



25 settembre 2020

Luca LONGHI

its tECH tALENT FACTORY

CYBAZE

documentazione project work

360° SCANNER

**SOMMARIO**

[1. INTRODUZIONE 2](#_Toc51937986)

[1.1 **Premessa** 2](#_Toc51937987)

[1.2 **Campo di applicazione** 2](#_Toc51937988)

[1.3 **Glossario e Acronimi** 3](#_Toc51937989)

[2. METODOLOGIE DI LAVORO E SVILUPPO DEL CODICE 4](#_Toc51937990)

[2.1 **Metodo di approccio alla richiesta e soluzione scelta** 5](#_Toc51937991)

[2.2 **Mockup** 6](#_Toc51937992)

[2.3 **Diagramma di Flusso** 7](#_Toc51937993)

[2.4 **Tabella delle versioni** 9](#_Toc51937994)

[2.5 **Snippet e GitHub** 10](#_Toc51937995)

[3. TEST ESEGUITI E LIMITAZIONI 11](#_Toc51937996)

[**3.1** **Tipologia di test** 11](#_Toc51937997)

[**3.2** **Limitazioni imposte esterne** 12](#_Toc51937998)

[4. CONCLUSIONI 13](#_Toc51937999)

[**4.1** **Sviluppi e features future** 13](#_Toc51938000)

[**4.2** **Conclusioni** 14](#_Toc51938001)

[5. CREDITI E SITOGRAFIA 15](#_Toc51938002)

1. **INTRODUZIONE**
2. **Premessa**

Questo progetto nasce inizialmente come un Project Work del percorso formativo dell’ITS Tech Talent Factory in collaborazione con l’azienda Cybaze.

****

Il progetto nasce come un progetto semplice, ma date le ambizioni degli studenti e l’entusiasmo visto anche dai professori, si è deciso di farlo diventare qualcosa di più grande, utilizzabile anche come buona pubblicità per il singolo componente del Team che si occuperà in futuro di collaborare con altre persone, o comunque di migliorare il progetto ed aggiungere nuove features, con un’utilizzo open source del software

Il progetto in questione è uno scanner a 360 gradi, in grado di identificare e catalogare in un database un indirizzo IP, la sua classe e le informazioni riferite ad esso, in grado di risolvere l’indirizzo IP di un server DNS, il suo certificato SSL e altre cose del cazzo. Il programma è stato pensato per dotare un analista di rete o un esperto di cybersecurity di uno strumento unico ed efficiente per poter identificare le cose sopra.

1. **Campo di applicazione**

Il programma in questione sarà utilizzato soprattutto nei SOC, da esperti di cybersecurity ed analisti di rete che dovranno analizzare in modo rapido e veloce indirizzi IP e Domini internet per verificare la stabilità e il grado di pericolo dei servizi che verranno controllati.

Utile anche a studenti e a Cloud Administrator. Inoltre sono previste delle nuove features che potrebbero rendere ancora più interessante l’utilizzo di questa applicazione anche al di fuori dei campi sopra descritti.

1. **Glossario e Acronimi**

|  |  |
| --- | --- |
| **GLOSSARIO / ACRONIMO** | **DESCRIZIONE** |
| DNS | Domain Name System, è un sistema utilizzato per assegnare nomi ai nodi della rete. Serve a semplificare le connessioni a livello umano, infatti ci connettiamo a [www.google.it](http://www.google.it) e non all’indirizzo IP associato di Google. |
| Certificato SSL | Il Certificato Secure Socket Layer è un protocollo di comunicazione che serve a proteggere la corrispondenza che passa tra due nodi attraverso la reti, dal punto A al punto B, senza che nel tragitto i dati possano essere intercettati o manipolati. |
| Indirizzo IP | L’Internet Protocol Address indica un indirizzo univoco di un dispositivo (PC, smartphone, tablet, ecc) all’interno di una rete locale o della rete globale. Serve quindi ad identificare quel dispositivo nella rete. |
| SOC | Un Security Operations Center è un centro da cui vengono forniti servizi finalizzati alla sicurezza dei sistemi informativi di una azienda interna o esterna |
| Dominio | Un insieme di dispositivi che condividono risorse di rete e database e che vengono amministrati come se fossero una entità unica con regole e procedure o aggiornamenti comuni viene definito Dominio |
| API | Le Application Programming Interface sono set di definizioni e protocolli con i quali vengono realizzati e integrati software e applicativi diversi. |
| Modulo PIP | Python Index Package, serve ad installare moduli di terze parti in modo semplice e veloce tramite semplici comandi |
| Librerie | Insieme di funzioni o strutture dati predefinite e predisposte per essere collegate a un programma o un software specifico, scaricabili tramite Modulo PIP |
| Funzione | Costrutto sintattico all’interno del quale è possibile raggruppare una sequenza di istruzione che può essere richiamata più volte all’interno di un software per non dover riscrivere più e più volte lo stesso pezzo di codice |
| Open Source | E’ un metodo di distribuzione del software e consiste nel pubblicare interamente il codice sorgente del software in modo che sia visibile a tutti e che possa essere riutilizzato e anche che si possa favorire la collaborazione tra creatore e utenti per poter migliorare le funzionalità o la stabilità del software, o per la eventuale segnalazione di falle di sicurezza. |
| Try - Except | Metodologia di sviluppo del codice che consente di eseguire delle determinate azioni (try), finchè non si verifica un errore generico o uno specifico errore, al quale verrà eseguita la parte di codice nella sezione except, in modo da poter diversificare ed identificare le tipologie di errore. |

