Protocolli di Autenticazione

Indice

- Tecniche di autenticazione e multi-factor authentication
- Protocolli di autenticazione
- Gestione delle password e dei certificati.
- Modelli e tecniche per autorizzazione e controllo dell’accesso
- Privileged Identity and Access Management (PAM)
- Cenni sulla segregazione e segmentazione delle reti

Indice

- Tecniche di autenticazione e multi-factor authentication
- Protocolli di autenticazione
- Gestione delle password e dei certificati.
- Modelli e tecniche per autorizzazione e controllo dell’accesso
- Privileged Identity and Access Management (PAM)
- Cenni sulla segregazione e segmentazione delle reti

Protocolli di autenticazione

- Tre categorie di protocolli fra dispositivi collegati in rete
  - PAP (Password Authentication Protocol)
  - CHAP (Challenge-Handshake Authentication Protocol)
  - EAP (Extensible Authentication Protocol).

Protocolli di autenticazione

- Tre categorie di protocolli fra dispositivi collegati in rete
  - PAP (Password Authentication Protocol)
  - CHAP (Challenge-Handshake Authentication Protocol)
  - EAP (Extensible Authentication Protocol).

Password Authentication Protocol

- L’autenticazione avviene attraverso la coppia <nome-utente,password>
- Non si utilizza alcuna forma di crittografia
- Vulnerabile ad attacchi
  - Intercettazione
  - Replica

Protocolli di autenticazione

- Tre categorie di protocolli fra dispositivi collegati in rete
  - PAP (Password Authentication Protocol)
  - CHAP (Challenge-Handshake Authentication Protocol)
  - EAP (Extensible Authentication Protocol).

Challenge-Handshake Authentication Protocol (CHAP)

- L’autenticazione avviene attraverso la coppia <nome-utente,password>
- CHAP mitiga le vulnerabilità di PAP utilizzando numeri casuali e crittografia in un protocollo di autenticazione in tre passi

Challenge-Handshake Authentication Protocol (CHAP)

1. Challange
   Il server invia al client una stringa casuale (chiamata *nonce*)
2. Il client genera un hash crittografico (es., MD5) usando il *nonce* e la password e lo invia al server
3. Il server confronta il valore ricevuto con quello calcolato localmente

Challenge-Handshake Authentication Protocol (CHAP)

- Vantaggi rispetto a PAP
  - Un attaccante che intercetta il traffico non può recuperare la password
  - Replicare il passo 2 non consente autenticazione perché la stringa casuale (nonce) è valida 1 sola volta
- Vulnerabilità CHAP
  - Le password sono memorizzate in chiaro sul server

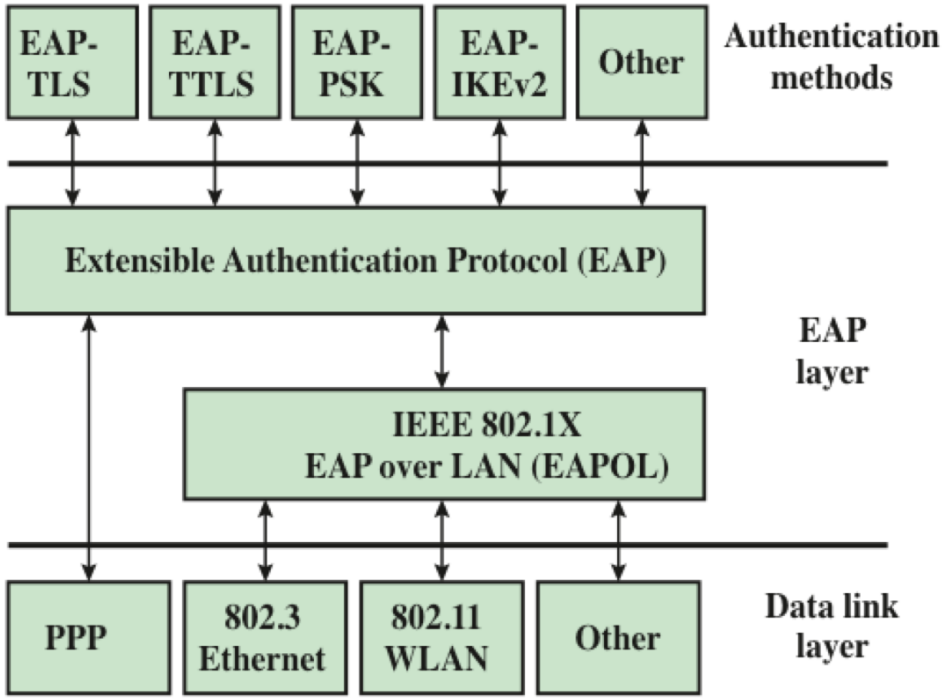
Protocolli di autenticazione

- Tre categorie di protocolli fra dispositivi collegati in rete
  - PAP (Password Authentication Protocol)
  - CHAP (Challenge-Handshake Authentication Protocol)
  - EAP (Extensible Authentication Protocol).

