FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Návrh systému pro monitoring marketů na darknetu
Student:	Bc. Josef Smrž
Vedoucí:	Ing. Jan Bouchner
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

1) Seznamte se s pojmem darknet, jeho prostředím a technologiemi, na základě kterých funguje.
2) Analyzujte obsah a strukturu trhů na darknetu a proveďte rešerši. Popište možnosti získávání tohoto obsahu.
3) Navrhněte systém, který bude automaticky stahovat, ukládat a monitorovat informace z vybraného trhu na darknetu. Systém umožní následné prohledávání strukturovaných a nestrukturovaných dat (fulltextové vyhledávání) a notifikace předem definovaných změn. Návrh by měl být obecný s ohledem na možnost přidání dalšího trhu.
4) Sestavte kalkulaci nákladů na vytvoření a provoz takového systému.
5) Zhodnoťte nefinanční benefity navrženého řešení a stanovte metriky pro měření úspěšnosti projektu.
6) Vytvořte prototyp systému. Rozsah prototypu bude dohodnut s vedoucím práce.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 10. ledna 2018

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Návrh systému pro monitoring marketů na darknetu

Bc. Josef Smrž

Katedra softwarového inženýrství
Vedoucí práce: Ing. Jan Bouchner

9. května 2018

Poděkování

Děkuji vedoucímu práce za věnovaný čas a podnětné rady a své rodině za podporu a zázemí poskutnuté při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Na Kladně dne 9. května 2018 .....

Odkaz na tuto práci

Smrž, Josef. *Návrh systému pro monitoring marketů na darknetu.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tématem této diplomové práce je návrh systému schopného sbírat informace o zboží, jenž je nabízeno v neveřejné části internetu, konkrétně na darknetových marketech. Systém je navržen pro možné vyhledávání v získaných datech a upozorňování na změny zboží prostřednictvím notifikačních zpráv.

Součástí práce je analýza projektu, pomocí které bylo možné navrhnout a rozplánovat vývoj výše zmíněného systému pro sekci kybernetické kriminality Policie ČR. Pro demonstraci návrhu systému byl implementován jeho prototyp.

Práce se též věnuje pojmům spojeným s darknetem, protokolu sítě Tor a způsobům automatizovaného sběru dat.

Klíčová slova darknet, darknetové markety, sběr dat, monitorovací systém

Abstract

This thesis presents the design of a system for harvesting information about goods sold on darknet markets. It enables users to search the gathered data and notifies them about changes that occur on the markets that are being tracked.

The thesis also includes a project analysis of the possible implementation of this system for the Cyber Crime Unit of the Police of the Czech Republic. For demonstration purposes, a prototype of the system was implemented.

Furthermore, this work explains the darknet, the Tor protocol and the means of automated data collection.

Keywords darknet, darknet markets, data collection, monitoring system

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Internet se postupem času stal přirozenou součástí našeho života. Je to pro nás zdroj informací i zábavy, využíváme ho ke komunikaci i k rutinním záležitostem, jako je nákup zboží a v dohledné době snad i styk s úřady.

To je internet, jak ho zná každý z nás. Ne každý uživatel internetu ale ví, že existuje i jeho „temná“ anonymní verze. Ta je známá pod označením **darknet** a je uživatelům přístupná pouze pomocí speciálního softwaru. Pro většinu zasvěcených symbolizuje darknet místo, kde mimo jiné prosperuje obchod s drogami, zbraněmi, padělky, uniklými daty a dalším **ilegálním** zbožím.

Obchodování tohoto zboží probíhá na online tržištích, neboli **darknetových marketech**, které fungují na principu obchodních platforem pro obchodníky po vzoru internetových obchodů jako jsou *Ebay*, *Amazon* či *Alibaba*. Platí se na nich výhradně pomocí **kryptoměn**, které proto díky své *relativní* anonymitě představují ideální prostředek a dlouhou dobu tak byly s darknetovými markety úzce spojovány.

Od uzavření největšího marketu své doby **Silk Road** zásahem FBI vzniklo mnoho jeho klonů a popularita darknetových marketů díky medializaci prudce stoupla. To mělo podobný efekt, jaký pro komerci způsobil rozmach internetu: velká část prodeje ilegálního zboží, v čele s drogami, se z „ulice“ přesunula online. Pro koncové spotřebitele se tím toto zboží stalo dostupnějším a jeho nákup bezpečnějším. Díky konkurenčnímu prostředí se snížila i jeho cena. Protože markety umožňují zanechat o prodejcích recenze, mohou zákazníci navíc nakupovat jen u těch renomovaných.

Se svou rostoucí popularitou se darknetové markety dostaly do hledáčku policie a bezpečnostních složek po celém světě. Zvlášť aktivní jsou v tomto ohledu americká FBI a nizozemská policie, které již mnoho prodejců dopadly a několik marketů zavřely.