1. **METODOLOGIE DI LAVORO E SVILUPPO DEL CODICE**

Esistono diversi metodologie di sviluppo del software, ovvero delle suddivisioni dei processi di sviluppo che servono ad ottimizzare le varie fasi dello stesso progetto ed è importante conoscere le varie tipologie per scegliere quella che più si può adattare al tipo di progetto e al tipo di team che lavora al progetto.

Le metodologie più diffuse sono 2 e sono frutto dell’evoluzioni di molti più modelli e sono: Modello a Cascata, Modello a Spirale

**Metodo a Cascata (Waterfall)**

E’ un modello di processo di sviluppo software con un flusso sequenziale lineare, ovvero con una fase che inizia solamente dopo il completamento totale della fase precedente, senza alcuna sovrapposizione delle fasi. L’intero processo è quindi diviso in più fasi chiaramente distinguibili tra loro dove la fine di una fase coincide con l’inizio della fase successiva. Ogni fase viene ben documentata data la scelta di finalizzare completamente ogni singola fase ed è perciò facile per chi si occuperà della fase successiva avere tutte le informazioni riguardanti la fase precedente.

|  |  |
| --- | --- |
| PRO | CONTRO |
| Semplice e Facile da capire  Documentazione prodotta completa  Facilità nella suddivisione dei lavori tra componenti del Team  Utile per piccoli progetti con requisiti chiari | Non adatto a progetti mutevoli  In caso di ritardo di un singolo modulo, tarderà tutto il progetto  Il Team deve essere compatto e con la stessa visione del progetto finale |

**Metodo a Spirale**

E’ un modello di processo di sviluppo software dove si fa molta attenzione all’analisi del rischio e al continuo miglioramento del progetto e implementazione di nuove funzioni, adatto quindi a progetti che richiedono una attenzione continua o in cui si vogliono implementare nuove funzioni o anche solo apportare modifiche grafiche.

|  |  |
| --- | --- |
| PRO | CONTRO |
| Continuo controllo dei rischi e delle eventuali modifiche  Adatto a progetti mutevoli nel tempo  Consente continue modifiche sia a livello applicativo che di sicurezza  Ogni componente del Team può comprendere ed apportare modifiche | Richiede Team più esperti dove tutti sono a conoscenza di ogni singolo passaggio del progetto  Richiede molto tempo  Produce una documentazione non completa inizialmente |

1. **Metodo di approccio alla richiesta e soluzione scelta**

Il primo passo per la risoluzione di questo progetto è stata sicuramente seguire dei corsi online di Python e della libreria TKinter per poter gestire la parte grafica della applicazione.

In seguito al corso si è fatta una analisi del progetto e delle richieste che esso richiedeva per poter analizzare il problema e scomporlo nei più piccoli problemi possibili in modo da poterli risolvere di volta in volta con maggiore efficienza e semplicità.

