

Università degli Studi di Perugia

Dipartimento di Matematica e Informatica



Corso di

Pianificazione della Sicurezza Informatica e
Informatica Forense

Progetto:

***"Recupero user_data di un utenza Facebook avendo
username/password e successivo invio a database con
relative informazioni garantendo sicurezza dei dati"***

Studenti

Simone Ojetti
Matteo Riganelli

Prof.

Alfredo Milani
Emanuele Florindi

a.a. 2015/2016



Assegnazione

Facebook. Sviluppare un modulo software lato server che consenta all'utente di fornire login e password di una utenza facebook, in risposta il sistema server dovrà: a) scaricare informazioni dal profilo (si consideri la possibilità di specificare quali informazioni si desiderano) b) generare un report della transazione con id unico e informazioni significative c) generare un file zip contenente report e dati scaricati, d) inserire nel proprio database l'hash relativo al file zip e inviare/rendere disponibile all'utente sia il l'hash che un link di download.

Sommario

Il progetto affronta l'estrazione e la memorizzazione dei dati di un account facebook essendo già a conoscenza del nome utente e della password dell'account del quale si vuole effettuare l'estrazione.

Con l'avvento dei *Social Network* ogni giorno viene generata e memorizzata una quantità enorme di informazioni. I *Social Network* rappresentano quindi una fonte potenzialmente infinita di dati degli utenti, i quali possono essere sfruttati sia a scopi scientifici sia a scopi commerciali.

I *Social Network*, al fine di permettere lo sviluppo di un ecosistema di applicazioni attorno alla propria piattaforma, mettono a disposizione delle interfacce per permettere l'accesso ai dati dei propri utenti (nel rispetto della loro privacy). Tuttavia, poiché essi fondano il loro stesso modello di business su tali dati, ne consegue che le modalità di accesso ai dati e la quantità di dati estraibili dai *Social Network* è molto ridotta.

In questa tesina si affrontano dapprima le modalità di accesso ai dati del Social Network Facebook e successivamente si affronta una discussione sulla modalità di estrazione di questi ultimi. Vengono inoltre valutati gli aspetti positivi e negativi delle varie tecniche, che porteranno alla realizzazione del software lato server per la creazione del servizio.

Si evidenziano perciò le limitazioni quantitative e qualitative dei corrispettivi meccanismi, illustrando infine la realizzazione della scelta effettuata con esempi.

1 Introduzione

L'estrazione dei dati dai *Social Network* è un argomento che suscita molto interesse. Infatti, tali informazioni offrono molte opportunità di ricerca e analisi. Di seguito si presentano i metodi per l'estrazione di informazioni da siti, in particolare il *Social Network Facebook* e la loro successiva analisi. In questo progetto sono state esplorate le API di Facebook, mostrandone il funzionamento, tramite esempi pratici realizzati nel linguaggio PHP con l'ausilio del Facebook-PHP-SDK. Prima di addentrarsi nell'esplorazione del meccanismo utilizzato in questo progetto per l'accesso ai dati è indispensabile presentare i concetti basilari dei *framework* su cui i meccanismi studiati sono costruiti.

2 Estrapolare i dati

Per poter estrapolare i dati da uno dei Social Network più famosi del mondo le tecniche che possiamo adottare sono essenzialmente di due tipi: basarsi sulle funzioni che ci offre Facebook stessa, mantenendo quindi alto rigore alle regole stabilite dai produttori; oppure basarsi su tecniche di scraping aventi una base illegale perché all'insaputa dell'utente, queste simulano la una navigazione in incognito nel World Wide Web.

2.1 API

Con il termine *Application Programming Interface* (API) si indica un insieme di procedure che vengono rese disponibili e permettono di formare un insieme di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità delle API, consiste nel fornire un livello di astrazione tra un servizio e il suo fruitore. Rappresentano quindi un'interfaccia di programmazione. Tale interfaccia ha lo scopo di permettere ad altre entità (e. g., librerie, software, utenti) di compiere un insieme di azioni su una determinata piattaforma di cui non si conoscono i dettagli implementativi. Per tale motivo, le API vengono spesso fornite per permettere oltre che l'utilizzo di un determinato servizio, anche la sua estensione da parte di altri attori.

