1. Fondamenti della Sicurezza dei Sistemi

Requisiti Base per il Confinamento

Per implementare politiche di sicurezza efficaci, un sistema deve:

- **Distinguere gli utenti** (identificazione univoca)
- **Identificare l'operazione richiesta** (tipo di accesso)
- **Identificare l'oggetto target** (risorsa da proteggere)
- **Prendere una decisione** (autorizzare o negare)

2. Modello AAA (Authentication, Authorization, Accounting/Auditing)

Authentication (Autenticazione)

- **Scopo**: Identificare l'utente nel sistema
- **Implementazioni**:
  - Username/password tradizionali
  - Chiavi crittografiche e certificati
  - Biometria (fingerprint, face recognition)
- **Principio IdUtente ↔ Utente**: Ogni utente umano dovrebbe avere un identificatore univoco

Authorization (Autorizzazione)

Processo in tre fasi:

1. **Richiesta di accesso** dal client
2. **Controllo di accesso** da parte del sistema
3. **Autorizzazione** finale (concessa/negata)

Auditing

Eventi tipici monitorati:

- Autenticazione (successo/fallimento)
- Richieste di accesso alle risorse
- Risultati delle operazioni autorizzate
- Comportamenti anomali del sistema

Livelli di Auditing:

- Access auditing (controllo accessi)
- System security auditing
- Network security auditing
- Compliance auditing (ISO 27001)

4. Modelli di Controllo Accesso

DAC (Discretionary Access Control)

- **Principio**: Utenti possono modificare permessi sulle proprie risorse
- **Vantaggi**: Flessibilità, facilità d'uso
- **Svantaggi**: Sicurezza delegata agli utenti
- **Esempi**: Permessi Unix (chmod), ACL Windows

MAC (Mandatory Access Control)

- **Principio**: Solo amministratori possono configurare i diritti
- **Vantaggi**: Sicurezza superiore, controllo centralizzato
- **Svantaggi**: Rigidità, complessità gestionale
- **Uso**: Ambienti militari, server critici

Approccio Ibrido MAC+DAC

- Accesso consentito solo se entrambi i controlli autorizzano
- "Isole di discrezionalità" confinate da "muri obbligatori"
- Esempio: Web server isolato via MAC, utenti separati via DAC

5. Protocolli di Sicurezza di Rete

CHAP/EAP (Challenge-Handshake Authentication Protocol)

Processo di autenticazione punto-punto:

1. Client presenta username al NAS

2. NAS invia challenge (ID + nonce)
3. Client risponde con hash calcolato
4. NAS verifica confrontando gli hash

IPsec

- **Obiettivo**: Garantire RID (Riservatezza, Integrità, Disponibilità)
- **Caratteristica**: Trasparente alle applicazioni
- **Implementazione**: End-to-end encryption a livello IP

SSL/TLS

Architettura:

- **Handshake Protocol** (Livello 7): Negoziazione parametri di sicurezza
- **Record Protocol** (Livello 4): Cifratura simmetrica e verifica integrità

Caratteristiche:

- Combina crittografia simmetrica e asimmetrica
- Trasparente alle applicazioni
- Utilizzato in HTTPS per web security

RADIUS

- **Funzione**: Delegare autenticazione a server centralizzato (KDC)
- **Trasporto**: UDP porte 1812/1823
- **Architettura**: Client/server di livello 7

Kerberos

- **Innovazione**: Introduzione dell'Authentication System (AS)
- **Processo**: Client → AS → TGS → Target Service
- **Tecnologia**: Crittografia simmetrica con ticket-based authentication

6. Firewall: Packet Filter vs Application Proxy

Packet Filter Firewall

Funzionamento:

- Opera ai livelli 3-4 OSI (rete/trasporto)
- Esamina header IP, porte sorgente/destinazione
- Ignorar il contenuto applicativo (payload)

Processo Decisionale:

1. Pacchetto arriva all'interfaccia di rete
2. Verifica sequenziale delle regole
3. Prima regola matchata determina l'azione (accept/deny/reject)
4. Logging dell'evento

Vantaggi:

- Alte prestazioni
- Trasparenza per l'utente
- Basso utilizzo risorse

Svantaggi:

- Sicurezza limitata (solo header inspection)
- Nessun controllo del contenuto applicativo
- Vulnerabile ad attacchi application-layer

Application Proxy Firewall

Funzionamento:

- Opera al livello 7 OSI (applicazione)
- Spezza la comunicazione client-server in due connessioni
- Esamina il contenuto applicativo completo

Architettura:

1. Connessione Client → Proxy
2. Connessione Proxy → Server
3. Proxy funge da intermediario intelligente

Vantaggi:

- Sicurezza elevata (deep packet inspection)
- Controllo granulare del contenuto

- Protezione application-specific

Svantaggi:

- Prestazioni ridotte (overhead processamento)
- Proxy dedicato per ogni servizio
- Configurazione client necessaria
- Vulnerabilità ai bug delle applicazioni proxy

7. Considerazioni Architetturali

Scelta del Firewall

Packet Filter: Adatto per:

- Reti ad alto traffico
- Filtering basico
- Ambienti dove le prestazioni sono critiche

Application Proxy: Adatto per:

- Ambienti ad alta sicurezza
- Controllo granulare delle applicazioni
- Compliance con regolamentazioni stringenti