

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5382-52598

**VYUŽITIE GRAFOVEJ DATABÁZY V PRAXI
BAKALÁRSKA PRÁCA**

2017

Juraj Kubričan

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-5382-52598

VYUŽITIE GRAFOVEJ DATABÁZY V PRAXI
BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná informatika
Číslo študijného odboru: 2511
Názov študijného odboru: 9.2.9 Aplikovaná informatika
Školiace pracovisko: Ústav informatiky a matematiky
Vedúci záverečnej práce: Ing. Maroš Čavojský

Bratislava 2017

Juraj Kubričan



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Michal Ližičiar**
ID študenta: 5982
Študijný program: Aplikovaná informatika
Študijný odbor: 9.2.9 aplikovaná informatika
Vedúci práce: Ing. Matúš Jókay, PhD.

Názov práce: **Anonymizácia internetového prístupu**

Špecifikácia zadania:

Cieľom práce je vytvoriť zásuvný modul pre internetový prehliadač, ktorý bude schopný buď náhodne alebo selektívne meniť informácie používané na identifikáciu používateľa pri jeho prístupe na cieľový server.

Úlohy:

1. Analyzujte dostupnosť a funkčnosť podobných modulov.
2. Analyzujte informácie používané na identifikáciu používateľa pri prístupe na stránku.
3. Navrhnite, implementujte a otestujte anonymizačný modul pre zvolený internetový prehliadač.

Zoznam odbornej literatúry:

1. YARDLEY, G. Better Privacy. [online]. 2012. URL: <http://nc.ddns.us/BetterPrivacy/BetterPrivacy.htm>.
2. ECKERSLEY, P. A Primer on Information Theory and Privacy. [online]. 2010. URL: <https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy>.

Riešenie zadania práce od: 24. 09. 2012

Dátum odovzdania práce: 24. 05. 2013

Michal Ližičiar
študent



prof. RNDr. Otokar Grošek, PhD.
vedúci pracoviska

prof. RNDr. Gabriel Juhás, PhD.
garant študijného programu

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Aplikovaná informatika
Autor:	Juraj Kubričan
Bakalárska práca:	Využitie grafovej databázy v praxi
Vedúci záverečnej práce:	Ing. Maroš Čavojský
Miesto a rok predloženia práce:	Bratislava 2017

Práca sa zaoberá vytvorením webovej aplikácie ktorá bude využívať grafovú databázu. V prvej časti sa nachádza prehľad technológií, ktoré sme použili na implementáciu projektu, ďalej sa tu nachádza priblíženie najpopulárnejších zastupiteľov grafových databáz, ich výhod a nevýhod. V ďalšej časti sa nachádza špecifikácia našej aplikácie cestovného plánovača.

Kľúčové slová: Využitie grafovej databázy v praxi

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Juraj Kubričan
Bachelor Thesis:	Graph database use in a real world application
Supervisor:	Ing. Maroš Čavojský
Place and year of submission:	Bratislava 2017

The bachelor thesis is about creating of a plugin for web browser, that modifies information used to identification of user during accessing a server. There is an overview of methods that increase anonymity during browsing websites, in the first part. The thesis also contains a list of the most used extensions nowadays, that function is a change of some identification components of browser or special ways of anonymization. In the next part of the thesis is an overview of the characteristics of web browser. By combination of these characteristics we can with high level of success identify a user, who have visited the web site. The last part of thesis contains project, implementation and testing of extension created for the web browser Mozilla Firefox. There is also description of source code of extension, the link between the characteristics of web browser, detected limitations and way how to solve them. The resulting extension increases anonymity of user by modification of some characteristic components of web browser or by blocking sending components, that can not be in extension changed. In comparison with most used modules nowadays, this module can modify HTTP headers including characteristics detected by JavaScript commands.