I **Policie České republiky** téma darknetových marketů řeší a poptává proto systém, který je bude monitorovat. Tato diplomová práce se zabývá návrhem takového systému spolu s realizačními aspekty jeho vývoje.

Cíle práce

Hlavním cílem práce je **návrh systému** pro monitorování obsahu marketů na darknetu poptávaný Policií ČR. Hlavním požadavkem na navrhovaný systém je schopnost prohledávat obsah marketu a upozorňovat uživatele na nové změny, jako je například vystavení nového zboží. Na základě návrhu bude vytvořen prototyp simulující základní koncept fungování systému.

Dalšími cíly práce jsou **analýza realizace projektu** z hlediska jeho finanční náročnosti, zhodnocení přínosů a rizik projektu a stanovení metrik pro měření jeho úspěšnosti.

Dílčím cílem je dále **popis prostředí darknetu**, jeho technologií a obsahu, se kterým je možné se na jeho webech setkat.

Struktura práce

Práce je rozdělena na 3 kapitoly, které zároveň představují fáze práce na tématu:

- **Kapitola 1** se věnuje vymezení pojmů souvisejících s darknetem a seznámení se s jeho prostředím a druhem uživatelů. V kapitole je dále popsána dominantní síť darknetu *Tor*, technologie, na které Tor funguje, a nástroje sloužící k připojení se k Toru.
- **Kapitola 2** obsahuje návrh systému a analýzu projektu jeho implementace a nasazení. Analytická část se zabývá požadavky, přínosy, riziky a finanční stránkou projektu, stejně tak jako existujícími řešeními pro sběr dat, zejména v prostředí darknetu. Dále je navržena architektura systému jsou popsány funkce jeho jednotlivých komponent.
- **Kapitola 3** popisuje průběh implementace prototypu navrženého systému.

Kapitola 1

Rešerše

Tato kapitola se zabývá teoretickým pozadím práce. V její první části je objasněn pojem darknet, stejně jako další pojmy, které s ním souvisí. Důraz je kladen na rozlišení pojmů, které bývají často zaměňovány. Druhá část se pak věnuje technickým aspektům sítě *Tor*, jakožto dominantní sítě na darknetu.

1.1 Darknet a související pojmy

V obecném povědomí je informace o tom, že existuje „nebezpečná“ část internetu sloužící k ilegální činnosti. Tento internetový prostor bývá prezentován jako místo plné drog, zbraní, dětské pornografie a dalšího ilegálního obsahu; a média ho často nekonzistentně označují jako *darknet*, *deep web* či *dark web* [9]. Tyto pojmy však nejsou ekvivalentní a zdá se, že tyto pojmy nejsou zcela správné užívány ani na webových stránkách, které se zabývají primárně internetem a jeho technologiemi. Pro podložení tohoto tvrzení je níže vybráno několik příkladů:

- Web **WhoIsHostingThis** ve své *infografice*[1] označuje deep web jako část internetu, která je 500x větší než *surface web*. Dále však tvrdí, že internet je tvořen ze 4 % surface webem a z 96 % deep webem, čímž si odporuje předchozímu tvrzení. V *infografice* se mluví i o tom, že pro přístup na deep web je zapotřebí speciálního prohlížeče, například *Tor Browseru*[2] [10]. Tato infografika byla přebrána i dalšími renomovanými weby, jako například Business Insider [11] nebo Boy Genius Report [12].

[1] Infografika je grafika, která poutavě prezentuje zajímavá data, často formou grafů a dalších vizualizací.

[2] Tor Browser je webový prohlížeč speciálně navržený pro přístup k internetu prostřednictvím sítě *Tor*. Více o Tor Browseru v sekci 1.2.4.2.

- **ICANN**[3] jako jedna z hlavních internetových autorit na svém webu popisuje deep web jako souhrn webových stránek, které nejsou indexovány webovými vyhledávači. Zmiňuje i to, že některé takové stránky nabízejí širokou nabídku ilegálních služeb a zboží [14].

- Na stránkách projektu **Informační gramotnost** se lze dočíst, že *povrchový web* je část webového prostoru, která se od *hlubokého webu* liší v indexaci webovými vyhledávači. Při vyhledávání tedy obsah hlubokého webu nebude mezi výsledky, jelikož se na tyto stránky nedostanou *web crawleři*[4], kteří by se o indexaci postarali. Zmiňuje i *temný web*, k němuž se lze připojit pouze pomocí speciálních nástrojů, je anonymní a bývá využíván k ilegální činnosti [15].

Na vybraných příkladech výše je vidět, že zmíněné pojmy bývají používány velmi volně a v tom, co je ještě deep web a co už je dark web není zcela jasný konsenzus ani v odborných kruzích. V této práci je terminologie sjednocena tak, jak pojmy interpretuje většina kompetentních zdrojů, a takřka se shoduje s definicí pojmů podle doktorky Monicy J. Barrattové. Ta terminologii ojedinělým způsobem shrnula v příspěvku na svých webových stránkách[5].

