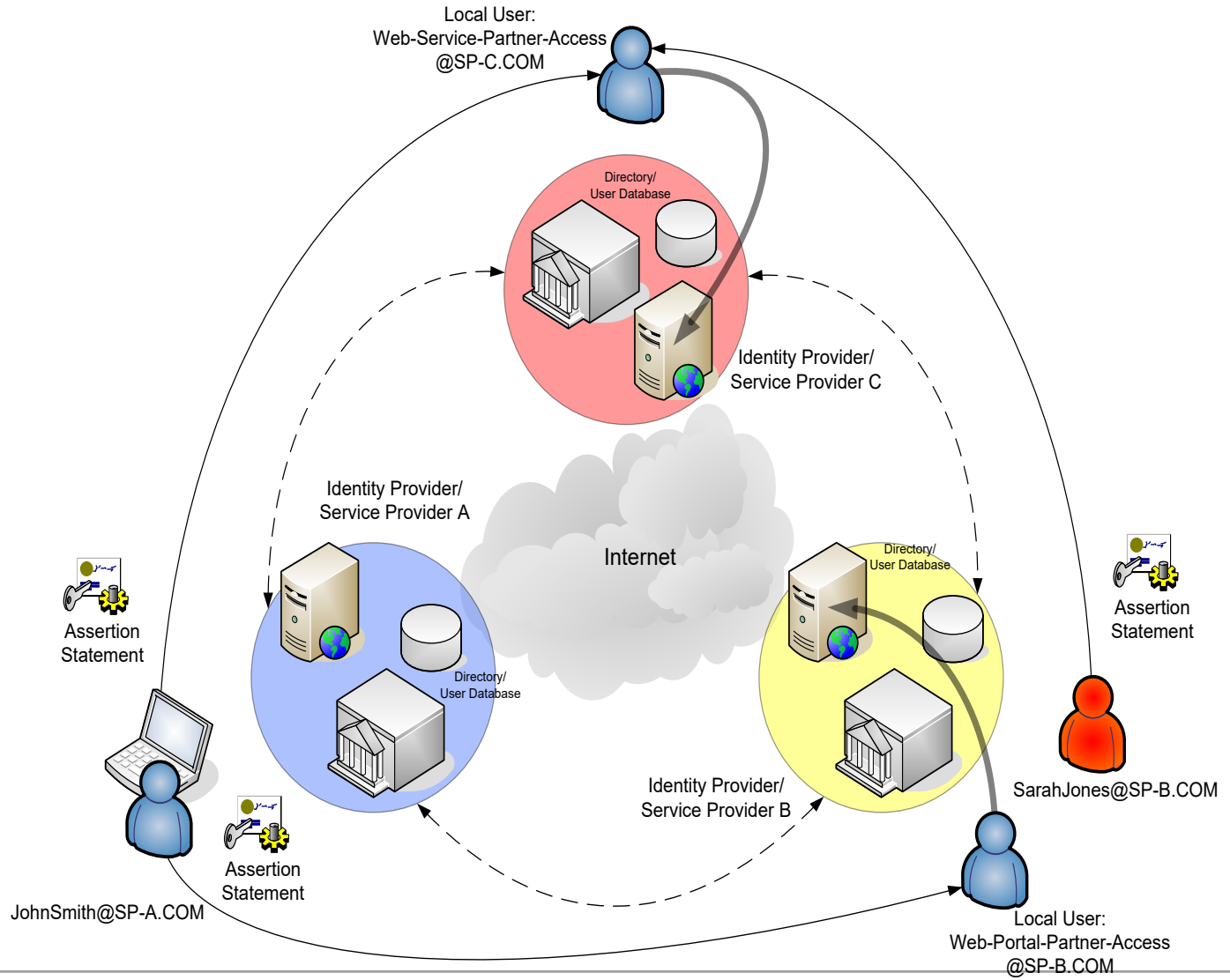
- XML = eXtensible Markup Language
- SOAP = Simple Object Access Protocol/Service Oriented Architecture Protocol

- XML is the universal document format on the Internet
- SOAP is the messaging protocol for XML – allows clients and servers on the Internet to communicate
- Typically runs on top of HTTP
- SAML and WS-Security allow XML to use its own authentication methods (independent from HTTP authentication)

SAML: Web Single Sign-on



SAML: Federated Identity



WS-Security

- WS-Security = Web Services Security
- WSS is managed by the OASIS consortium

- Encapsulates security and authentication information within SOAP messages
- Authentication Methods for WS-Security include:
  - SAML assertions
  - Kerberos Tickets
  - X.509 certificates