Le richieste iniziali sono state principalmente tre:

* Un modulo che si occupasse di identificare la provenienza di un Indirizzo IP (interno o esterno) inserito da tastiera e la classe o le informazioni pubbliche dello stesso.
* Un database per la gestione manuale delle informazioni riguardanti un particolare Indirizzo IP, informazioni quali:
  + Indirizzo IP
  + Cliente
  + Dominio
  + Dispositivo
  + Servizi Esposti
  + Eventuali Note Aggiuntive

Grazie a queste informazioni è iniziato il lavoro, integrato poi in seguito con la gestione anche grafica del programma, dato l’anticipo con cui si è deciso di iniziare il progetto.

In seguito sono poi state aggiunte nuove richieste opzionali per il progetto che sono state aggiunte in seguito ovvero:

* DNS Resolver
* SSL Scanner
* Mail Scanner
* DownDetector
* Network Port Scanner

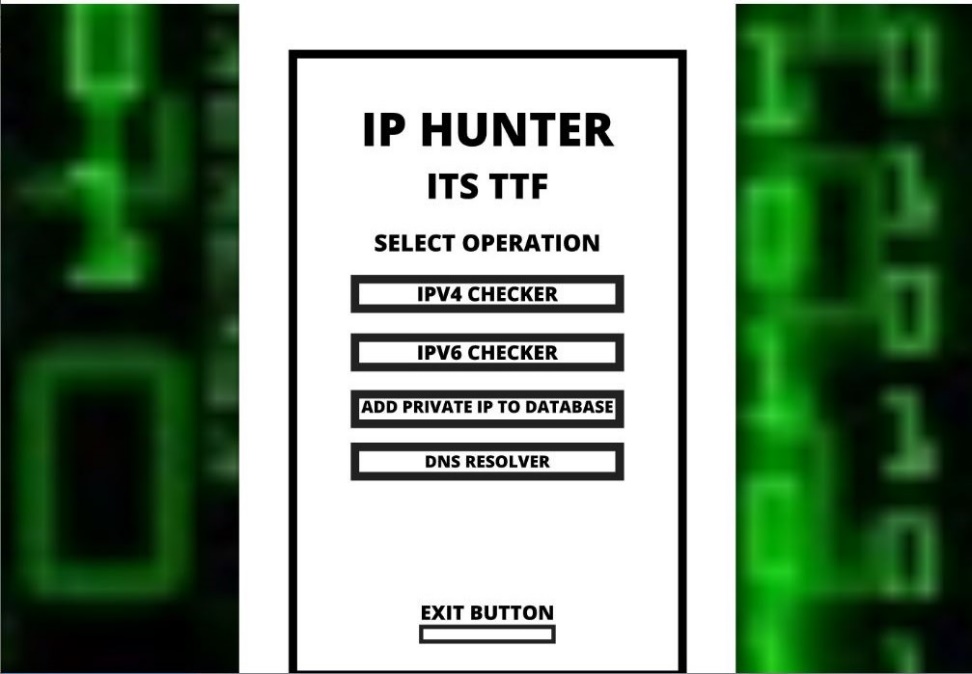
Per la risoluzione di questo progetto la metodologia di sviluppo scelta è stata sicuramente più tendente al metodo a spirale date le scarse conoscenze iniziali del linguaggio di programmazione Python utilizzato e date anche le aggiunte di features in corso di programmazione e anche la possibilità futura di poter ulteriormente ampliare le features del programma in questione.

L’unica pecca di questo metodo in questo caso particolare è stata proprio la mancanza di una documentazione preliminare dalla quale attingere per avere informazioni sul progetto, ma essendo un progetto in continua evoluzione, la documentazione si è piano piano creata nel tempo anche in base alle esigenze dei vari moduli e features aggiunte con il passare del tempo.

Un vantaggio di questo metodo e la continua messa in discussione del lavoro svolto, delle features aggiunte e la possibilità di poter continuare ad effettuare test di vario tipo per apportare modifiche di sicurezza e di stabilità del programma di volta in volta senza una vera e proprio sessione totalmente dedicata alla risoluzione di questi problemi.

1. **Mockup**

Il mockup può essere considerata una realizzazione a scopo puramente illustrativo ed espositivo del progetto che si sta creando, serve quindi a presentare in veste grafica il risultato finale in tutta la sua bellezza ed è diverso da un prototipo proprio perché si basa su qualcosa di non ancora realizzato a cui però si punta, facendo intendere quali sono le intenzioni funzionali e anche estetiche del progetto sin dall’inizio. In questo caso il mockup è stato sviluppato dopo aver valutato anche le possiblità reali del progetto in modo da non distaccare troppo il modello del mockup da quello reale, in modo quindi da rendere il mockup il più possibile simile a un effettivo prototipo del progetto, non avendo competenze e conoscenze nella realizzazione di interfacce grafiche.