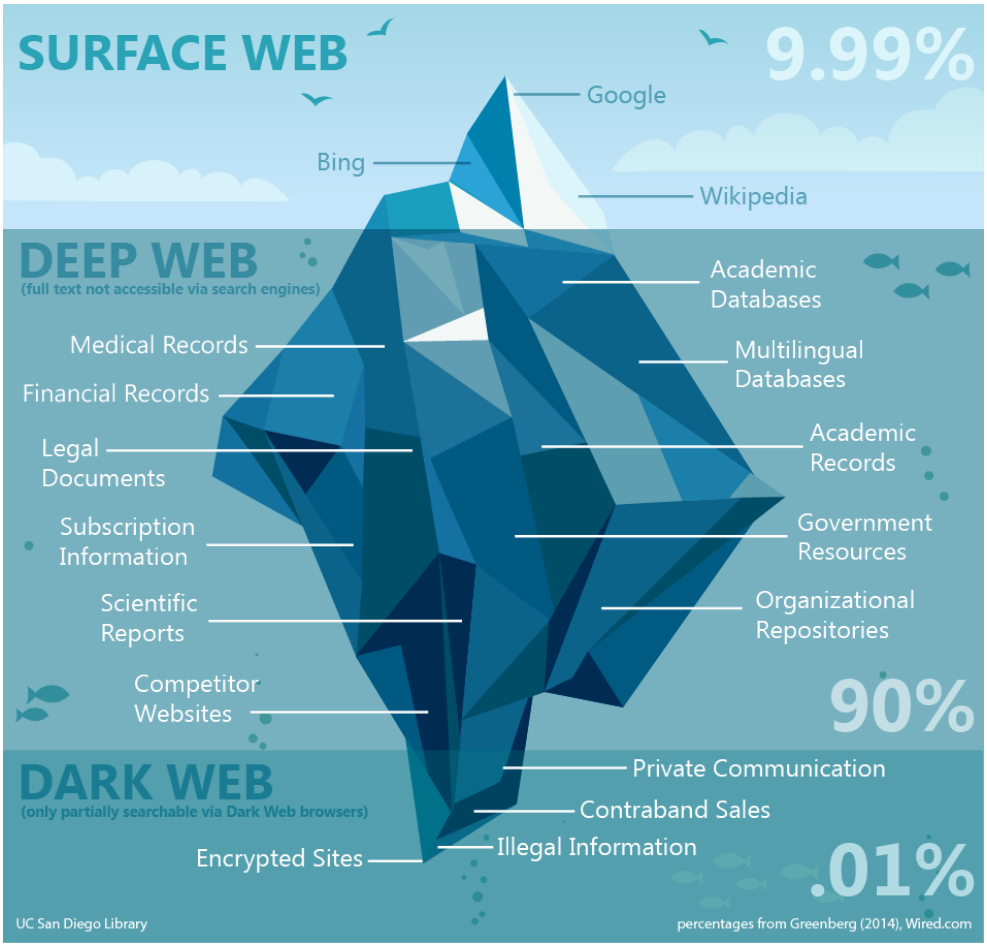
Klasifikace *World Wide Webu*[6] na surface web, deep web a dark web bývá též znázorňována jako plující ledová hora, jejíž část nad hladinou představuje surface web a většinová ponořená část pak znázorňuje deep web. Na obrázku 1.1 je vidět jedna ze zdařilejších grafik tohoto typu, která je používána knihovnou University of California San Diego [16].

[3] ICANN je americká nezisková organizace, jejíž hlavním úkolem je celosvětová koordinace a správa DNS a přidělování rozsahů IP adres regionálním registrátorům a dalším organizacím [13].

[4] Více o web crawlerech viz. 1.1.1.

[5] Příspěvek „A discussion about dark net terminology“ je dostupný na adrese `http://monicabarratt.net/a-discussion-about-dark-net-terminology`.

[6] World Wide Web, zkráceně WWW nebo též web, je systém dokumentů a dalších zdrojů, které jsou mezi sebou propojeny hypertextovými odkazy, a který je dostupný prostřednictvím sítě internet.



Obrázek 1.1: Graficky znázorněná klasifikace WWW používaná knihovnou University of California San Diego

1.1.1 Surface web

Pro ujasnění terminologie je vhodné začít „na povrchu“. **Surface web**, česky povrchový web, je označení pro souhrn obsahu na webovém prostoru, který je možno vyhledat pomocí tradičních internetových vyhledávačů.