* Summary

- Select the authentication method that meets your security requirements
- Practical security approach: only as secure as the budget allows
- Certificates provide the best balance between ease of use and security
- Single Sign-on does not exist today; certificate based authentication provides the closest match
- One authentication method is required for access to applications and services, a different one to access the infrastructure

* Resources

Mechanics of User Identification and Authentication Companion Web Site

http://www.iamechanics.com

IETF Request for Comments

http://www.ietf.org

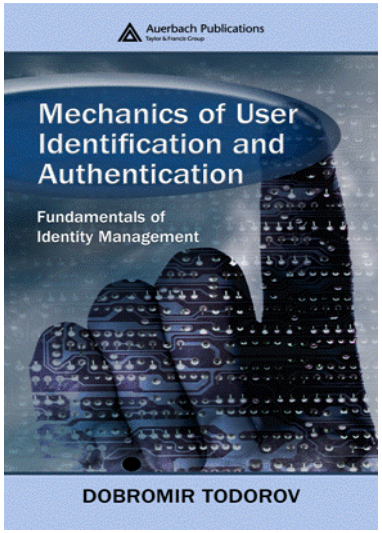
* Questions and Answers

Questions?

Visit http://www.iamechanics.com

Mechanics of User Identification and Authentication: Fundamentals of Identity Management

Publisher: **Auerbach Publications**
Available Languages : **English**
ISBN-10: **1420052195**
ISBN-13: **978-1420052190**



Contact

Dobromir Todorov

Dobromir.Todorov@bt.com

dobri@itce.com

http://www.iamechanics.com/Contact.html

EXTRA SLIDES:

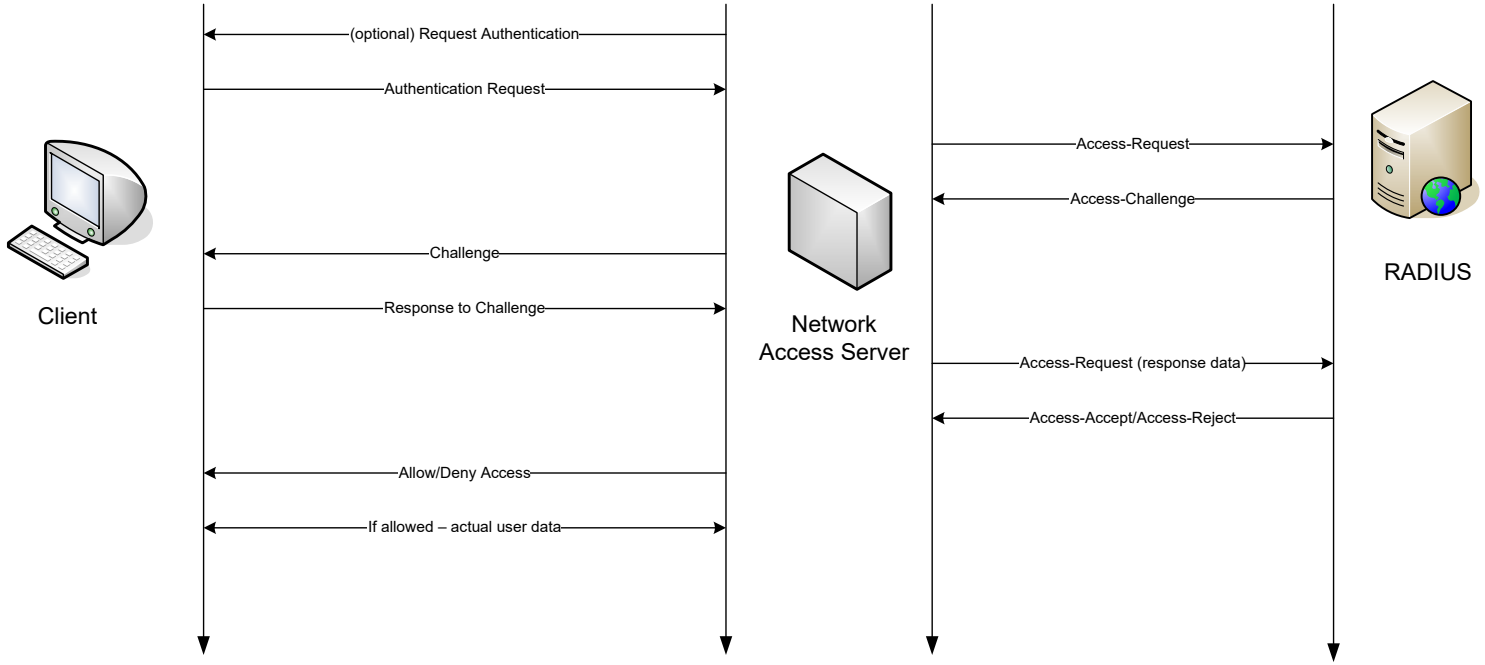
- Authenticating Access to the Network Infrastructure
- Authenticating Access to Applications and Services