Esempio pagina di Mockup

Per poter vedere il Mockup originale completo, seguire questo link qua sotto:

<https://prezi.com/uymwxc7hs6-f/?utm_campaign=share&utm_medium=copy>

1. **Diagramma di Flusso**

Il diagramma di flusso è una rappresentazione grafica molto basilare delle funzioni di un software che si sta creando e di quali siano le interazioni tra le varie parti del software. Serve a definire bene i ruoli e dei “sottoprogetti” che possono essere assegnati a gruppi di lavoro diversi o ad organizzare meglio le tempistiche del progetto suddividendo il problema principale in tanti piccoli problemi di facile risoluzione ed inoltre serve ad aiutare il gruppo stesso di lavoro per capire meglio le interazioni che possono esserci tra le varie componenti del progetto.

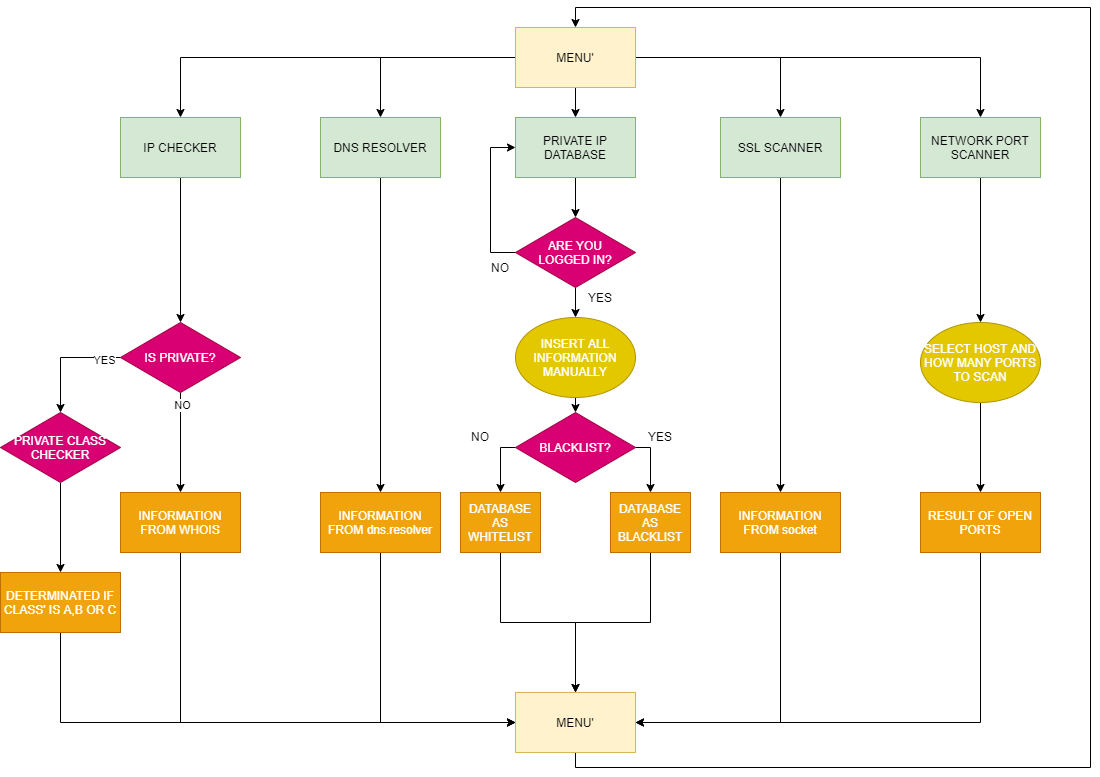


Diagramma di Flusso - 360° Scanner

In questo caso il progetto si basa su un diagramma di flusso iniziale, modificato in seguito ad aggiunte di features e di interazioni tra i vari moduli.

E’ infatti importante considerare il diagramma di flusso come il punto di partenza di ogni progetto ma anche di ogni singola modifica perché ci aiuta a tenere sotto controllo tutti i moduli del progetto, le loro interazioni e le nuove features grazie ad una visualizzazione grafica, più veloce ed immediata rispetto alla visione diretta del codice sorgente del progetto in questione.