Aby webové vyhledávače mohly na základě zadaných klíčových slov vrátit relevantní výsledky, musí mít nejdříve všechny webové stránky zaindexovány. K zařazení stránek do indexů se používají *web crawleři*. web crawler (např. **Googlebot** nebo **SeznamBot**) je program, který automatizovaně prochází všechny dostupné webové stránky, sbírá o nich informace, a na základě jejich kvality rozhodne o jejich indexaci. Mezi stránkami se crawler pohybuje pomocí odkazů, je tedy důležité, aby na stránky, které mají být zaindexovány, vedlo

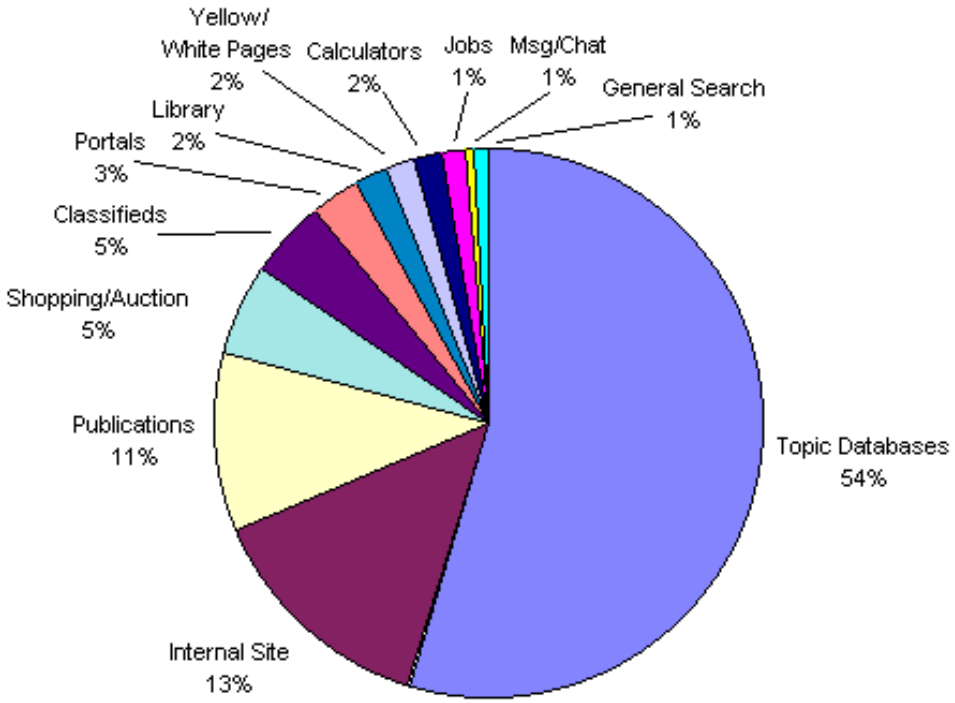
dostatek odkazů, pokud možno z kvalitních webových stránek. Všechny tyto indexované weby tedy tvoří surface web.

Je možné se setkat ještě s dalšími, ekvivalentně užívanými pojmy, jako jsou *visible web* nebo *clearnet*. Označení *clearnet* je pak často používáno především v komunitě uživatelů darknetu.

1.1.2 Deep web

Právě kolem **deep webu** (česky *hluboký web*) je asi nejvíce nejasností, jelikož bývá až příliš často špatně používán a mylně zaměňován s dark webem.

Obecně se dá říci, že deep web je přesný opak *surface webu*, tedy jakýsi jeho doplněk. Jde o webový obsah, který z nějakého důvodu není webovými vyhledávači indexován, a nemůže proto být uživatelům vrácen jako výsledek hledání. Tohoto obsahu je velké množství a často se uvádí, že velikost deep webu je oproti *surface webu* o několik řádů větší. Podle studie z roku 2000 byla velikost veřejně dostupného, nijak nechráněného obsahu deep webu až 550x větší. Celková velikost obsahu na deep webu pak byla odhadována jako až 2000x větší oproti *surface webu* [1]. Zastoupení různých druhů obsahu deep webu podle studie je znázorněno na obrázku 1.2.



Obrázek 1.2: Zastoupení obsahu na deep webu podle jeho druhu [1]

Poměr indexované a neindexované části webového prostoru se dnes ale zřejmě výrazně liší a důvodů je hned několik. Jedním z nich je to, že v době zpracování studie byl **Google** v provozu teprve dva roky a během svého působení se v pokrytí obsahu, který je schopen vyhledat, nepochybně zlepšil. Dalším důvodem k tomuto tvrzení je vznik nových služeb, které uchovávají obrovská množství dat. To se týká jak těch vyhledatelných, jako například bezplatně dostupný videoobsah, tak i dat, která jsou vyhledávačům skryta, jako například data sociálních sítí, či placené *streamovací*[7] služby. Velký podíl v neindexovaném obsahu mají také data *cloudových*[8] úložišť.