* Authenticating Access to the Network Infrastructure

- Centralised Authentication
- Local Access
  - Wired
  - Wireless
- Remote Access
- IPSec Peer Identities and Authentication
- VPN Access

Centralised Network Infrastructure Authentication

- TACACS+
- RADIUS
- Certificate Based



Authenticating Local Wired Access

- Legacy: No authentication for access to the wired infrastructure; relies upon physical security.
- IEEE 802.1x (Port Based Network Access Control)
  - network devices must authenticate to the switch in order to access network resources
  - Aka EAP over LAN (EAPOL)
  - Some supplicants can authenticate both the machine and the user
- Network Admission Control (NAC)
  - Authenticate the client
  - Verify client compliance: software and hardware versions, settings, patches
  - Compliant clients receive a certificate and can access the network
  - Incompliant clients may be quarantined and allowed to remediate discrepancies

Authenticating Local Wireless Access

- Open Authentication/No encryption – the user does not provide any form of authentication and can send and receive data on the WLAN unrestrictedly
- Open Authentication/WEP encryption – the user does not provide authentication information. Still needs WEP password to communicate; WEP is VERY weak
- Shared Authentication/WEP – susceptible to various types of attacks; VERY insecure
- WPA (pre-802.11i) – Open Authentication + 802.1x
- WPA (pre-802.11i) – Open Authentication + Pre-shared key (PSK)
- WPA2 (802.11i) – Open Authentication + 802.1x
- WPA2 (802.11i) – Open Authentication + Pre-shared key (PSK)

Remote Access

- Terminal line authentication – interactive plaintext
- Point-to-Point Protocol (PPP) PAP/CHAP (and flavours)
  - PPP Link Control Protocol used for authentication messages
  - Authentication info sent as LCP paratmers
  - PAP = Password Authentication Protocol – cleartext username and password
  - CHAP = Challenge Authentication Protocol – client receives a challenge
  - Point-to-Point Protocol (PPP) EAP
- PPPoE (PPP over Ethernet)

IPSec (IKE/ISAKMP) Identities

- IKE allows communicating peers to be identified and authenticated
- Peer Hostname, IP Address or Group name (XAuth) can be used for identification
- If certificate authentication is used, Group name can be derived from Certificate DN
- Authentication methods:
  - Pre-shared key (password)
  - Certificates
  - (MS Only) Kerberos (uses Kerberos tickets & keys)