Nello specifico il progetto si suddivide quindi in 5 macro funzioni ovvero:

* ***IP Hunter & Checker:***

Dato un indirizzo IP, viene scansionato tramite la libreria netaddr e la libreria ipaddress per poter verificare che sia un indirizzo IP valido e per verificare che sia esterno o interno, e nel caso fosse interno, grazie all’utilizzo di variabili di stato, può essere inoltre identificata la classe di indirizzo IP a cui appartiene l’indirizzo IP dato in input.

In caso di indirizzo IP esterno vengono interrogati i siti di whois e di abuseipdb per poter avere tutte le informazioni possibili sull’indirizzo IP testato come ad esempio il paese, la posizione, i report totali e le segnalazioni degli utenti rispetto a quell’indirizzo IP.

* ***DNS Resolver:***

E’ un modulo molto semplice che permette tramite la libreria socket e dns.resolver di poter risalire all’indirizzo IP di un sito o di un nome DNS inserito in input, e integrato con il modulo SSL Scanner spiegato sotto, permette di poter identificare anche la veridicità e il potenziale rischio di un sito/indirizzo IP.

* ***Private IP Database:***

Questo modulo si occupa di immagazzinare tutte le informazioni inserite manualmente dall’analista che vuole identificare un indirizzo IP privato in whitelist o blacklist, in base a ricerche effettuate sul dispositivo, alle porte aperte e quindi ai servizi esposti. Per poter accedere a questo modulo è necessario essere utenti registrati in un database a parte che gestisce unicamente gli accessi al database contenente gli indirizzi IP.

* ***SSL Scanner:***

Permette di scansionare un sito web tramite il modulo socket per poter verificare la presenza ed applicazione di un certificato SSL al sito e per quindi verificare la sicurezza (o presunta tale) di quel sito. Fornisce anche il tipo di versione di certificato che si sta verificando.

* ***Network Port Scanner:***

Grazie a quest’ultimo modulo possiamo selezionare un host interno della rete e scansionare un numero a scelta, a partire dalla prima porta, di porte dell’host, per verificare quindi a quali pericoli il device è esposto e se è tutto coincidente con i servizi esposti che mette a disposizione o di cui si serve.

1. **Tabella delle versioni**

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | VERSIONE | DESCRIZIONE | CAP. /SEZ. MODIFICATI |
| 26/08/2020 | 0.08.26.1 | First application |  |
| 27/08/2020 | 0.08.27.4 | First test | Gestione del try-except per gestione errore inserimento indirizzo IP |
| 28/08/2020 | 0.08.28.2 | Multiframe | Gestione dei sottoframe per visualizzazione delle informazioni |
| 31/08/2020 | 0.08.31.1 | Whois, abuseipdb | Implementazione delle API di whois ed AbuseIPDB |
| 06/09/2020 | 0.09.06.11 | Menu | Creazione menù per gestione delle varie features aggiunte |
| 09/09/2020 | 0.09.09.4 | Dns resolver | Implementazione e test DNS resolver |
| 11/09/2020 | 0.09.11.2 | Window & credits | Aggiunta tasto credits e modifica gestione delle finestre |
| 12/09/2020 | 0.09.12.1 | Clear function | Creazione funzioni per pulizia del codice |
| 13/09/2020 | 0.09.13.14 | Graphic database | Creazione parte grafica di gestione database |
| 14/09/2020 | 0.09.14.2 | Database | Gestione del database tramite utente non loggato |
| 16/09/2020 | 0.09.16.1 | Login/registration | Pagina di login/registrazione e database login |
| 17/09/2020 | 0.09.17.1 | Optimization & functions | Ottimizzazione codice in funzioni |
| 18/09/2020 | 0.09.18.3 | Network Port Scanner | Implementazione gestione network port scanner |
| 21/09/2020 | 0.09.21.1 | SSL Scanner | Implementazione SSL Scanner |
| 25/09/2020 | 1.09.25.8 | Optimization & button | Ottimizzazione codice e posizionamento dei pulsanti e aggiunta commenti mancanti |