Hlavními druhy obsahu klasifikovaného jako deep web jsou:

- cloudové služby,
- intranety,
- sociální sítě,
- placený multimediální obsah,
- databáze prohledávatelné pouze z webu jejich přidružené služby, jako:
  - veřejně dostupné databáze státní správy,
  - rezervační systémy,
  - knihovní systémy,
  - vyhledávače letenek a ubytování,
- weby záměrně limitující přístup robotů (většinou pomocí *Turingových testů*[9] jako je *CAPTCHA*),
- dynamicky generovaný obsah,
- webové stránky nebo jejich části, na které nevedou žádné odkazy z indexovaných stránek.

1.1.3 Dark web

Velmi malá část deep webu se označuje jako **dark web**, což je obsah nacházející se na tzv. **darknetech**, sítích, které se snaží o zachování anonymity a zamezení sledování jejich uživatelů [17]. K tomu *darknety* používají speciální šifrované protokoly a připojit se k nim je tak možné pouze pomocí speciálních

[7] Streamováním se označuje přehrání audiovizuálního obsahu bez nutnosti nejprve stáhnout celé dílo. Přehrávání se spustí ihned po stažení začátku a další stahování probíhá na pozadí během přehrávání.

[8] Cloudové služby jsou online služby poskytující výpočetní výkon a umožňující uchování a zpracování dat a na vzdálených, často decentralizovaných serverech.

[9] Pojmenovaný podle Alana Turinga, Turingův test je pokus ověřit, zda jeho vstupy zadává člověk nebo stroj.

programů, primárně *darkwebových* prohlížečů. Mezi významné *darknety* patří například **I2P**, **Freenet** nebo **Tor**, který se díky své velikosti (aktuálně kolem 2,5 mil. uživatelů [18]) stal k pojmu darknet takřka synonymem. Z toho důvodu se bude tato práce dále zabývat právě Torem.

Motivace k používání „anonymní“ verze internetu může být různá. Podle jeho autorů Toru využívají:

- **žurnalisté**, kteří se obávají perzekuce nebo odplaty;
- **policejní složky a bezpečnostní služby** pro získávání anonymních tipů nebo pro utajení během vyšetřování;
- **redakce zpravodajství** pro získávání anonymních svědectví;
- **občané zemí s represivními režimy** pro získání necenzurovaných informací;
- **aktivisté a blogeři** k bezpečnému šíření svých názorů;
- **armáda** k utajení komunikace během operací;
- **běžní lidé** chránící si své soukromí [19].

Vzhledem k tomu, že je aktivita na dark webu špatně monitorovatelná a identifikace jeho uživatelů je obtížná, je na něm právo vymahatelné jen velmi těžko. Díky tomu jsou *darknety* ideálním útočištěm i pro zločince.

Stránky a služby na dark webu se pro jejich skrytí před běžnými uživateli internetu nazývají **hidden services**, česky *skryté služby*. Ty jsou v případě Toru přístupné na speciálních adresách končících *.onion*. Mezi hlavní druhy obsahu, se kterým je možné se na dark webu setkat, patří:

- markety nabízející převážně ilegální zboží,
- svobodná diskuzní fóra a blogy,
- weby pro *whistleblowery*[10],
- ilegální pornografie,
- anonymní chatovací služby,
- weby pro ilegální šíření autorského obsahu.

[10]Whistleblower je člověk informující veřejnost o vědomém nelegálním či neetickém chování organizace, pro kterou pracuje nebo pracoval.

1.1.3.1 Darknetové markety

Oblastí dark webu, se kterou je veřejnost nejblíže seznámena, jsou díky jejich medializaci právě **darknetové markety** s ilegálním zbožím. Jde o online tržiště, která by šla v prostředí *clearnetu* přirovnat ke službám jako jsou *eBay* nebo *Amazon*. Jedná se tedy o weby, které obchodníci využívají jako *platformy* pro prodej svého zboží.

Dalším typem *darknetových* webů určených k prodeji zboží je obdoba klasického e-shopu. Těm se často přezdívá **vendor shopy**, protože se jedná o web jednoho prodejce (anglicky *vendora*).

Využívání *marketů* se zdá jako populárnější. Přestože si ukrajují z každého obchodu provizi, přináší pro obchodníky i zákazníky mnoho výhod. Některé z nich jsou:

- Obchodníkům odpadá starost s vývojem, provozem a inzerováním vlastních webových stránek.
- Z důvodu vysokého počtu podvodných webů mají zákazníci k zavedeným marketům vyšší důvěru.
- Markety nabízejí možnost platby za zboží pomocí *escrow*[11].
- Po realizovaném obchodě může zákazník zanechat obchodníkovi veřejnou *recenzi s hodnocením*, což motivuje obchodníka a zvyšuje důvěru dalších zákazníků.
- Markety mívají širokou nabídku zboží.