Extensible Authentication Protocol (EAP)

- EAP è uno schema per la realizzazione di protocolli di autenticazione
- Utilizzato da standard IEEE 802.1X che specifica modalità di accesso alla rete da parte di un client
  - Ad es., autenticazione di un dispositivo su rete WiFi

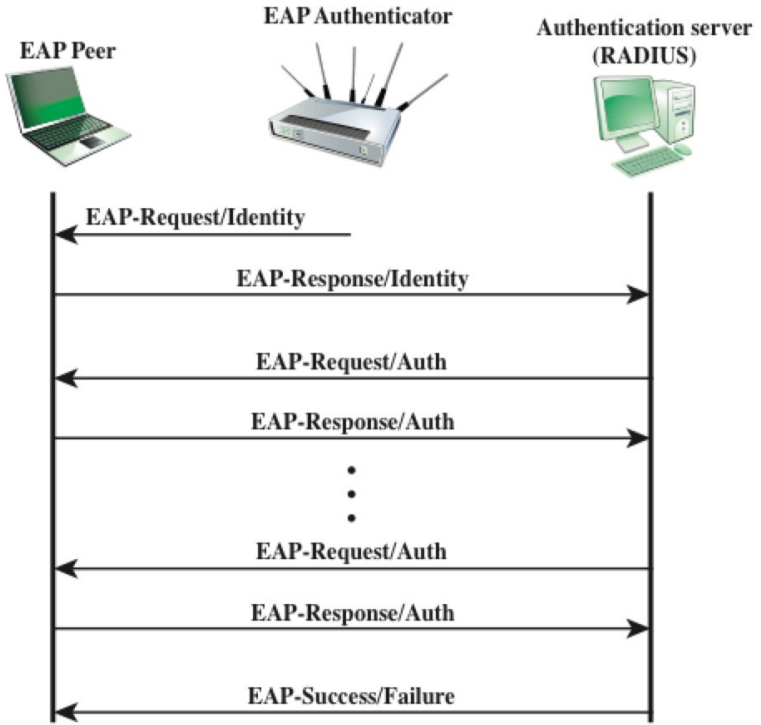
Extensible Authentication Protocol (EAP)



Metodi di autenticazione EAP

Figura da W. Stallings, Network Security, 2017

Extensible Authentication Protocol (EAP)



- Il client accede a authenticator dopo aver ricevuto autorizzazione da un server

Figura da W. Stallings, Network Security, 2017

Pre-Shared Keys (PSK)

- L’autenticazione fra due o più sistemi richiede la memorizzazione di stringhe binarie (chiavi) in fase di configurazione.
  - Può essere un’unica chiave memorizzata in tutti i dispositivi
  - Può esserci un dispositivo *master* che memorizza le chiavi degli altri dispositivi
- Il protocollo WEP per WiFi si basa su PSK

Protocollo WEP (Wired Equivalent Privacy)

- Autenticazione WiFi fra dispositivo e Access Point (AP) mediante unica chiave condivisa
  1. AP invia un numero casuale al dispositivo
  2. Il dispositivo cifra il numero usando la chiave (memorizzata sul dispositivo e su AP) e lo invia a AP
  3. AP riceve il messaggio cifrato e lo decifra usando la chiave. Se ottiene il numero casuale generato al primo passo, l'autenticazione va a buon fine

Protocollo WEP (Wired Equivalent Privacy)

- Autenticazione WiFi fra dispositivo e Access Point (AP) mediante unica chiave condivisa
  1. AP invia un numero casuale al dispositivo
  2. Il dispositivo cifra il numero usando la chiave (memorizzata sul dispositivo e su AP) e lo invia a AP
  3. AP riceve il messaggio cifrato e lo decifra usando la chiave. Se ottiene il numero casuale generato al primo passo, l'autenticazione va a buon fine

Protocollo WEP (Wired Equivalent Privacy)