1. **Snippet e GitHub**

# ---------------------------------------------------------------------  
# --------------------------SSL SCAN WINDOWS---------------------------  
# ---------------------------------------------------------------------  
  
 def quartotasto():  
 def sslscan():  
 hostname = str(entry5.get())  
  
 context = ssl.create\_default\_context()  
 try:  
  
 with sk.create\_connection((hostname, 443)) as sock:  
 with context.wrap\_socket(sock, server\_hostname=hostname) as ssock:  
 print(ssock.version())  
 print(“Site " + hostname + " use a \n" + ssock.version() + " certificate)  
  
 except :  
 print("Site not valid or SSL not on site")

Nel caso specifico di un progetto Open Source, uno snippet si può identificare come un semplice frammento di codice che serve all’utente finale per capire meglio il funzionamento del codice senza però visualizzarlo tutto o anche solo per capire in che modo è organizzato il codice. E’ importante perciò selezionare una parte ben specifica del codice.

In questo caso si tratta della sezione di codice riguardante la scansione di un sito sulla porta 443, ovvero la porta che serve normalmente il protocollo HTTPS, per verificare la presenza o meno di un certificato valido per il servizio. La sezione di codice differisce da quella originale per il fatto che non comprende la parte di gestione grafica dell’applicazione ma gestisce solamente l’applicazione da riga di comando.

Per poter verificare il codice completo è possibile andare su GitHub, un servizio di hosting per progetti software che consente l’aggiornamento del software con diverse versioni anche da parte di utenti esterni, molto utile per quanto riguarda i software open source aperti anche alla modifica del codice direttamente da altri utenti. Ovviamente queste modifiche vanno approvate dal creatore di quel particolare progetto o software, ed è perciò impossibile manipolare in modo malevolo il codice, se non per una svista del proprietario iniziale del codice.

Nel link qua sotto è possibile visualizzare il codice completo con integrazione della parte grafica su GitHub:

<https://github.com/jinxo71i/IP-Hunter---Cybaze-Project-Work>

1. **TEST ESEGUITI E LIMITAZIONI**
2. **Tipologia di test**

Le tipologie di test effettuati sono principalmente due:

* **Test di progettazione**
* **Test di usabilità**

Il test di progettazione è un test che si basa sulla buona riuscita o meno dei singoli moduli, si tratta quindi di verificare il corretto funzionamento in condizioni ideali della macchina e verificare che non ci siano errori di codice che permettono il corretto funzionamento del progetto seguendo lo standard e la circolarità definita dal diagramma di flusso descritto poco sopra.

Il test di usabilità invece serve a verificare tutti i possibili errori in condizioni non ideali, quindi ad esempio con mancanza di connessione internet, con una digitazione errata degli indirizzi IP o dei domini da testare, e verificare che ogni possibile errore sia stato corretto o reindirizzato e che questo non porti l’applicazione in uno stato di errore dal quale non si può uscire, dovendo quindi riavviare l’applicazione. Ogni singolo blocco del diagramma di flusso contiene infatti una risoluzione dei problemi o, in molti casi, anche solo una segnalazione di un errore, ad esempio un nome utente non registrato che quindi non può accedere al database, o un indirizzo IP scritto in modo sbagliato che quindi non è possibile testare dato che è inesistente.

Sono stati svolti molti test per verificare ogni possibile utilizzo, consono o meno, dell’applicazione, in modo da poter verificare che il programma sia in grado di gestire ogni situazione senza il rischio di andare in crash o in down in nessuna situazione, requisito fondamentale per un programma che deve essere utilizzato principalmente all’interno di un SOC, dove la velocità nell’avere risultati e la stabilità è il requisito principale e fondamentale su cui si fonda il lavoro di un analista che lavora in un SOC.

In molti casi si sono create volutamente delle situazioni estreme sia a livello di codice che a livello di utilizzo dell’applicazione proprio per visualizzare e gestire la maggior parte di degli errori possibili e creare così delle eccezioni e delle risoluzioni per quei particolari errori.

I tipi di errori più comuni da risolvere sono stati in ogni modulo l’inserimento di indirizzi IP e di domini web errati nella sintassi, che quindi non permettevano una corretta interazione con il funzionamento del modulo stesso. Ognuno di questi elementi viene gestito con il metodo try-except (descritto nel glossario), in modo da poter gestire ogni singolo tipologia di errore.

Altri errori sono stati riscontrati soprattutto nella gestione delle tabelle di un database e in caso di conflitto nell’inserimento di una primary key all’interno di un database che fosse già presente nella tabella del database, problemi risolti con semplici modifiche di righe di codice utilizzando dei comandi appositi per la gestione di queste situazioni grazie alla libreria SQLite3.