Mettere a disposizione delle API di un software significa dare ad altri la possibilità di interagire con la piattaforma di tale software e, eventualmente, di estendere le funzioni e le caratteristiche della sua struttura base. In altri termini, le API sono lo strumento primario utilizzato per permettere l'interazione ad alto livello con i software (o, generalmente, con un'implementazione a più basso livello). Tutti i maggiori *Social Network* esistenti forniscono, infatti, delle API. Quando usate nel contesto *Web*, le API sono tipicamente definite come un insieme di possibili richieste HTTP che restituiscono un messaggio di risposta con una struttura ben definita (XML o JSON, solitamente).

PROFILE

Array ([name] => Matteo Riganelli [first_name] => Matteo [last_name] => Riganelli [birthday] => Facebook\GraphNodes\Birthday Object ([hasDate] => 1 [hasYear:Facebook\GraphNodes\Birthday:private] => 1 [date] => 1990-12-23 00:00:00.000000 [timezone_type] => 3 [timezone] => Europe/Berlin [location] => Array ([id] => 114759761873412 [name] => Mumbai, India) [hometown] => Array ([id] => 107852709248187 [name] => Perugia, Italy) [school] => Array ([id] => 106405179396153 [name] => Liceo Classico A. Mariotti) [type] => High School [id] => 1641985876126030) [1] => Array ([id] => 111950378821457 [name] => University of Mumbai) [type] => College [id] => 1641985962792688)) [religion] => Cristianesimo () [gender] => male)

SPECIFIC-PROFILE

-ID_PROFILO: 1638938933097391

-EMAIL: matteoriganelli@gmail.com

-NOME_COMPLETO: Matteo Riganelli

-NOME-Matteo

-COGNOME-Riganelli

-SESSO-male

-DATA-NASCITA: 23-12-1990

-SITO-WEB: <http://www.provasito.com/>

-CITTA-ATTUALE: Mumbai, India

-HOMETOWN: Perugia, Italy

-RELIGIONE: Cristianesimo

-STUDI

Liceo Classico A. Mariotti
University of Mumbai

array(2) { [0] => array(3) { ["school"] => array(2) { ["id"] => string(15) "106405179396153" ["name"] => string(26) "Liceo Classico A. Mariotti" } ["type"] => string(16) "High School" [id] => string(16) "1641985876126030" } [1] => array(3) { ["school"] => array(2) { ["id"] => string(15) "111950378821457" ["name"] => string(20) "University of Mumbai" } ["type"] => string(8) "College" [id] => string(16) "1641985962792688" } }

Figura 1: Esempio estrapolazione dati utilizzando le API Facebook

Essendo lo strumento apparentemente più idoneo per l'estrazione dei dati del Social Network Facebook, c'è da aggiungere che per poter effettuare tali operazioni, si ha la necessità di creare un account sviluppatore nel sito (<https://developers.facebook.com/>) e associare la richiesta dei dati ad un AppID/AppSecretID associate allo sviluppatore. Alla richiesta dei dati, inoltre l'account obiettivo, verrà sollecitato all'approvazione di lettura dei dati dall'applicazione, perciò tale tecnica, previa opportune considerazioni, offre già delle limitazioni importanti se non disponiamo delle autorizzazioni da parte dell'utente. Inoltre lavorando con certi tipi di dati sensibili, nella maggior parte dei casi vi è la necessità di un approvazione da parte di Facebook.