Keywords: Graph database use in a real world application

Obsah

Úvod	1
1 Analýza problému	2
1.1 Relačné	2
1.2 Grafové	2
2 Cieľ práce	3
3 Návrh riešenia	4
3.1 Výber grafovej databázy	4
3.1.1 NEO4J	4
3.1.2 OrientDB	4
3.1.3 ArangoDB	4
4 Implementácia	5
4.1 Použíte technológie	5
4.1.1 NEO4J	5
4.1.2 NeoEloquent ORM	5
4.1.3 PHP	5
4.1.4 Laravel framework	5
4.1.5 Mapbox.js	5
4.1.6 Rome2Rio Api	5
4.1.7 Funkcionalita2	5
4.2 Vzhľad	5
Záver	6
Zoznam použitej literatúry	7
Prílohy	I
A Štruktúra elektronického nosiča	II
B Algoritmus	III

Zoznam obrázkov a tabuliek

Zoznam skratiek a značiek

WWW - World Wide Web

Zoznam algoritmov

B.1 Ukážka algoritmu	III
--------------------------------	-----

Úvod

Pri návrhu aplikácie treba myslieť na štruktúru dát a podľa toho vybrať správnu databázu/DBMS. Správny výber DBMS vie zabezpečiť rádovo nižšie prístupové časy a tým aj väčšiu scalability. Pri aplikáciách kde sú dáta štruktúrované do uzlov a prepojení, je vhodné zvážiť použitie grafovej databázy. My sme navrhli a naimplementovali aplikáciu cestovného plánovača, ktorý potrebuje uchovávať dáta o destináciách a cestách medzi nimi. Preto je pre túto aplikáciu vhodné využiť grafový DBMS

1 Analýza problému

1.1 Relačné

Relačná databáza je databáza, v ktorej sú údaje uložené podľa relačného databázového modelu podľa E. F. Codda z roku 1970. Podľa Relačného modelu sú dáta uložené v tabuľkách. Jeden riadok tabuľky je jeden záznam. Stlpec tabuľky reprezentuje jeden atribút objektu. Väzby (vzťahy) medzi tabuľkami sú riešené pomocou unikátnych identifikátorov tzv. kľúčov. Jediná tabuľka obsahuje kľúč, čo je atribút ktorý unikátne identifikuje každý záznam a druhá tabuľka obsahuje tzv cudzí kľúč, atribút ktorý odkazuje na záznam v prvej tabuľke. Výhodou tohto spôsobu ukladania dát je jednoduchosť a prehľadnosť. Nevýhoda tohto prístupu sa však ukazuje v škálovateľnosti. Pri vyhľadávaní každé toto prepojenie pridáva výpočtovú komplexitu, keďže v každej ďalšej prepojennej tabuľke treba vyhľadať záznam s požadovaným kľúčom ($O(\log(n))$). Všetky používané relačné databázové systémy riešia tento problém škálovateľnosti použitím indexov a rôznymi inými optimalizáciami, no pokiaľ sú naše dáta štrukturované s veľa prepojeniami systém sa spomalí.

1.2 Grafové

Grafové databázy ukladajú dáta vo vrchoch(node) a hranách(edge).

2 Ciel' práce

Cielom práce je navrhnuť a na implementovať aplikáciu, ktorá využíva grafovú databázu, ďalej zhodnotiť výhody tohto riešenia oproti iným typom databáz.

3 Návrh riešenia

3.1 Výber grafovej databázy

3.1.1 NEO4J

3.1.2 OrientDB

3.1.3 ArangoDB

4 Implementácia

4.1 Použité technológie

4.1.1 NEO4J

Neo4J je najpoužívanejšia grafová databáza s

4.1.2 NeoEloquent ORM

4.1.3 PHP

4.1.4 Laravel framework

4.1.5 Mapbox.js

4.1.6 Rome2Rio Api

Rozšírenie tiež okrem splnenia špecifikácie malo pre prehľadnosť a overenie funkčnosti zobrazovať údaje, ktoré boli na server odoslané. Zoznam údajov odoslaných na server, sa mal ukladať do krátkodobej histórie, aby nemal používateľ k dispozícií len najnovšie údaje, ale aj údaje odoslané v nejakom časovom období.