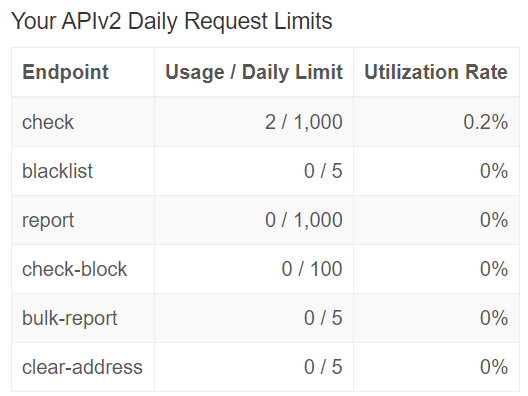
I test finali sono stati effettuati utilizzando una serie di indirizzi IP da testare in un file .txt consegnato dal docente Pietro Melillo, in modo da verificare indirizzi IP non noti e con cui si sarebbe potuto incorrere in problemi ed errori e per alcuni di questi indirizzi IP sono stati segnalati effettivamente degli errori non ancora gestiti tramite try-except dal programma e c’è stato quindi modo di modificare nuovamente il codice per gestire quelle eccezioni.

1. **Limitazioni imposte esterne**

Utilizzando delle librerie esterne o delle API si può incorrere in limitazioni all’utilizzo delle stesse a meno che non vengano eseguiti pagamenti una tantum o ricorrenti mensilmente.

In questo caso per evitare una delle limitazioni si è deciso di apportare delle modifiche al progetto nel momento del suo sviluppo iniziale, ovvero di utilizzare una richiesta web per poter avere informazioni per quanto riguarda il modulo whois, in modo da non avere la limitazione mensile di interrogazioni del modulo.



Per quanto riguarda AbuseIPDB invece esistono delle limitazioni di mille interrogazioni al giorno, più che sufficienti per progetti di questo tipo, ma è giusto segnalare nel caso si volesse usare una singola API per più persone che utilizzano contemporaneamente il programma come può succedere ad esempio all’interno di un SOC. Per utlizzare il modulo è necessario quindi essere in possesso di una Key per le API di AbuseIPDB, chiave che viene garantita e consegnata dopo una veloce registrazione sul sito di AbuseIPDB.

IPVoid invece, inizialmente implementato nel progetto, è stato eliminato dal codice poiché dava poche informazioni in più rispetto ad AbuseIPDB e perché comunque non garantisce un minimo giornaliero o mensile che viene poi azzerato, ma solamente circa 300 interrogazioni iniziali, dopodichè bisogna necessariamente pagare per poter aggiornare il numero dei propri crediti necessari ad interrogare le API del sito. E’ perciò prevista la sua implementazione come commento nel codice, ma di base non viene attivata come funzionalità.

Una delle limitazioni più grandi, in attesa di un aggiornamento del programma, è che il programma può utilizzare un database condiviso ma non è in grado di modificare degli indirizzi IP già presenti in un database, features utile per non dover aprire un database con gli appositi programmi e per poter gestire eventuali doppi accessi da più postazioni sullo stesso indirizzo IP.

1. **CONCLUSIONI**
2. **Sviluppi e features future**

Il progetto è stato sviluppato in poco più di un mese e viene presentato come una bozza di qualcosa che verrà integrato poi con altri moduli e a cui si vogliono aggiungere nuove features come ad esempio:

