

Progetto Finale TAV 24/25

Project Work A: Firewall Software Multithread

Realizzare un **Firewall Software** che filtra pacchetti di rete provenienti da diversi client. Il firewall analizza i pacchetti, li confronta con una serie di **regole di sicurezza**, e decide se **accettare** o **bloccare** il pacchetto.

Gli attacchi di rete sono simulati da **più thread concorrenti** che inviano pacchetti pseudocasuali (su protocolli e porte). Il numero di client deve essere configurabile.

L'obiettivo è di simulare un ambiente realistico di sicurezza di rete.

Specifiche tecniche

Sono riportate di seguito le specifiche tecniche minimali che il sistema deve possedere:

- Dependency Injection (DI):
 - Usare iniezione dei componenti.
 - Separare bene logica di filtering e gestione delle regole.
 - Consentire di cambiare il set di regole senza modificare il codice.
- Aspect Oriented Programming (AOP):
 - Creare un Aspect per intercettare i pacchetti bloccati.
 - Il logging dei pacchetti bloccati deve avvenire automaticamente (senza modificare il codice).
- Thread e Sincronizzazione:
 - Simulare più client di rete: ogni thread rappresenta un client che invia pacchetti.
 - Proteggere adeguatamente l'accesso al set di regole.
- Clonazione:
 - Consentire di clonare un set di regole per testare nuove regole senza alterare il firewall attivo.

Infine, occorre gestire una console interattiva per modificare regole in modalità "live". È lasciato allo studente la definizione dei casi d'uso principali.

La soluzione proposta deve essere altamente modulare e facilmente manutenibile.

Dettagli per la consegna del PW

Sviluppare un progetto in Java che realizzi le specifiche tecniche progettuali, utilizzando un sistema di costruzione del codice. Rilasciare un progetto git che contenga:

- Il codice sorgente della soluzione sviluppata, ben commentato, comprensivo di una batteria di test adeguata a verificare tutti gli aspetti critici del sistema.
- La documentazione (max 3 pagine) che illustri:
 - La modellazione UML della soluzione progettata;
 - Come è stata realizzata la Dependency Injection;
 - Come è stato implementato l'Aspect;
 - Come sono gestiti Thread e sincronizzazione;
 - Come avviene la clonazione.
- Un file README.txt con Istruzioni su come compilare ed eseguire il progetto e le indicazioni delle dipendenze software necessarie.

Note aggiuntive

Per quanto non espressamente indicato, lo studente è libero di attuare proprie scelte implementative.

Progetto Finale TAV 24/25

Project Work B: Intrusion Detection System (IDS)

Costruire un **IDS** (Intrusion Detection System) che monitora eventi sospetti (es. login strani, accessi a file riservati, login fuori orario,).

Il sistema deve essere progettato per ricevere eventi da diverse fonti ed effettuare analisi in tempo reale.

L'obiettivo è di simulare un ambiente di monitoraggio di sicurezza, corredato da una console di monitoraggio di eventi e allarmi.

La soluzione proposta deve essere altamente modulare e facilmente manutenibile.

La soluzione proposta deve essere altamente modulare e facilmente manutenibile.

Specifiche tecniche

Sono riportate di seguito le specifiche tecniche minimali che il sistema deve possedere:

- **Dependency Injection (DI):**
 - Iniettare i moduli per la raccolta degli eventi, l'analisi e la gestione delle allerte
 - Consentire la sostituzione del modulo di analisi senza impattare sul codice.
- **Aspect Oriented Programming (AOP):**
 - Creare un Aspect che:
 - Eseguire il logging di tutti gli eventi sospetti.
 - Mandare notifiche automatiche (Alert) su eventi critici (che superano una soglia di criticità per tipologia di evento).
- **Thread e Sincronizzazione:**
 - Ogni fonte genera eventi su un thread separato.
 - Proteggere l'accesso all'elenco degli alert.
- **Clonazione:**
 - Consentire di clonare una configurazione di analisi per testare nuove policy senza alterare il sistema attuale.

Infine, occorre gestire una console interattiva che mostra in tempo reale eventi e allarmi.
È lasciato allo studente la definizione dei casi d'uso principali.

Dettagli per la consegna del progetto

Sviluppare un progetto in Java che realizzi le specifiche tecniche progettuali, utilizzando un sistema di costruzione del codice. Rilasciare un progetto git che contenga:

- Il codice sorgente della soluzione sviluppata, ben commentato, comprensivo di una batteria di test adeguata a verificare tutti gli aspetti critici del sistema.
- La documentazione (max 3 pagine) che illustri:
 - La modellazione UML della soluzione progettata;
 - Come è stata realizzata la Dependency Injection;
 - Come è stato implementato l'Aspect;
 - Come sono gestiti Thread e sincronizzazione;
 - Come avviene la clonazione.
- Un file README.txt con Istruzioni su come compilare ed eseguire il progetto e le indicazioni delle dipendenze software necessarie.

Note aggiuntive

Per quanto non espressamente indicato, lo studente è libero di attuare proprie scelte implementative.

○