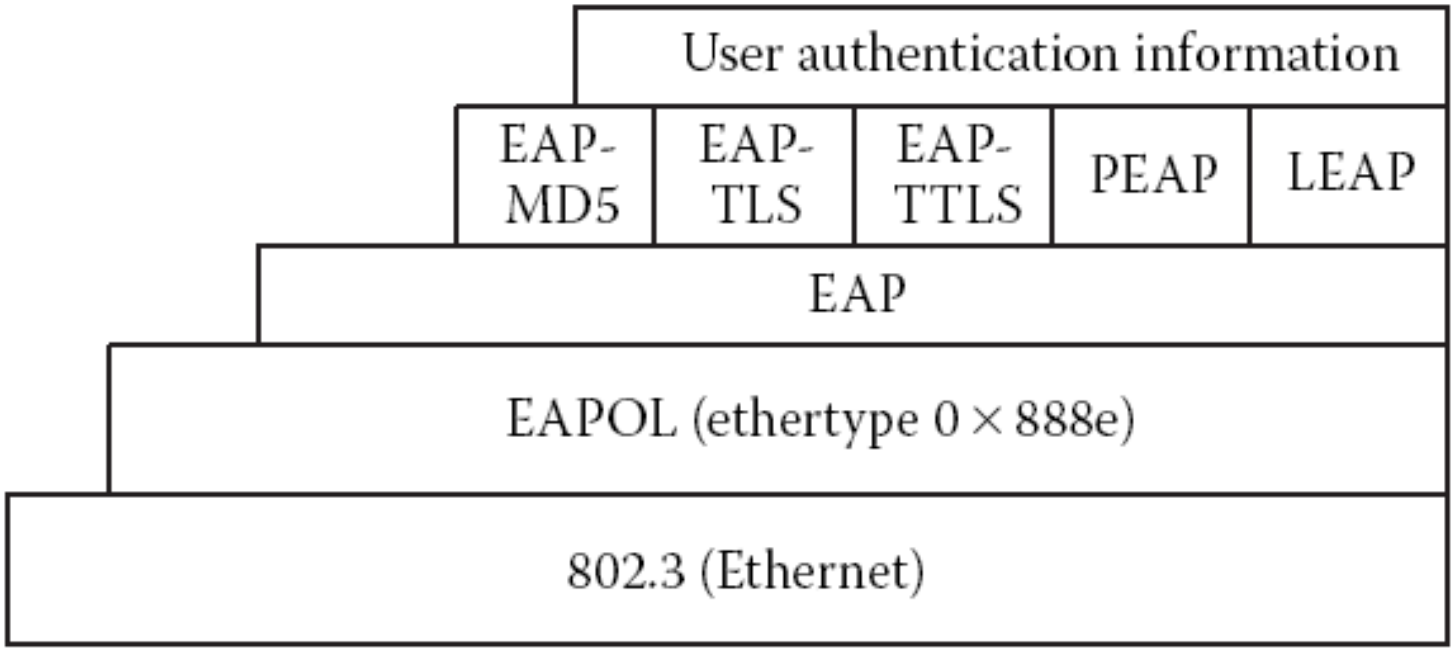
VPN Access

- PPTP = PPP over GRE (IP/47) + Control Channel
- L2TP = PPP over UDP/1720 – same authenticatiom methods apply
  - Potentially over IPSec/ESP Tunnel (Microsoft)
- IPSec Site-to-Site or Dynamic Multipoint VPN
  - Authenticates parties using Pre-shared Keys or Certificates
  - Peer authentication rather than user authentication (Layer 3)
- XAuth Authentication
  - Used by some VPN solutions– Cisco, Nortel
  - Not an IETF standard (security reasons)
  - Group Name = IPSec peer Identity (or portion of DN in certificate)
  - Group Password = Pre-shared password (or certificate based)
  - XAuth – user level authentication using ISAKMP CFG Messages on top of IKE channel
  - Problem: Anyone from the group can spoof the VPN Server and obtain either a plaintext password, or an encrypted password

EAP – Tunneling Authentication – 1/2

- EAP = Extensible Authentication Protocol
- EAP is not an authentication protocol per se
- NAS (router) does not need to understand the authentication mechanism; authentication tunnel can be end-to-end
- EAP does not require IP
  - Can run on top of Ethernet – EAPOL
  - Can run on top of PPP
  - Can run on top of RADIUS - tunneling
- Authentication mechanisms can use EAP as a transport
  - EAP
  - EAP-TLS – Certificate Mapping (authentication) over EAP
  - EAP-TTLS – TLS protected other protocols within EAP
  - PEAP – MS-CHAPv2 on top of EAP-TLS
  - LEAP – MS-CHAPv2 on top of EAP
  - EAP-FAST – similar to EAP-TLS but does not require certs

EAP – Tunneling Authentication – 2/2



* Authenticating Access to Applications and Services

- Delegation and Impersonation
- User Authentication and Security at the Application Level
- GSS-API/SSPI
- Kerberos
- NT Challenge/Response
- SPNEGO
- SASL
- TLS/SSL
- Application Specific Mechanisms