2.2 Scraping

Il web scraping (detto anche web harvesting o web data extraction) è una tecnica informatica di estrazione di dati da un sito web per mezzo di programmi software. Di solito, tali programmi simulano la navigazione umana nel World Wide Web attraverso l'implementazione di basso livello dell'Hypertext Transfer Protocol (HTTP) o l'incorporamento di un vero e proprio browser, come Internet Explorer o Mozilla Firefox. Il web scraping è strettamente correlato all'indicizzazione dei siti Internet; tale tecnica è attuata mediante l'uso di bot dalla maggior parte dei motori di ricerca. D'altro canto, il web scraping si concentra di più sulla trasformazione di dati non strutturati presenti in Rete, di solito in formato HTML, in metadati che possono essere memorizzati e analizzati in locale. Il web harvesting è altresì affine alla web automation, che consiste nella simulazione della navigazione umana in Rete attraverso l'uso di software per computer. Il web scraping si può usare per confrontare prezzi online, monitorare dati meteorologici, rilevare modifiche in un sito internet, nella ricerca scientifica, per il web mashup e il web data integration.

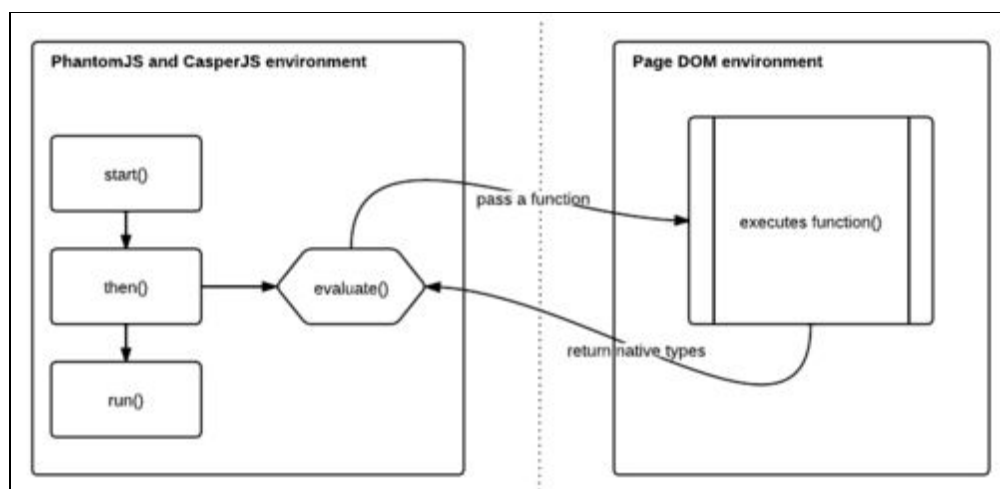


Figura 2: Schema rappresentativo funzionamento Scraping

2.2.1 Facebook Scraping

Facebook adotta una politica estremamente difensiva relativamente allo *scraping* dei suoi dati, adottando tecniche molto avanzate di rilevamento dei *crawler* e dei processi di *scraping*. Il *software* di difesa genera automaticamente dei codici di verifica nel caso ritenga, dall'analisi dei *log* e dei percorsi di navigazione, che il visitatore non sia un umano bensì un *crawler*. Tale *software* opera anche controlli sull'indirizzo IP al fine di verificare che sia incluso in una lista di indirizzi IP fissi cui hanno concesso (chiaramente con contratto commerciale) l'esecuzione di processi di *scraping*.

Connettendosi all'URL '<https://facebook.com/robots.txt>' si ha conferma di tale politica. Infatti, il file *robots.txt* vieta a qualsiasi User Agent, che non sia baiduspider, Googlebot, msnbot (e altri omessi), di accedere a qualsiasi percorso.

```
User-agent: baiduspider
Disallow: /ac.php
Disallow: /ae.php
Disallow: /ajax/
Disallow: /album.php
Disallow: /ap.php
Disallow: /autologin.php
Disallow: /checkpoint/
Disallow: /contact_importer/
Disallow: /feeds/
Disallow: /l.php
Disallow: /o.php
Disallow: /p.php
Disallow: /photo.php
Disallow: /photo_comments.php
Disallow: /photo_search.php
Disallow: /photos.php
Disallow: /sharer/
```

```
User-agent: Googlebot
Disallow: /ac.php
Disallow: /ae.php
Disallow: /ajax/
Disallow: /album.php
Disallow: /ap.php
Disallow: /autologin.php
Disallow: /checkpoint/
Disallow: /contact_importer/
Disallow: /feeds/
Disallow: /l.php
Disallow: /o.php
Disallow: /p.php
Disallow: /photo.php
```