- Autenticazione WiFi fra dispositivo e Access Point (AP) mediante unica chiave condivisa
  1. AP invia un numero casuale al dispositivo
  2. Il dispositivo cifra il numero usando la chiave (memorizzata sul dispositivo e su AP) e lo invia a AP
  3. AP riceve il messaggio cifrato e lo decifra usando la chiave. Se ottiene il numero casuale generato al primo passo, l'autenticazione va a buon fine

Protocollo di Needham-Schroeder

- Proposto nel 1978
- Un dispositivo ha il ruolo di terza parte fidata
- L'autenticazione fra dispositivi è mediata dalla terza parte fidata
- Ciascun dispositivo condivide una chiave con la terza parte

Protocollo di Needham-Schroeder

- Indichiamo con
  - Sam terza parte fidata
  - Alice e Bob due dispositivi che intendono comunicare
- Alice e Sam condividono la chiave K_{AS}
- Bob e Sam condividono la chiave K_{BS}

Protocollo di Needham-Schroeder

1. Alice vuole comunicare con Bob e invia la richiesta a Sam insieme a un *nonce* N_A
2. Sam crea una chiave simmetrica K_{AB} e la invia ad Alice. Invia anche ad Alice la stessa chiave cifrata con la chiave K_{BS}

$$S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$

3. Alice invia a Bob la chiave K_{AB} cifrata con la chiave K_{BS}
4. Bob decifra il messaggio e usa K_{AB} per cifrare un nonce N_B che invia ad Alice

Protocollo di Needham-Schroeder

1. Alice vuole comunicare con Bob e invia la richiesta a Sam insieme a un *nonce* N_A
2. Sam crea una chiave simmetrica K_{AB} e la invia ad Alice. Invia anche ad Alice la stessa chiave cifrata con la chiave K_{BS} $S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. Alice invia a Bob la chiave K_{AB} cifrata con la chiave K_{BS}
4. Bob decifra il messaggio e usa K_{AB} per cifrare un nonce N_B che invia ad Alice

Protocollo di Needham-Schroeder

1. Alice vuole comunicare con Bob e invia la richiesta a Sam insieme a un *nonce* N_A
2. Sam crea una chiave simmetrica K_{AB} e la invia ad Alice. Invia anche ad Alice la stessa chiave cifrata con la chiave K_{BS} $S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. Alice invia a Bob la chiave K_{AB} cifrata con la chiave K_{BS}
4. Bob decifra il messaggio e usa K_{AB} per cifrare un nonce N_B che invia ad Alice

Protocollo di Needham-Schroeder

1. Alice vuole comunicare con Bob e invia la richiesta a Sam insieme a un *nonce* N_A
2. Sam crea una chiave simmetrica K_{AB} e la invia ad Alice. Invia anche ad Alice la stessa chiave cifrata con la chiave K_{BS} $S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. Alice invia a Bob la chiave K_{AB} cifrata con la chiave K_{BS}
4. Bob decifra il messaggio e usa K_{AB} per cifrare un nonce N_B che invia ad Alice

