






1 INFORMAZIONI GENERALI

Candidato	Nome: Filippo	Cognome: Zinetti
	 filippo.zinetti@samtreveno.ch	
Luogo di lavoro	Scuola Arti e Mestieri / CPT Trevano-Canobbio	
Orientamento	<input type="checkbox"/> 88601 Sviluppo di applicazioni <input checked="" type="checkbox"/> 88602 Informatica aziendale <input type="checkbox"/> 88603 Tecnica dei sistemi	
Superiore professionale	Nome: Fabio	Cognome: Piccioni
	 fabio.piccioni@edu.ti.ch	
Perito 1	Nome: Antonio	Cognome: Fontana
	 antonio.fontana@rsi.ch	
Perito 2	Nome: Claudio	Cognome: Bortoluzzi
	 claudio.bortoluzzi@rsi.ch	
Periodo	Dal 03.05.2021 al 27.05.2021 (secondo griglia oraria per il progetto LPI)	
Orario di lavoro	Secondo orari della convocazione della DFP	
Numero di ore	80	
Pianificazione 80h (in ore o %)	Analisi: 10	
	Implementazione: 50	
	Test: 10	
	Documentazione: 30	

2 PROCEDURA

- Il candidato realizza il lavoro autonomamente sulla base del quaderno dei compiti ricevuto il 1 ° giorno.
- Il quaderno dei compiti è approvato dai periti. È anche presentato, commentato e discusso con il candidato. Con la sua firma, il candidato accetta il lavoro proposto.
- Il candidato ha conoscenza della scheda di valutazione prima di iniziare il lavoro.
- Il candidato è responsabile dei suoi dati.
- In caso di problemi gravi, il candidato o il superiore professionale avvertono immediatamente il perito.
- Il candidato ha la possibilità di chiedere aiuto, ma deve menzionarlo nella documentazione.
- Alla fine del tempo a disposizione per la realizzazione del LPI, il candidato deve inviare via e-mail il progetto al superiore professionale e al perito 1. In parallelo, una copia cartacea della documentazione dovrà essere fornita in duplice copia (superiore professionale e perito). Quest'ultima deve essere in tutto identica alla versione elettronica.

3 TITOLO

Butler extensions, Inventory & Behaviour

4 HARDWARE E SOFTWARE DISPONIBILE

1 PC fornito dalla scuola con i tool necessari per lo svolgimento del progetto.

5 PREREQUISITI

Butler software. Il nuovo progetto si appoggia su un progetto software precedentemente sviluppato dall'allievo. Il software di base è composto da un modulo server ed un modulo client multiplatforma. Al momento le funzionalità gestite da questi due componenti sono la parte di comunicazione sicura tra di loro, ed un modulo client per mostrare avvisi istantanei e personalizzabili su tutti i client conosciuti dal componente Butler server.

6 DESCRIZIONE DEL PROGETTO

Il progetto consiste nello sviluppo di due estensioni. Il software di base, chiamato "Butler", è composto da due componenti separati, una componente Server ed un componente Client. Il software è stato sviluppato in Python dall'allievo. La componente client è multiplatforma.

I due componenti comunicano tra di loro dei messaggi in modo asincrono e sicuro. Su questa piattaforma di comunicazione è già stata sviluppata una prima estensione del client in grado di eseguire istruzioni inviate dalla console del componente Server.

Sfruttando questa infrastruttura di base l'allievo dovrà implementare due nuove estensioni, una per la raccolta delle informazioni hardware e software dei clients, ed una seconda estensione per raccogliere informazioni comportamentali sul traffico di rete del client.

I due moduli permetteranno di elaborare lato server i dati recuperati dai client conosciuti.

Il primo modulo permette di elaborare un inventario del parco macchine.

Il secondo modulo permette di analizzare il traffico di rete dei client per segnalare eventuali comportamenti che fuoriescono dal traffico abituale ("standard deviation" del traffico di rete).

Più in dettaglio questo secondo modulo dovrà avvisare l'amministratore che un computer sta facendo elaborazioni "speciali", che non rientrano tra le elaborazioni abituali di un client di rete. L'allievo dovrà studiare un sistema in grado di apprendere i comportamenti standard dei client (flussi di rete soltanto) creare quindi un modello che verrà confrontato per ogni client. Se dovessero riscontrarsi delle differenze dal modello standard, il client dovrà segnalarlo e sarà compito dell'amministratore analizzare manualmente il traffico "sospetto" e decidere se metterlo in una white list per il client specifico oppure se validare tale traffico per tutti i client (modifica del modello).

Per fare un esempio:

tutti i client avranno delle connessioni con i/il server Active Directory, con il Butler Server, con il File Server. Quindi si potrebbe ipotizzare un modello standard di comunicazione che preveda del traffico TCP o UDP a destinazione di questi specifici server. Se il modulo Butler client dovesse verificare una nuova connessione SSH verso un nuovo indirizzo IP, si tratterebbe di un comportamento che esce dal modello standard e quindi da segnalare.

I moduli devono poter essere attivati o disattivati dall'amministratore per ogni singolo client.

L'allievo sarà confrontato con:

- lo sviluppo di estensioni ad un software preesistente, in linguaggio Python
- la ricerca di soluzioni per la raccolta delle informazioni dal sistema operativo
- lo studio delle informazioni necessarie di comunicazione tra client e server in modo da ottenere una implementazione "leggera" che non sovraccarichi la rete
- la suddivisione dei carichi computazionali da eseguire su client o su server

Qui di seguito i requisiti riassunti:

- Server -> adattare Butler server per la gestione dei due nuovi moduli (attivazione/disattivazione dei componenti lato client, e raccolta informazioni provenienti dai client)
- Client -> Estensione "**inventory**"
Raccolta delle informazioni sui software installati, e sulle caratteristiche del client (hostname, mac, ip, OS, build, ecc.).
Limitarsi ad informazioni di base utili a scopo di inventario.
- Server -> interfaccia per rappresentare i dati raccolti dall'estensione "**inventory**"
- Client -> estensione "**behaviour**"
Notifica quando del traffico di rete non corrisponde ad un modello prestabilito. (Prevedere una fase iniziale di apprendimento)
- Server -> interfaccia per il modulo "**behaviour**" con:
 - o notifica degli elementi "sospetti" con visualizzazione del traffico che ha scatenato la notifica
 - o Gestione del modello comportamentale standard
 - o Gestione delle whitelist specifiche per client

7 RISULTATI FINALI

Il candidato è responsabile della consegna al superiore professionale e al perito:

- Una pianificazione iniziale (entro il primo giorno)
- Una documentazione del progetto
- Un diario di lavoro giornaliero da consegnare **entro le 18:00**

8 PUNTI TECNICI SPECIFICI VALUTATI

La griglia di valutazione definisce i criteri generali secondo cui il lavoro del candidato sarà valutato (documentazione, diario, rispetto degli standard, qualità, ...).

Inoltre, il lavoro sarà valutato sui seguenti 7 punti specifici (punti da A14 a A20):

1. [159] – *Analisi del problema (programmazione)*
2. [161] – *Concezione, Design (programmazione)*
3. [164] – *Codifica: Gestione degli errori*
4. [119] – *Utilità (applicazione)*
5. [165] – *Implementazione della soluzione (programmazione)*
6. [166] – *Stile di codifica; Leggibilità del codice*
7. [146] – *Soddisfazione dell'utente: GUI, utilizzazione*

9 FIRMA

Candidato
(Canobbio, 03.05.2021)

Superiore professionale
(Canobbio, 03.05.2021)

Perito 1
(luogo e data)

Perito 2
(luogo e data)