4.1.7 Funkcionalita2

Samozrejmosťou bolo nastavenie zapnutia rozšírenia pri štarte, prípadne interval zmeny odosielaných údajov.

4.2 Vzhľad

Dôležitou požiadavkou kladenou na rozšírenie bolo príjemné používateľské rozhranie. Z tohto dôvodu malo rozšírenie obsahovať zoznam modifikovaných vlastností a tlačidlo pre prístup k nastaveniam rozšírenia v jednoduchnej a praktickej forme. Predpokladaný vzhľad je zobrazený na obrázku č. ??.

[1] Z tohto dôvodu malo rozšírenie obsahovať zoznam modifikovaných vlastností a tlačidlo pre prístup k nastaveniam rozšírenia v jednoduchnej a praktickej forme. Predpokladaný vzhľad je zobrazený na obrázku č. ??.

Záver

Cieľom práce bola analýza anonymizačných modulov, identifikačných prvkov prehliadača a vytvorenie anonymizačného modulu pre internetový prehliadač.

Analýzou najpoužívanějších modulov a vlastností prehliadača, ktoré slúžia na identifikáciu používateľa, sme zistili aktuálny stav a funkcionality rozšírení, ktorými je možné anonymizovať prístup na internet. Väčšina týchto rozšírení modifikuje len časť vlastností prehliadača, ktoré sú odosielané na server, alebo úplne blokuje ich odosielanie. Nami vytvorené rozšírenie dokáže modifikovať väčšinu identifikačných prvkov rozšírenia, pričom dodržiava súvislosti medzi vlastnosťami (používateľský agent odosielaný v hlavičke dopytu je totožný s používateľským agentom zisťovaním pomocou JavaScript príkazu, súvislosť medzi šírkou a dĺžkou rozšírenia obrazovky). Dokáže blokovat údaje, ktoré sú posielané v otvorenej podobe na server a obsahujú informácie o identifikačných údajoch prehliadača, ktoré sa nedajú na úrovni rozšírení modifikovať.

Testovanie rozšírenia nám overilo funkčnosť a správnosť implementácie. Rozšírenie dokáže buď vždy, alebo v časových intervaloch modifikovať väčšinu charakteristických prvkov prehliadača odosielaných na server, a tým zvyšuje anonymitu používateľa.

Zoznam použitej literatúry

- [1] PRATA, Stephen. *Mistrovství v C++*. [prekl.] Vozák David, Beroun Libor, Dokoupil Petr, Ptáček Lubomír Sokol Boris. 3. Praha : Computer Press, 2007. s. 1119. ISBN: 008021680.

Prílohy

A	Štruktúra elektronického nosiča	II
B	Algoritmus	III

A Štruktúra elektronického nosiča

```
\
\Bakalarska_praca.pdf
\FEIk_Identity.xpi
\FEIkIdentity
\FEIkIdentity\chrome.manifest
\FEIkIdentity\install.rdf
\FEIkIdentity\content
\FEIkIdentity\content \function.js
\FEIkIdentity\content \options.xul
\FEIkIdentity\content \overlay.xul
\FEIkIdentity\content \window.js
\FEIkIdentity\content \window.xul
\FEIkIdentity\defaults
\FEIkIdentity\defaults\preferences
\FEIkIdentity\defaults\preferences \prefs.js
\FEIkIdentity\locale
\FEIkIdentity\locale \sk-SK
\FEIkIdentity\locale \sk-SK\options.dtd
\FEIkIdentity\locale \sk-SK\window.dtd
\FEIkIdentity\skin
```

B Algoritmus

Algoritmus B.1 Ukážka algoritmu

```
1  /* Hello World program */
2
3  #include <stdio.h>
4
5  struct cpu_info {
6      long unsigned utime, ntime, stime, itime;
7      long unsigned iowtime, irqtime, sirqtime;
8  };
9
10 main()
11 {
12     printf("Hello World");
13 }
```