Disallow: /photo_comments.php
Disallow: /photo_search.php
Disallow: /photos.php
Disallow: /sharer/

User-agent: msnbot
Disallow: /ac.php
Disallow: /ae.php
Disallow: /ajax/
Disallow: /album.php
Disallow: /ap.php
Disallow: /autologin.php
Disallow: /checkpoint/
Disallow: /contact_importer/
Disallow: /feeds/

Disallow: /l.php
Disallow: /o.php
Disallow: /p.php
Disallow: /photo.php
Disallow: /photo_comments.php
Disallow: /photo_search.php
Disallow: /photos.php
Disallow: /sharer/

User-agent: *
Disallow: /

Il file robots.txt, come già specificato, di per sé funge da mero indicatore e non esclude quindi alcuna operazione dal punto di vista tecnico. Perciò è teoricamente possibile, prendendo tutte le dovute precauzioni, effettuare lo *scraping* di Facebook semplicemente non tenendo in considerazione il succitato *file*.

La conferma concreta che il processo di *scraping* di Facebook è possibile, seppur molto ostico, deriva direttamente dal fatto che, in letteratura, esistono prove tangibili del fatto che tale processo è stato effettuato e portato a termine con successo in più casi.

3 Progetto

Abbiamo deciso di utilizzare il metodo dello scraping per estrapolare i dati, invece di appoggiarci all'utilizzo delle Api per i seguenti motivi:

Le API godono di molto vantaggi quali:

- disponibilità dei dati
- affidabilità
- sicurezza

però hanno le seguenti caratteristiche negative

- necessitano dell'associazione all'utilizzo di un utente sviluppatore registrato (AppID, AppSecretID)
- autorizzazione da parte dell'utente del quale si richiede accesso ai dati
- autorizzazione per ogni tipologia di richiesta che andiamo ad effettuare.

Mentre tecniche di scraping fondamentalmente ci permettono di ottenere la stessa tipologia di dati, evitando un'autenticazione di qualsiasi tipo dell'utente che effettua la richiesta, infatti è un meccanismo il cui funzionamento simula un browser mosso da un umano e necessita solo delle credenziali dell'utente bersaglio proprietario del profilo facebook che vogliamo analizzare.

Il software comprensivo di tutto il funzionamento comprende un cuore centrale sviluppato lato server mediante linguaggio PHP, Javascript. Accessibile poi lato client mediante apposita pagina HTML, PHP. Di seguito si andrà ad analizzare ogni singola componente.

3.1 Login e Registrazione al portale

L'accesso allo script che effettua l'analisi e lo scraping del profilo è controllato da un sistema di accesso con credenziali; username e password. Questa scelta è stata presa per preservare l'accesso e garantire che solo chi è realmente autorizzato possa effettuare delle scansioni.

Figura 3: Schermata Login al portale per il Dump dei dati

Accedendo al portale la prima pagina che verrà caricata sarà login.php che permetterà, appunto, di inserire le proprie credenziali e accedere alla dashboard per la visualizzazione/effettuare dei report. E' possibile effettuare nuove registrazioni per nuovi utenti mediante form apposito (registration.php) accessibile solo da utenti già autorizzati (account Admin, di default). Un utente per registrarsi potrà mandare una mail

all'amministratore fornendo Username, Password ed email (opzionale) che vorrà utilizzare per la registrazione o potrà richiedere la registrazione ad un utente già registrato. In questo modo la registrazione non sarà aperta a tutti e il portale sarà utilizzabile solo da una determinata cerchia di persone di fiducia. La creazione di un nuovo account è immediata e l'account risulterà subito operativo e funzionante. Una volta completato l'accesso con le giuste credenziali sarà possibile scegliere quali operazioni compiere.