Na darknetu se za zboží a služby platí výhradně v **kryptoměnách**, z nichž nejpoužívanějšími jsou *Bitcoin*, *Litecoin* a *Monero*. Hlavním důvodem používání *kryptoměn* je anonymita, které je při jejich používání možné dosáhnout. Ta je důležitá vzhledem ke zmíněnému ilegálnímu zboží, se kterým je zde obchodováno. Jedná se například o:

- drogy
- zbraně
- kradené zboží
- falešné dokumenty, doklady a peníze,
- pornografie,
- návody k ilegální činnosti,

[11] Platba přes escrow účet, česky vázaný účet, se dá přirovnat k obchodu s uložením peněz u notáře. Při objednání zákazník zaplatí na escrow účet ovládaný marketem a až po obdržení zboží a zpokojenosti zákazníka jsou peníze převedeny obchodníkovi.

- služby, jako jsou:
  - nájemné vraždy,
  - hackerské útoky,
- kradené citlivé informace, např:
  - osobní údaje,
  - e-mailové adresy,
  - uniklá data.

1.2 Tor

Tor je otevřený software chránící anonymitu svých uživatelů na internetu. Funguje na základě tzv. **onion routingu** (česky cibulové směrování), což je technika, při které se přenášená data několikanásobně šifrují a před jejich doručením se přeposílají přes několik dalších uzlů v síti. Ačkoliv vzniklo implementací *onion routingu* více, tou původní a nejpoužívanější z nich zůstává Tor, na což jeho autoři (*The Tor Project*) poukazují přímo v jeho názvu. Ten je odvozený ze zkratky pro **The Onion Routing** – Tor, kterou často bývá nesprávně označován [20].

Tor je jeho uživateli nejčastěji používán k prohlížení webových stránek, a to jak *surface webu*, tak i *skrytých služeb Toru*, které se nazývají **onion services**. *Tor Project* pro tento účel vyvinul speciální webový prohlížeč **Tor Browser**, který je předkonfigurován pro připojení k Toru a oproti standardním prohlížečům je nastaven pro maximální anonymitu. Ačkoliv je Tor typicky používán pro procházení webu, je možné ho používat jako *proxy*[12] i pro jinou síťovou komunikaci, např. e-mail, chat nebo sdílení souborů.

1.2.1 Vznik Toru

Vývoj *onion routingu* začal v roce 1995 jako projekt amerického námořnictva, *Office of Naval Research (ONR)*, a později byl financován i americkou agenturou *Defense Advanced Research Projects Agency (DARPA)*. Během svého vývoje prošel *onion routing* třemi hlavními stádii, označovanými jako generace 0, 1 a 2. Zatímco generace 0 a 1 byla obdobími návrhu a raného expetimentování s protokolem, generací 2 se označuje již období od začátku vývoje Toru v roce 2002 [21].

Financování bezpečnostními agenturami skončilo v roce 2006, kdy byl jako nezisková organizace založen *The Tor Project* [22].

[12] Proxy server funguje jako prostředník, který přeposílá datový tok, který obdrží, k jeho adresátovi.

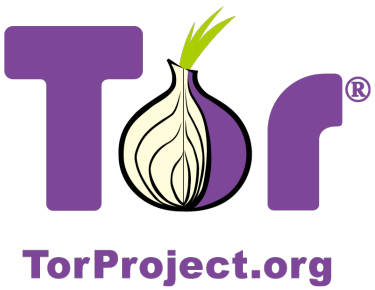
1.2.2 Onion routing

Tato část se věnuje *onion routingu* jako hlavní technologii, na základě které Tor funguje.

1.2.2.1 Motivace

Při standardním přenosu dat po síti si mezi sebou počítače posílají *pakety*[13], které spolu s přenášenými daty obsahují i *metadata*[14], jako jsou údaje o odesílateli a adresátovi. V případě, že je datový přenos odposloucháván a analyzován, je pomocí těchto údajů možné zjistit, se kterými servery uživatel komunikuje, sledovat jeho chování a tuto komunikaci pak například i blokovat. Blokování přístupu na vybrané webové stránky je například běžnou praxí v zemích uplatňujících cenzuru.

Onion routing je způsobem, jak se tomuto sledování bránit, a jak si na internetu zachovat svobodu a anonymitu. Jde o protokol fungující na *aplikační vrstvě*[15] architektury *TCP/IP*[16] bývá nejvíce používán při prohlížení webu, lze ho však použít pro komunikaci jakékoliv služby používající *TCP pakety*. Při *onion routingu* neprobíhá komunikace se serverem napřímo, ale prostřednictvím několika uzlů na síti za použití několika vrstev kryptografie. Na základě přirovnání tohoto vrstvení ke slupkám cibule získal *onion routing* svůj název a *The Tor Project* tento motiv používá i ve svém logu (obrázek 1.3).



Obrázek 1.3: Motiv cibule v logu The Tor Project [2]

[13]Paket je blok dat přenášený po síti, který má danou maximální velikost a kromě uživatelských dat obsahuje i informace sloužící k jeho doručení. V běžném životě ho je možné přirovnat k dopisu.