* Aggiunta di un IP privato al database direttamente dalla pagina di IP Hunter & Checker
* DNS Resolver che mostra più di un risultato nel caso il sito si appoggiasse a più indirizzi IP
* Possibilità di modificare i vari valori di un indirizzo IP già inserito nel database anche tramite l’interfaccia grafica della app, senza quindi dover utilizzare un programma apposito per la gestione dei database
* Implementazione del mail scanner (che finora presenta il problema di essere solo ed unicamente a pagamento)
* Inserimento del tasto ‘abort scan’ per poter interrompere le scansioni delle porte aperte di un host. Ora infatti non è possibile se non tornando indietro nel menù, perdendo però i risultati ottenuti fino a quel momento
* Scelta di range di porte da controllare, in questo momento il controllo si applica sempre dalla prima porta al numero di porte che si desidera in input, ma essendo le porte più di 60000, si potrebbe voler testare una porta specifica o un range di porte specifico per poter snellire il programma e velocizzare il processo di controllo.
* Gestione temporanea degli accessi, anche una volta riavviata l’applicazione rimane in cache l’accesso effettuato, consentendo così all’utilizzatore della app di non doversi loggare ad ogni avvio della app.
* Creazione di una vera e propria applicazione Android
* Creazione di un sito con integrata l’applicazione per utilizzo diretto da browser web tramite Web Framework FLASK.
* Implementazione di un algoritmo di hashing e di salt per la gestione delle password nel database
* Creazione dell’autenticazione a 2 o più fattori (mail o numero di telefono), per poter accedere alla sezione di modifica del database
* Invio tramite mail di ogni operazione effettuata e di ogni risultato di ogni operazione effettuata
* Prevista la creazione dello stesso identico progetto ma senza un supporto per quanto riguarda la parte grafica, utilizzando quindi solamente la riga di comando dal momento che il programma verrà principalmente utilizzo da analisti di un SOC e quindi da persone già in possesso delle conoscenze utili per poter utilizzare i vari moduli da riga di comando senza dover per forza essere in possesso di una interfaccia user-friendly.

1. **Conclusioni**

Il progetto risulta ancora in fase di sviluppo nonostante sia arrivato il termine per poterlo sviluppare all’interno del project work. Molte delle implementazioni mancanti sono dovute anche alla mia personale impossibilità di lavorare nell’ultima settimana di project work ma il progetto rimane online e disponibile in modalità open source per poter apportare modifiche nel tempo e per poter accettare collaboratori esterni o modifiche consigliate e per rendere pubblico il codice in modo da poter più facilmente identificare falle di sicurezza o di funzionamento della applicazione.

Il progetto viene perciò presentato in una sua primissima versione, con l’intenzione però già dalle prossime settimane di implementare a livello di funzionalità e di miglioramento e ottimizzazione del codice molte delle features già presenti e soprattutto molte di quelle previste nel punto precedente, sempre utilizzando il metodo principale agile utilizzato finora in modo da poter garantire lo sviluppo più velocemente anche se con più possibilità di presentare un programma che necessità di molte modifiche poi in seguito.

Il progetto ha inoltre un sito dedicato per la presentazione del progetto in modo rapido ed efficace, al momento disponibile presso [www.lucalonghi.cloud](http://www.lucalonghi.cloud), dove vengono spiegate in modo rapido le varie funzioni dell’applicazione, le implementazioni future, il collegamento a GitHub per il download e la possibilità di contattarmi direttamente tramite mail per eventuali segnalazioni, accorgimenti o consigli.

Il sito è sviluppato con linguaggio html e markdown ed è un sito statico hostato anche esso su github(pages), la deploy del sito viene effettutata su Netlify gratuitamente e collegato poi tramite DNS a un dominio di mia proprietà. In questo modo ci si può creare un sito statico utile a questo tipo di siti o anche ad altri in modo totalmente gratuito se non si vuole utilizzare un nome di dominio personalizzato che altrimenti costerebbe comunque non più di una quindicina di euro annui.

1. **CREDITI E SITOGRAFIA**

Il progetto è stato realizzato grazie al project work organizzato da ITS Tech Talent Factory e l’azienda Cybaze, che ha messo a disposizione i docenti Pietro Melillo e Andrea Vercesi come supervisori del progetto, e grazie anche alla collaborazione e allo scambio reciproco di informazioni ed idee con il “Team E” (Giulio Monaco, Davide Marino e Ashan Perera).

Per la realizzazione del progetto sono state utilizzate molteplici fonti di informazione e documentazione riportate qui sotto nella sitografia:

* [Corso base Python](https://www.youtube.com/watch?v=rfscVS0vtbw&t=10576s&ab_channel=freeCodeCamp.org)
* [Corso interfaccia grafica Python](https://www.youtube.com/watch?v=YXPyB4XeYLA&t=9784s&ab_channel=freeCodeCamp.org)
* [StackOverflow](https://stackoverflow.com/)
* [SimplifiedPython](https://www.simplifiedpython.net/)
* [Reddit](https://www.reddit.com/r/Python/)
* [Udemy](https://www.udemy.com/)
* [Netlify (per il sito web)](https://www.netlify.com/)
* [Cloudinary](https://cloudinary.com/)