Ogni utente potrà:

- Richiedere una nuova scansione
- Visualizzare il report di tutte le scansioni già effettuate
- Registrare un nuovo utente
- Effettuare il Logout

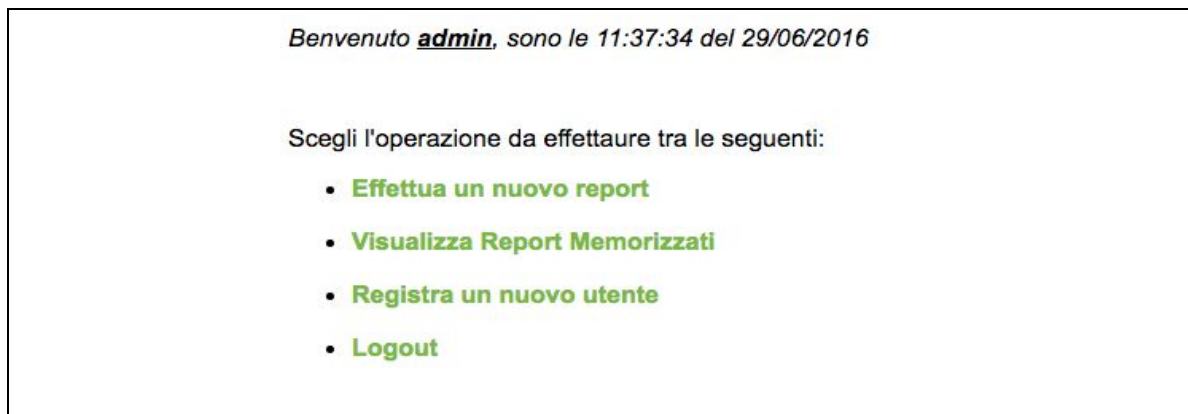


Figura 4: Schermata delle opzioni effettuando l'accesso al portale

Nel primo caso si aprirà la pagina relativa alle scansioni degli account Facebook target (/master/index.php).

Nel secondo caso si aprirà la pagina di gestione dei report (dashboard.php). Le ultime due scelte permettono di registrare un nuovo utente e di effettuare il logout dalla piattaforma.

Per gestire l'utenza, il portale si basa su di un database di nome "register" in particolare la tabella "user" serve a memorizzare le credenziali. Per ogni utenza viene memorizzato:

- ID (chiave primaria)
- username
- email
- password (hash)
- data in cui viene effettuata la registrazione

Id	username	email	password	trn_date
1	Simone	simone@simone.it	47eb752bac1c08c75e30d9624b3e58b7	2016-06-16 16:51:27
2	admin	sjmon3@gmail.com	21232f297a57a5a743894a0e4a801fc3	2016-06-22 10:05:04
3	Simone	adaad@aaa.it	33dcafe50c23b002f05b3b23512886aa	2016-06-23 10:14:55
5	ciccio	ciccio@cio.io	27b4b5b01b0d1fcab2046369720ff75e	2016-06-28 09:29:35

Figura 5: Schemrata tabella utenti del portale (PhpMyAdmin)

La password viene memorizzata non in chiaro, bensì viene memorizzato il suo hash in modo tale che anche chi avrà accesso al database non avrà possibilità di risalire alla password in chiaro scelta dagli utenti.