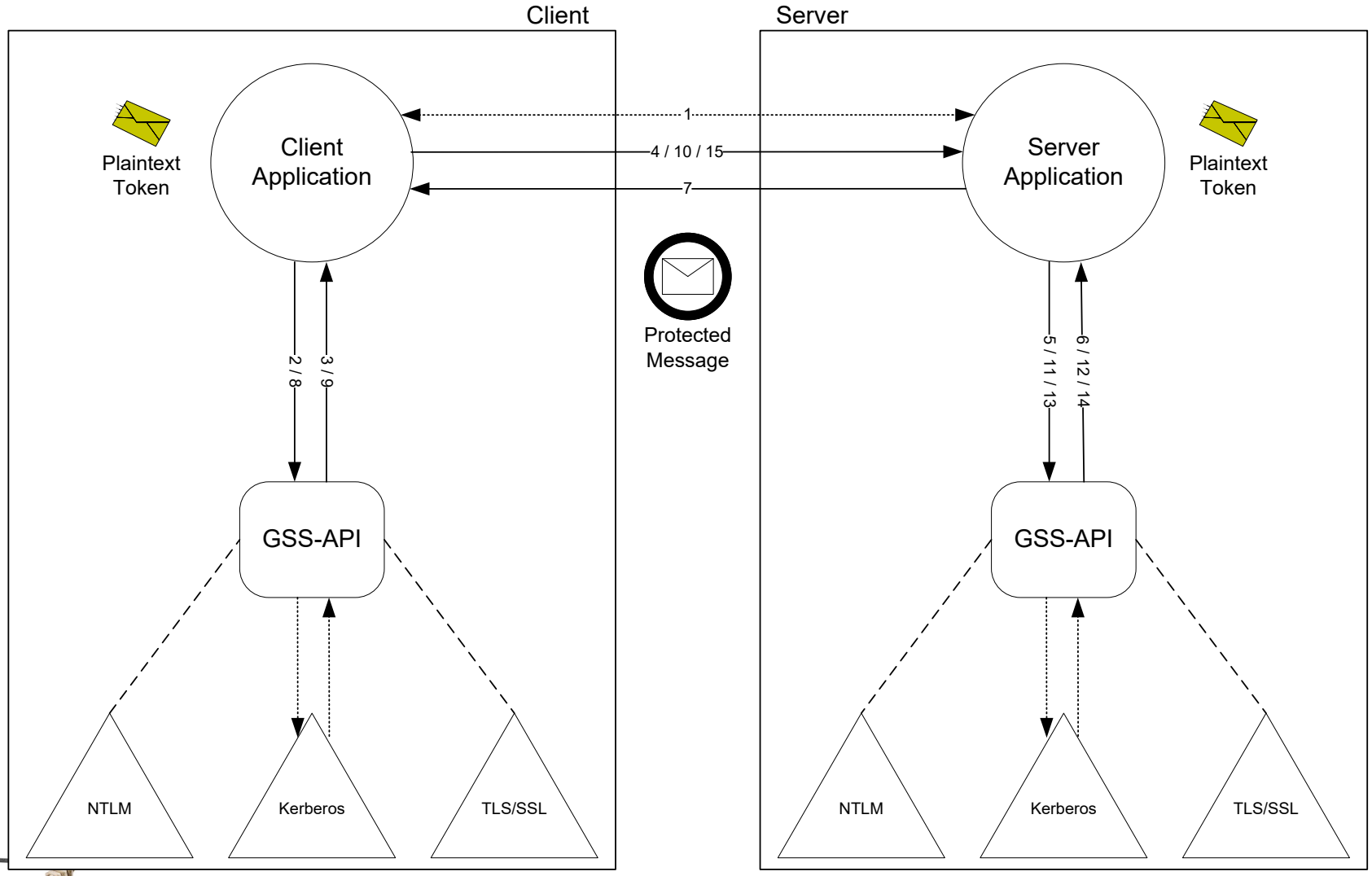
User Authentication and Security at the Application Level

- Establishes who the user is, so that the user can be granted access to resources and activity logged
- May happen as part of the application protocol or externally, using a dedicated protocol
- May be transparent to the end user (typical for ticket based authentication and SSO), or interact with the user
- May be provided by the operating system or be part of an abstraction layer, or be specific to the application
- Will typically provide key material or other support for network traffic (application payload) protection

GSS-API/SSPI – 1/2

- GSS-API – Generic Security Services API
- Layer of abstraction for applications
- Reusable security for applications – you don't need to rediscover the wheel
- Supports multiple authentication methods; however, the only one really used is Kerberos
- Besides authentication, supports network traffic protection (encryption/message integrity checking)
- GSS-API will use protection keys derived from the authentication protocol; in the case of Kerberos there are three standard methods to derive a session key from the Kerberos keys
- Not a fully pledged network protocol; runs on top of the application protocol (HTTP, FTP, LDAP)
- Configurable independently from the application
- GSS-API is generally compatible with Microsoft's SSPI specification
- GSS-API and SSPI don't work together with encryption and authentication