Protocollo di Needham-Schroeder

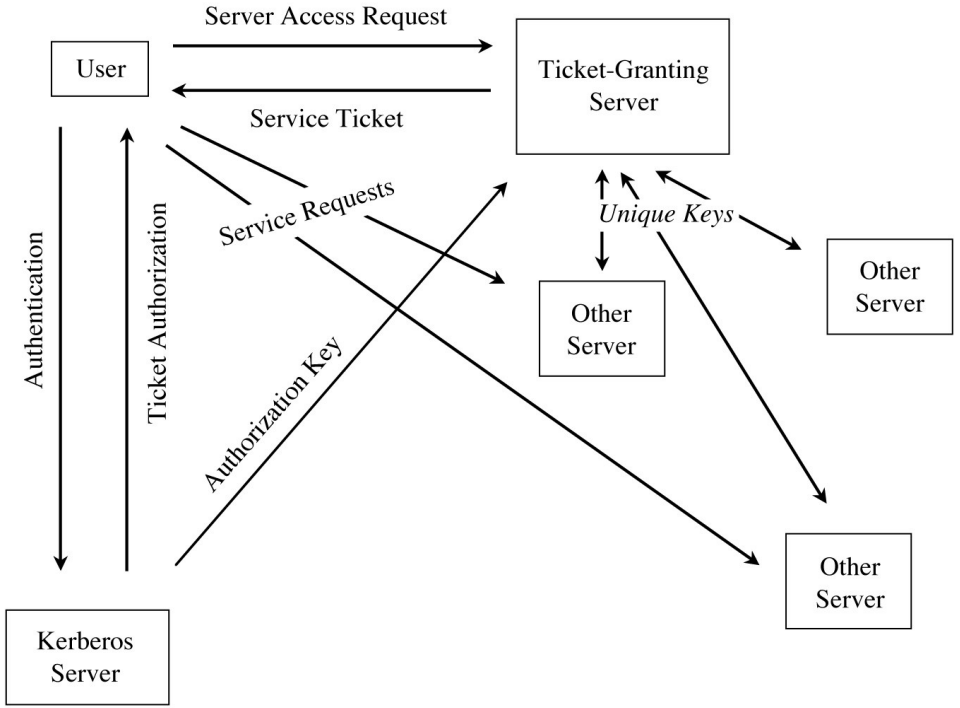
- Bob autentica Alice perché ha inviato un messaggio con una chiave nota solo a Bob e a Sam
  - Sam può generare il messaggio cifrato solo per soggetti *fidati*, come Alice, che hanno chiavi condivise
- Alice prova l'autenticità di Bob perché è in grado di decifrare il messaggio che ha inviato Sam
- Kerberos (MIT) è derivato da questo protocollo

Approfondimento 1: Extensible Authentication Protocol

- EAP-MD5 fornisce il Supporto EAP a livello di base. EAP-MD5 fornisce solo l'autenticazione unidirezionale: non supporta l'autenticazione reciproca del client e della rete. Non supporta chiavi di sessione dinamiche.
- EAP-TLS (Transport Layer Security) fornisce l'autenticazione basata su certificati. Supporta autenticazione reciproca del client e della rete. Si basa su certificati lato client e lato server. Può essere utilizzato per generare chiavi dinamiche di sessione (ad es., per autenticazione WiFi).
- EAP-TTLS (Tunneled Transport Layer Security) è stato sviluppato da Funk Software e Certicom, come estensione di EAP-TLS. Questo metodo prevede l'autenticazione reciproca basata su certificati tramite un canale crittografato (o tunnel). A differenza di EAP-TLS, EAP-TTLS richiede solo i certificati lato server.
- EAP-FAST (Flexible Authentication via Secure Tunneling) è stato sviluppato da Cisco. Invece di utilizzare un certificato per ottenere l'autenticazione reciproca. EAP-FAST autentica per mezzo di PAC (Protected Access Credential) che può essere gestito dinamicamente dal server di autenticazione.
- EAP-SIM. Metodo di autenticazione e per la distribuzione di chiavi di sessione sviluppato in ambito GSM. EAP-SIM utilizza una chiave basata su sessione dinamica, derivata dalla scheda SIM e dal server, per crittografare i dati. EAP-SIM richiede l'immissione di un codice di verifica utente (PIN).
- EAP-AKA. (Authentication and Key Agreement) Metodo di protocollo di autenticazione estensibile per l'autenticazione UMTS.
- LEAP (Lightweight Extensible Authentication Protocol) è un tipo di autenticazione EAP utilizzato principalmente in Cisco Aironet WLAN. Crittografa le trasmissioni di dati utilizzando chiavi generate dinamicamente e supporta l'autenticazione reciproca. Cisco ha concesso in licenza LEAP ad altri produttori.
- PEAP (Protected Extensible Authentication Protocol) fornisce un metodo per trasportare i dati di autenticazione in modo sicuro, inclusi i protocolli di vecchia generazione basati su password. PEAP utilizza il tunneling tra i client PEAP e un server di autenticazione.