3.3 Visualizzazione Report

La pagina che visualizza i report effettuati si basa sulla tabella "dump" del database "register". Tale tabella serve per memorizzare i dati di output ottenuti da una scansione. Come già accennato in precedenza al termine della scansione tutti i file ottenuti vengono compressi in uno zip di cui viene calcolato l'hash, tale hash sarà fondamentale per poter verificare l'integrità del pacchetto zip una volta scaricato localmente sulla propria macchina. In questo database verranno memorizzati:

- ReportID (primaria, incrementale)
- NomeCognome
- Data
- Hash
- Link al download

ReportID	NomeCognome	Data	Hash	Link
1	Simone Ojetti	23/6/2016	71f841640ed9a304d8bd36bd89c578a5	http://localhost:8888/registration/master/report_S...
2	Simone Ojetti	23/6/2016	bfaa61e9a0e177cbc066f71651c7adde	http://localhost:8888/registration/master/report_S...

Figura 6: Schemata tabella utenti del portale (PhpMyAdmin)

La pagina mostrerà una tabella a 6 colonne in cui ogni riga corrisponderà ad un preciso report effettuato e completato.

Direttamente da questa pagina sarà possibile visualizzare l'hash del file creato e il link al download (file che risiede localmente nel server che ospita il portale).

Queste sono le operazioni eseguite:

ID	Nome e Cognome	Data	Hash	Download Link
1	Simone Ojetti	23/6/2016	71f841640ed9a304d8bd36bd89c578a5	Clicca per scaricare
2	Simone Ojetti	23/6/2016	bfaa61e9a0e177cbc066f71651c7adde	Clicca per scaricare

Torna alla [home](#) oppure esegui il [logout](#)

Figura 7: Schemata tabella report effettuati

3.4 Effettua Nuovo Report

L'utente cliccando su apposita richiesta, viene reindirizzato ad un apposita pagina web (*master/index.php*), contenente 2 campi per l'inserimento del nome utente e password dell'utente del quale si vuole effettuare report, i checkbox per la scelta di quali informazioni si vogliano recuperare dall'account e il pulsante di invio richiesta del servizio.

Compilati i vari campi l'utente, cliccando l'apposito pulsante permette l'esecuzione di *master/action.php*. Tale file permette essenzialmente due cose principali: controllare l'iscrizione dell'utente al social Network Facebook e in caso positivo controllare quali checkbox sono stati selezionati per poter effettuare lo scraping.

Lo scraping è reso possibile grazie a **CasperJS**.

3.4.1 CasperJS

Poco dopo l'uscita di *PhantomJS* (modulo javascript, browser dipendente, per l'interazione automatica con le pagine web) è stato sviluppato dal Nicolas Perriault scrive *CasperJS*, una suite di librerie la quale estende le capacità del già esistente *PhantomJS*, sempre per l'interazione automatica con le pagine web.

Permette di fornire API Javascript per la navigazione automatica, recuperare screenshot. E' basato su WebKit il che lo rende un ambiente di navigazione simile a Firefox, Google Chrome e gli altri browser. E' un software openSource rilasciato sotto licenza BSD.

Siccome *PhantomJS* e quindi *CasperJS* sono eseguibili senza l'utilizzo di interfaccia utente, è possibile utilizzare dei javascript per poter eseguire attacchi contro siti web.

3.4.2 Installazione

Per poter eseguire gli script è necessario effettuare un'installazione di tutti i componenti necessari citati nel paragrafo precedente.

- *PhantomJS* : agisce come un browser per poter eseguire vari test
- *CasperJS* : scritto per *PhantomJS*, permette di eseguire ulteriori utility per il testing come poter cliccare determinati componenti di una pagina web e eventi di login

Step 1 - Installare PhantomJS

Se si usa NPM è possibile eseguire il seguente comando

```
$ npm install phantomjs -g
```

Altrimenti è necessario scaricare (<http://phantomjs.org/download.html>) effettuando lo spostamento manuale del file *bin/phantomjs* all'interno della propria directory */usr/bin/local*.

