
Statement of Work Progetto

NetGun

RIFERIMENTO	STATEMENT OF WORK, INGEGNERIA DEL SOFTWARE
DESTINATARIO	Università degli Studi Di Salerno
VERSIONE	0.1
DATA	11/10/2022
PRESENTATO DA	Carlo Colizzi, Giulio Incoronato, Antonio Mazzearella
APPROVATO DA	

Revision History

Data	Versione	Descrizione	Autori
11/10/2022	0.1	Prima stesura	Carlo Colizzi, Giulio Incoronato, Antonio Mazzearella
16/10/2022	0.2	Seconda stesura	Carlo Colizzi, Giulio Incoronato, Antonio Mazzearella

Piano Strategico/Strategic Plan

A fronte del crescente numero di attacchi informatici riscontrati negli ultimi anni, e' necessario individuare le possibili minacce presenti nei sistemi. Così che queste possano essere gestite adeguatamente.

Obiettivi di Business/Business Needs

Attualmente, per raggiungere gli obiettivi, si vuole offrire ai Penetration Tester e Network Analyst, uno strumento per facilitare l'individuazione di mal configurazioni e CVE all'interno di infrastrutture di rete.

Ambito del Prodotto/Product Scope

Il tool deve:

1. Rilevare ed acquisire dati su tutti i servizi offerti dall'infrastruttura di rete;
2. Rilevare mal configurazioni sui servizi offerti dall'infrastruttura di rete;
3. Consigliare tools per interagire con i servizi offerti dall'infrastruttura di rete;
4. Rilevare CVE presenti nei servizi offerti dall'infrastruttura di rete;
5. Fornire un report sulle analisi effettuate.

Scenari

- **Scenario 1:** Mario é un Penetration Tester freelancer, essendo una figura molto ricercata, il suo numero di clienti aumenta molto rapidamente, questo non gli permette di effettuare test come si deve per ogni cliente. Decide quindi di utilizzare lo scanner NetGun, il quale, effettuando vari scanner ed enumerazioni in maniera automatica, gli permette di velocizzare i test. Tramite lo scanner riesce ad automatizzare le task ripetitive, così da avere più tempo per gestire più clienti.
- **Scenario 2:** Cristina é la Network Analyst di un azienda Campana, il suo direttore, a seguito di un attacco informatico ad un azienda affiliata, le chiede di effettuare un report su eventuali problematiche di sicurezza legate ai server aziendali. Cristina é ancora alle prime armi, e per evitare di dimenticare qualcosa, si affida a NetGun, il quale la guida nel processo di testing, fornendole varie informazioni sui servizi e consigliandole dei tool di riferimento.
- **Scenario 3:** Rosa lavora come Web developer, ha dimestichezza con i pc e con i server. Decide di comprare i componenti e assemblare un server casalingo, in modo da avere un cloud di facile accesso per contenere tutti i file su cui lavora. Data la sensibilità dei dati che archivia, ha bisogno di tutta la sicurezza possibile. Le viene in aiuto NetGun, che tramite uno scanning del server rivela varie mal configurazioni. Ora Rosa può essere più serena quando accede ai suoi dati.

Data di Inizio e di Fine

Inizio: Ottobre 2022.

Fine: Gennaio - Febbraio 2023.

Deliverables

RAD, SDD, ODD, Matrice di Tracciabilità, Test Plan, Test Case Specification, Test incident Report, Test Summary Report, Manuale D'Uso, Manuale Installazione e ogni altro documento richiesto per lo sviluppo del sistema.

Vincoli/Constraints

Vincoli collaborativi e comunicativi.

- Rispetto scadenze delle scadenze intermedie/di fine progetto **definite nello statement of work**;
- Budget/Effort non superiore a $50 \cdot n$ ore dove n sono i membri del team;
- Uso di sistemi di versioning - GitHub in particolare;
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo;
- Utilizzo di tool di per la suddivisione dei task e attività (Trello o similare);
- Utilizzo di tool di comunicazione tracciabile (Slack).

Vincoli tecnici

Analisi e specifica dei requisiti

- Specifica di **minimo 2** e **massimo 4** scenari per ogni membro del team;
- Specifica di **minimo 2** e **massimo 4** requisiti funzionali e non funzionali per ogni membro del team;
- **Esattamente** uno use case per ogni membro del team - i casi d'uso aggiuntivi **non** saranno valutati;
- **Esattamente** un sequence diagram ogni due membri del team - i sequence diagram aggiuntivi **non** saranno valutati;
- **Esattamente** un diagramma a scelta tra statechart e activity diagram ogni due membri del team - ulteriori diagrammi **non** verranno valutati;
- Specifica di un class diagram per team - eventuali object diagram **non** verranno valutati.

System Design

- Specifica di **minimo 2** e **massimo 4** design goal per ogni membro del team;
- Definizione di **un diagramma** di decomposizione dei sottosistemi per team, con annessa descrizione e motivazione all'uso;

- Definizione di **un deployment diagram** per team, con annessa descrizione e motivazione all'uso.

Object Design

- Uso di **minimo** uno e **massimo** due design pattern per team (devono essere selezionati tra quelli presentati a lezione);
- Uso di UML;

Testing

- Ogni studente dovrà effettuare il testing di unità, tramite category partition, di **esattamente** un metodo di una classe sviluppata;
- Ogni studente dovrà effettuare il testing di sistema, tramite category partition, di **esattamente** una funzionalità del sistema sviluppato.

Criteri di Accettazione/Acceptance Criteria

- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Adeguato utilizzo del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Adeguato utilizzo di Slack, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Adeguato utilizzo di Trello, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Documentazione adeguata. Verranno usati tool di **plagiarism detection** per identificare casi in cui gli studenti hanno copiato da progetti di anni precedenti e/o da altre fonti.
- Appropriato test di unità di un metodo sviluppato, che preveda il **rispetto dei vincoli**.
- Appropriato test di sistema di una funzionalità del sistema sviluppato, che preveda il **rispetto dei vincoli**.

Criteri di premialità

- Uso adeguato di sistemi di **build**;
- Uso adeguato di un processo di **continuous integration** tramite Travis;
- Uso adeguato di tool di controllo della qualità (ad esempio, **CheckStyle**);
- Adozione di processi di **code review**;

- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).