[14]Metadata jsou tzv. „data o datech“. Jedná se o informace, které se k daným datům vztahují. V případě paketů se jedná o směrovací informace sloužící ke korektnímu doručení příjemci.

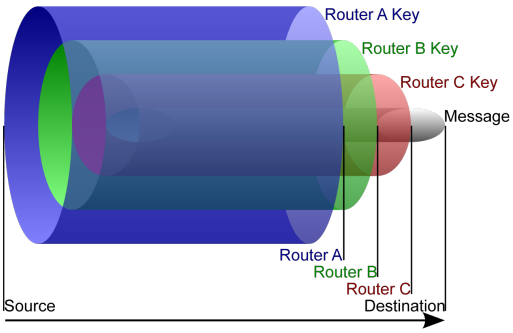
[15]Aplikační vrstva je tvořena množinou protokolů a služeb poskytující aplikacím přístup ke komunikaci po síti.

[16]TCP/IP je rodina protokolů, které jsou základními kameny fungování síťového přenosu. Čtyřmi vrstvami modelu TCP/IP jsou aplikační, transportní, síťová a fyzická.

1.2.2.2 Princip fungování onion routingu

Níže je popsán základní princip fungování *onion routingu* při přístupu na standardní webový server na *clearnetu*:

1. Nejdříve si klient na síti vybere několik uzlů, které jsou v případě Toru **tři**. S těmito třemi uzly si vymění šifrovací klíče, které budou používány k šifrování a dešifrování zpráv. Úloha všech tří uzlů je stejná, a sice dešifrování a přeposílání dat dál. Klient odesílaná data vytvořenými klíči zašifruje, neboli je „obalí“ do několika vrstev viz. obrázek 1.4.



Obrázek 1.4: Zpráva postupně zašifrovaná třemi různými klíči, kdy každý z uzlů zná pouze jeden z nich [3]

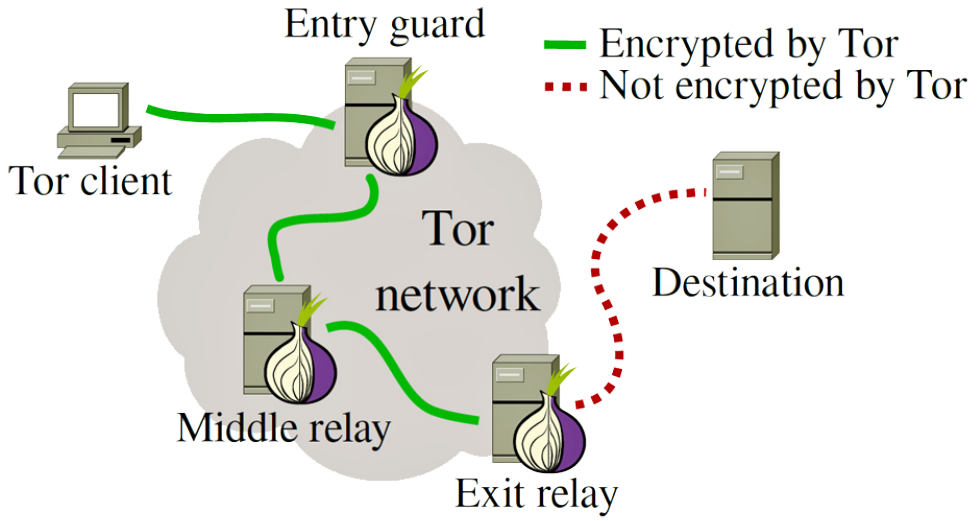
2. Trojnásobně zašifrovanou zprávu klient odešle prvnímu z uzlů, označovanému jako **entry guard**. Ten pomocí svého klíče zprávu částečně rozšifruje a zjistí tím i adresu uzlu dalšího v pořadí, kterému stále ještě dvakrát zašifrovanou zprávu přepošle.

3. Následující uzel se nazývá **middle relay**, jelikož jde o prostředního z uzlů. Ten s obdrženou zprávou udělá to samé, čímž opět získá adresu dalšího uzlu, kterému přepošle nyní již jen jedním klíčem zašifrovanou zprávu.

4. Posledním z Tor uzlů je **exit relay**, který ze zprávy po jejím obdržení „sloupne“ poslední vrstvu kryptografie. Rozšifrováním takto získal standardní paket, který nyní odešle na server.

5. Předání odpovědi od serveru klientovi probíhá na stejném principu, pouze v opačném chodu. Poté, co *exit relay* od serveru obdrží odpověď, zašifruje ji svým klíčem a vrátí *middle relayi*. Ten provede totéž a předá odpověď *entry guardu*, který ji zašifruje potřetí a konečně vrátí klientovi.

Takové propojení uzlů se nazývá **Tor circuit** – „obvod“, kterým „protéká“ komunikace od klienta až k serveru (obrázek 1.5).