Per testare la corretta installazione di PhantomJS eseguire il seguente comando da terminale

```
$ phantomjs -v
```

Questo restituirà la versione del software installata. Scrivendo anche un primo "Hello World" per una corretta verifica del funzionamento, creando una file "test.js" con all'interno la seguente dicitura:

```
console.log("Hello PhantomJS");
```

Eseguendo tale file tramite console con il seguente comando

```
phantomjs test.js
```

Step 2 - Installare CasperJS

Se si usa NPM è possibile eseguire il seguente comando

```
$ npm install casperjs -g
```

Altrimenti è necessario eseguire uno dei seguenti tipi di installazione vedi (<http://docs.casperjs.org/en/latest/installation.html>).

Per testare il corretto funzionamento di CasperJS eseguire il seguente comando da terminale

```
casperjs -version
```

Questo restituirà la versione del software installata. Scrivendo anche qui un primo "Hello World" per una corretta verifica del funzionamento, creiamo una file "casper_test.js" avente come prima dicitura l'istanziamento della classe Casper, la quale permetterà di eseguire il metodo `create()`.

```
var Casper = require('casper').create();
```

Per avviare Casper invece si utilizza il metodo `.start()`, metodo che permetterà di navigare effettivamente all'interno di un sito e poter quindi eseguire i vari metodi di Casper.

```
// casper.start(url, callback);  
casper.start('http://google.com', function({})
```

Infine è possibile fare qualcosa all'interno della pagina, come ad esempio restituire un messaggio contenente il titolo grazie al metodo `getTitle`,

stampando eventualmente il messaggio in vari stili diversi (INFO, ERROR, WARNING, COMMENT

```
casper.start('http://google.com', function(){
    // this.echo(messageToPrint, style)
    this.echo(this.getTitle, 'INFO')
});
```

Infine si esegue il tutto includendo alla fine del file

```
casper.run()
```

Il codice completo è il seguente

```
var casper = require('casper').create();
casper.start('http://google.com/', function() {
    this.echo(this.getTitle(), 'INFO');
});
casper.run();
```

Eseguibile sempre da terminale con il comando

```
$ casperjs casper_test.js
```

3.4.3 Note tecniche

Andando ad accedere via pagina PHP, abbiamo necessità di richiamare apposite funzioni mediante linguaggio PHP, invocando come prima cosa successiva alla richiesta dell'utente l'utilizzo di PhantomJS e CasperJS:

```
putenv("PHANTOMJS_EXECUTABLE=/usr/local/bin/phantomjs");
echo "Running PhantomJS version: ";
echo exec('/usr/local/bin/phantomjs --version 2>&1');
echo "<br />";
echo "Running CasperJS version: ";
echo exec('/usr/local/bin/casperjs --version 2>&1');
```

Questi permetteranno l'invocazione delle due librerie e quindi possiamo richiamare tutti i vari script .js che eseguiranno il recupero delle informazioni.

Verifica Profilo + action.php

Come prima cosa viene effettuato una verifica della correttezza dei dati inseriti, andando a controllare se i dati inseriti corrispondono ad un account veramente esistente, grazie allo script `'verifica_profilo.js'`.

```
$ex = '/usr/local/bin/casperjs verifica_profilo.js '.$username.' '.$password;
```

`$username` e `$password` sono le variabili inserite dall'utente in input e che saranno passate allo script per la verifica. Tale script oltre alle funzioni basilari prima citate permetterà di eseguire login in una pagina, nel nostro caso `'Facebook.com'`, passandogli come parametro l'username e la password definiti dall'utente.

```
var arg0 = casper.cli.get(0);
var arg1 = casper.cli.get(1);

casper.start('http://www.facebook.com/',function(){
    this.sendKeys("#email",arg0);
    this.sendKeys("#pass", arg1);
});
```

Se la richiesta va a buon fine, vengono subito memorizzati uno screenshot della pagina (`loggato.png`) dell'utente, notificando che siamo entrati effettivamente nella pagina dell'utente desiderato, il nome utente, memorizzato nel file `name.txt`

Le successive richieste quindi avverranno solo se è stata eseguita questa parte di codice, perchè significa che possiamo controllare dati di un'utenza realmente esistente.