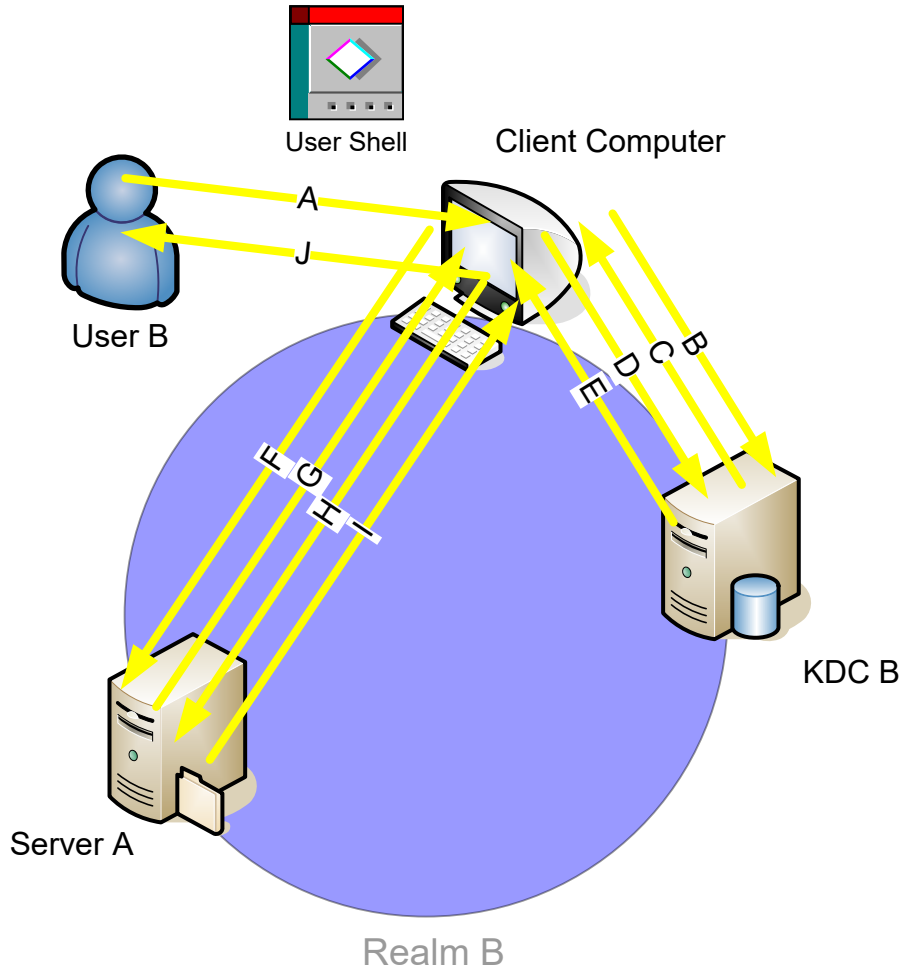
GSS-API/SSPI – 2/2



Kerberos

- Authentication protocol, developed as part of the MIT Athena project
- Describes an authentication system that depends on a trusted third party (Kerberos Distribution Centre) that forms a Kerberos domain (realm)
- Current Kerberos version V relies upon GSSAPI for authentication to applications
- Many applications support GSS-API/Kerberos out of the box
- Some applications may need to be “Kerberised”
- Kerberos supports impersonation and delegation
- Kerberos supports transitive trusts
- Mutual authentication and Peer-to-Peer authentication
- Supported natively on Windows 2000+
- Available for virtually all UNIX flavours