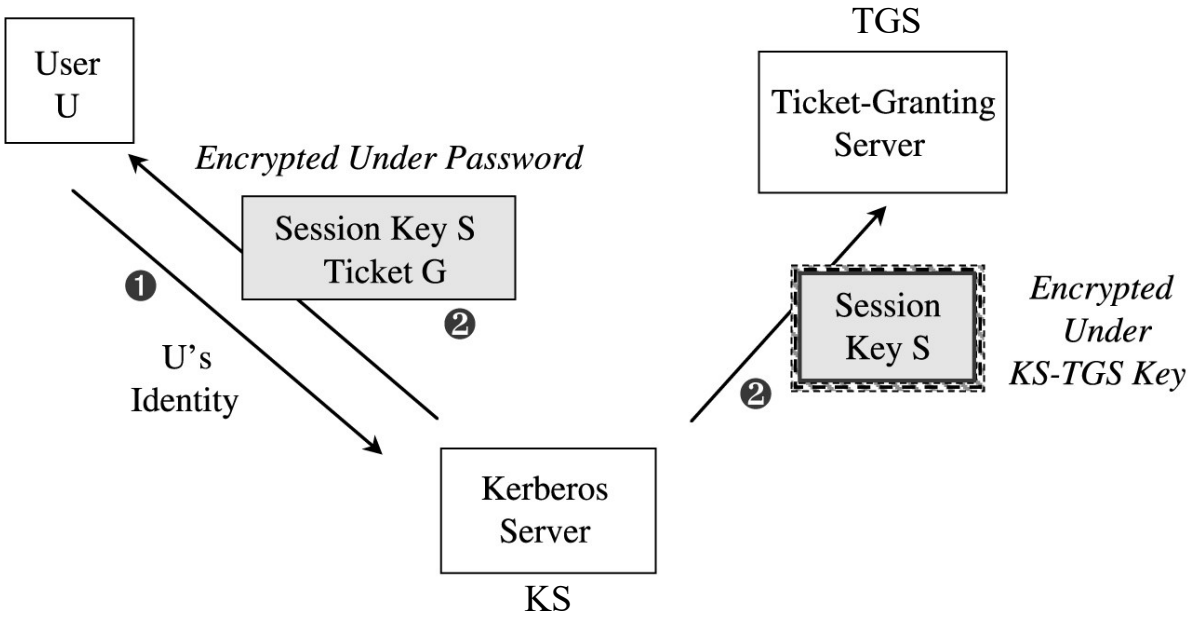
https://www.intel.com/content/www/us/en/support/articles/000006999/network-and-i-o/wireless-networking.html

Approfondimento 2: Kerberos

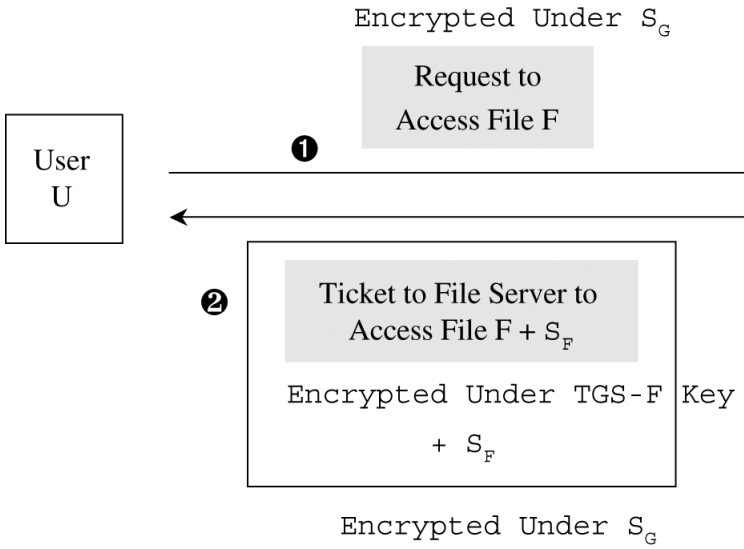


- Il server Kerberos gestisce l'autenticazione di tutti i soggetti
- Un secondo server (Ticket-Granting Server - TGS) ha memorizzate le chiavi condivise con ciascuno dei soggetti che comunicano attraverso il servizio
- L'autenticazione avviene mediante messaggi cifrati con chiavi note solo al singolo soggetto e al TGS in modo analogo a quanto proposto dal protocollo di Needham-Schroeder

Approfondimento 2:
Kerberos – autenticazione utente



Approfondimento 2:
Kerberos – richiesta di una risorsa



$$U \rightarrow TGS: \{A, F\}_{S_G}$$
$$TGS \rightarrow U: \{T_S, L, S_F, F, \{T_S, L, S_F, U\}_{S_{GF}}\}_{S_G}$$
$$U \rightarrow F: \{T_S, L, S_F, U\}_{S_{GF}} \ \{A, T_A\}_{S_F}$$
$$F \rightarrow U: \{T_A + 1\}_{S_F}$$

S_G è la chiave di sessione, parte del *ticket* G
S_{GF} è la chiave condivisa fra TGS e F
T_A e T_S rappresentano *timestamp*
L è il *time-to-live* del *ticket*