Le richieste che possono essere fatte sono per le seguenti informazioni:

- immagine profilo
- panoramica utente
- lavoro e istruzione
- luoghi
- informazioni contatto
- familiari, relazioni
- tue informazioni
- avvenimenti importanti
- amici
- foto

The screenshot shows a web interface with a header bar displaying a welcome message: "Benvenuto **admin**, sono le 16:48:23 del 28/06/2016". Below the header, there is a section titled "Inserisci le credenziali d'accesso:". This section contains two input fields: "Email/Telefono:" with a placeholder "inserisci nome" and "Password:" with a placeholder "inserisci password". Below these fields is a section titled "Seleziona cosa vuoi memorizzare nel file zip:". This section contains a list of checkboxes with the following labels: "Immagine Profilo", "Panoramica Utente", "Lavoro e Istruzione", "Luoghi", "Informazioni Contatto", "Familiari e Relazioni", "Le tue informazioni", "Avvenimenti Importanti", "Amici", and "Foto". Below the list is a green button labeled "Invia". At the bottom of the form, there is a link that says "Torna alla **home**".

Figura 8: Inserimento dati per dump

A seconda delle checkbox selezionate quindi verranno eseguiti gli script corrispondenti, generando ognuno i propri file contenenti i dati dell'utente. I file così memorizzati verranno quindi inclusi in un file .zip denominato

report_nomeutente_gma_hms

(gma = giorno, mese, anno; hms= ore, minuti, secondi)
del time stamp rappresentante l'operazione effettuata.
Creato il file zip con tutti i dati, i file poi vengono cancellati lasciando soltanto il file .zip, del quale andremo poi a creare l'hash.
Alla fine quindi verranno restituiti i passaggi effettuati, notificando i dati estratti, l'hash del file e un link per il download del report effettuato.



Figura 9: Schermata Report effettuato con successo

Il tutto viene poi memorizzato nel database (visualizzabile grazie alla sezione 'visualizza report') memorizzando i seguenti dati:

Nome e Cognome, Data Estazione, Hash File, Link Download

Così che un utente registrato possa scaricare e visualizzare i report, anche in una fase successiva.

Queste sono le operazioni eseguite:				
ID	Nome e Cognome	Data	Hash	Download Link
4	Matteo Riganelli	24/6/2016	ea6238f15665bba667e4459f7938cc5a	Clicca per scaricare
5	Matteo Riganelli	27/6/2016	c0fd90cf9eb50e531cc321505c5fd3c0	Clicca per scaricare
6	Matteo Riganelli	28/6/2016	3ef2bcc486eff4097277b71d1857fbb0	Clicca per scaricare
7	Matteo Riganelli	28/6/2016 - 10:25:40	b9f73f1030a315d59fc63a6b5fb8ce8b	Clicca per scaricare
8	Matteo Riganelli	28/6/2016 - 10:30:2	56612b15fd33454e45c6b330aa98ae00	Clicca per scaricare
9	Matteo Riganelli	28/6/2016 - 16:52:36	50f89472742d8e9c382f2b147c0e83df	Clicca per scaricare

[Torna alla home](#) oppure [esegui il logout](#)

Figura 10: Visualizzazione Report presenti nel DB

Conclusione

Il software, grazie a CasperJS permette di realizzare un modulo lato server per l'estrapolazione dei dati di un utenza Facebook.

Ulteriori Sviluppi riguardano l'aumento delle performance in misura di tempo dell'estrazione dei dati e una compattazione del codice per la parte riguardante la scelta delle informazioni da estrapolare dall'account.

Sarebbe possibile implementare un meccanismo di multi account in modo tale che non tutti gli utenti possano avere accesso a tutte i report effettuati.

Bibliografia

- <http://docs.casperjs.org/en/latest/>
- <https://developers.facebook.com/docs/web>