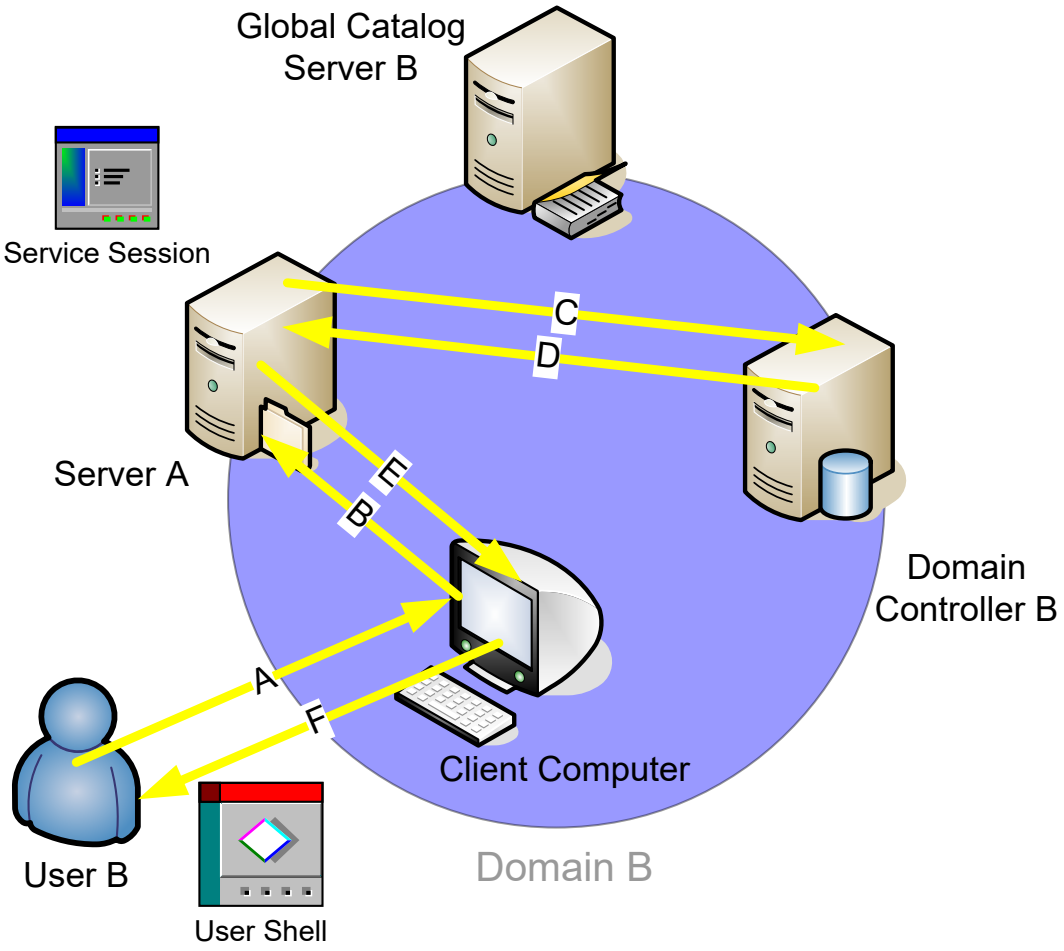
Indirect (Ticket Based) Authentication: Kerberos



NT Challenge/Response

- Supported by Microsoft operating systems and by many third party products
- Utilises Windows NT and LAN Manager (LM) password hashes
- Supports impersonation but not delegation
- Does not support transitive trusts
- Flavours:
  - Lan Manager (LM) version 1
  - Lan Manager (LM) version 2
  - NTLM version 1
  - NTLM version 2

Direct Network Authentication: NTLM

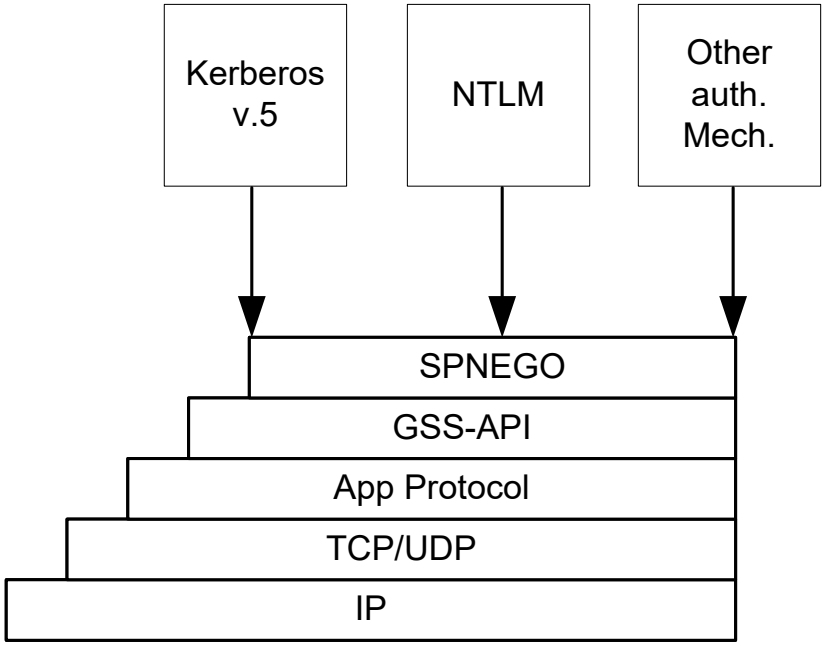


NTLM Authentication Mechanism Comparison

Parameter	LM v.1	NTLM v.1	LM v.2	NTLM v.2	NTLM2 SR
Authentication Mechanism	Challenge/ Response	Challenge/ Response	Challenge /Response	Challenge/ Response	Challenge/ Response
Trust model	Username and password must be known to client; LM Hash must be known to server or trusted domain controller	Username and password OR LM hash must be known to client; NT Hash must be known to server or trusted domain controller	Username and password OR NTLM hash OR NTLMv2 hash must be known to client; NT Hash must be known to server or trusted domain controller	Username and password OR NT hash must be known to client; Username and Domain name must be known to client and server; NT Hash must be known to server or trusted domain controller	Username and password OR NTLM hash must be known to client; NT Hash must be known to server or trusted domain controller
Mutual Authentication	No	No	No	No	No
Protocol Security (1-5)	2 (Weak)	3 (Moderate)	4 (Good)	4 (Good)	4 (Good)
Challenge/ Seed	Server only: Random/Time based, 8 bytes	Server only: Random/Time based, 8 bytes	Server based: Random/Time based Client: Random	Server based: Random/Time based Client: Random & Time based	Server based: Random/Time based Client: Random
Response Hash/ Encryption approach	DES ECB using a 56+56+16-bit key	DES ECB using a 56+56+16-bit key	HMAC-MD5 using a 128-bit key	HMAC-MD5 using a 128-bit key	DES ECB using a 56+56+16-bit key

SPNEGO

- A separate authentication mechanism in GSS-API & SSPI
- However, does not provide for actual authentication
- Negotiates the preferred authentication method
- (primarily Windows NT+) Kerberos and NTLM supported
- May be susceptible to authentication downgrade attacks
- Microsoft calls this protocol Negotiate SSP

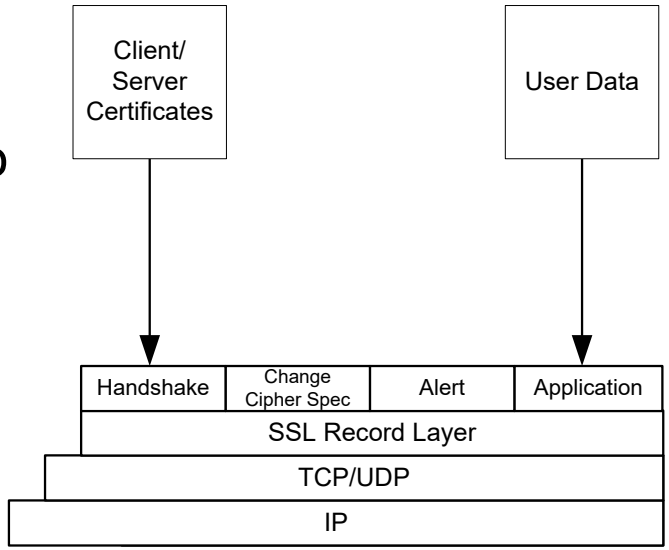


SASL

- SASL = Simple Authentication and Security Layer (RFC 2222)
- Layer of abstractions for applications to access user authentication functions and have their network traffic protected
- Widely used by legacy TCP interactive applications:
  - SMTP
  - POP3
  - IMAP
  - LDAP (modified)
- Applications include a command verb to enter SASL negotiation
- SASL relies upon pluggable authentication mechanisms to do the actual job:
  - Kerberos IV
  - GSS-API
  - S/Key
  - CRAM-MD5/Digest-MD5

TLS/SSL

- SSL 3.1 = TLS 1.0
- SSL typically utilises a separate port for protected traffic
- TLS recommends that the application 'upgrades' the communication channel to TLS by using verbs such as STARTTLS
- Both provide for:
  - Peer Identification and authentication (typically by X.509 certs)
  - Channel encryption
  - Channel authentication
- Server authentication only
- Server and Client authentication
- Client Certificate Mapping



Application Specific Mechanisms

- HTTP
  - Anonymous
  - Basic
- Telnet
  - Login
  - Authentication Option (also includes traffic encryption!)
- Sun RPC
  - AUTH_NULL - no authentication
  - AUTH_UNIX – UID and GID
  - AUTH_DES – Private/public keys and DES encryption
  - RPCSEC_GSS – GSS-API/Kerberos
- MS SQL Authentication
- Oracle Authentication