Obrázek 1.5: Uzly Tor sítě plnící úlohy jednotlivých článků Tor circuitu [4]

Adresy uzlů Tor sítě jsou pro její funkčnost veřejně dostupné. Toho mohou využít poskytovatelé internetu, kteří si přejí internet cenzurovat, a komunikaci s těmito adresami blokovat. Z toho důvodu existuje i speciální druh uzlů, a to **bridge relaye**, jejichž adresy nejsou zveřejňovány a jejich blokování je tedy těžší.

I přes použití bridge je ale možné Tor detekovat analýzou datového toku, což se některým cenzorům daří. Z toho důvodu bývá navíc použita *obfuskace*[17] datového toku, která v případě Toru nese název **pluggable transports** [23].

1.2.2.3 Vlastnosti protokolu

Z popsaného principu fungování *onion routingu* vyplývá při jeho použití několik vlastností:

- **Entry guard** zná adresu klienta, ale nezná obsah zprávy, ani adresu serveru, ke kterému se klient připojuje.
- **Middle relay** nezná adresu klienta, obsah zprávy, ani adresu serveru.
- **Exit relay** zná adresu serveru a pokud není použito šifrované spojení, pak i obsah zprávy. Nezná však adresu klienta, ví pouze, že někdo ze Tor sítě se serverem komunikuje.

[17] Obfuskace je technika, která má za cíl ztížit čitelnost. Typickým příkladem je obfuskace zdrojového kódu prováděná za účelem chránění know-how.

- **Entry guard** ani **middle relay** nevědí, kolikátým uzlem v pořadí jsou, tedy kolika klíči je ještě zpráva zašifrována. Komunikace mezi klientem a *entry guardem* se totiž od té mezi *entry guardem* a *middle relayem* v ničem neliší. *Entry guard* tak neví, zda stroj, od kterého požadavek přišel, je přímo uživatel Toru, či pouze jeden z uzlů.

- Jelikož je seznam uzlů sítě Tor veřejný a protokol komunikuje na speciálním portu, poskytovatel internetového připojení ví, že **klient** komunikuje po síti Tor. Neví však s kým ani co je obsahem komunikace.

- Podobně tomu je v případě **serveru**. On i jeho poskytovatel internetového připojení díky veřejnému seznamu uzlů může zjistit, že s ním komunikuje někdo z Tor sítě, ale neví kdo. Pokud přenos není šifrován ještě pomocí *SSL*, zná i obsah zpráv.

- Z pohledu serveru se zdá, že s ním komunikuje **exit relay**, ale skutečný klient bývá zpravidla tisíce kilometrů daleko.

Dalším důsledkem návrhu fungování Toru je delší **doba odezvy a nižší rychlost přenosu** oproti ekvivalentní přímé komunikaci. Důvodů je hned několik:

- Namísto jednoho požadavku na server a jedné odpovědi klientovi se kvůli třem prostředníkům zvýší jejich počet na **čtyřnásobek**.

- Protože jsou často vybrané uzly rozesety po zeměkouli, každý datový přenos trvá dlouhou dobu.

- Síť funguje na principu solidarity zprostředkujících uzlů, bez nároku na odměnu. Kvalita jejich připojení tedy nemusí být dostatečná.

- Každý uzel může obstarávat až tisíce spojení, díky tomu nemusí zvládat obsloužit požadavky dostatečně rychle.

- Šifrováním a dešifrováním se zvyšuje režie.

1.2.3 Onion services

Jak již bylo zmíněno, skryté služby fungující na síti Tor se nazývají **onion services**. Zpravidla se jedná o webové stránky, které jsou umístěné na serverech, u kterých je z různých důvodů třeba, aby jejich poloha byla nevystopovatelná a aby komunikace s nimi byla neblokovatelná. Z toho důvodu komunikují s okolím pouze prostřednictvím Toru. Vytvoření takové služby je poměrně snadné a přímočaré a stačí k němu pouze instalace Toru a standardní webový server. Jednoduchý návod k jejímu zprovoznění poskytuje *Tor Project* na adrese `https://www.torproject.org/docs/tor-onion-service.html.en`.

1.2.3.1 Onion adresy

Pro připojení ke skryté službě musí uživatel znát její adresu. Ta má typicky 16 znaků (56 v případě 3. verze protokolu[18]), za kterými následuje koncovka **.onion**[19], na základě čehož se pro ni vžil také název **onion adresa**.

Onion adresy skrytých služeb jsou odvozeny z jejich vygenerovaných veřejných klíčů. Z toho důvodu typicky vypadají jako nic neříkající změť znaků. Příkladem je adresa *suw74isz7wqzpmgu.onion*, která *whistleblowerům* slouží k nahlašování na *WikiLeaks* [26], nebo *lchudifyeqm4ldjj.onion*, což je adresa tržiště *Dream Market*.

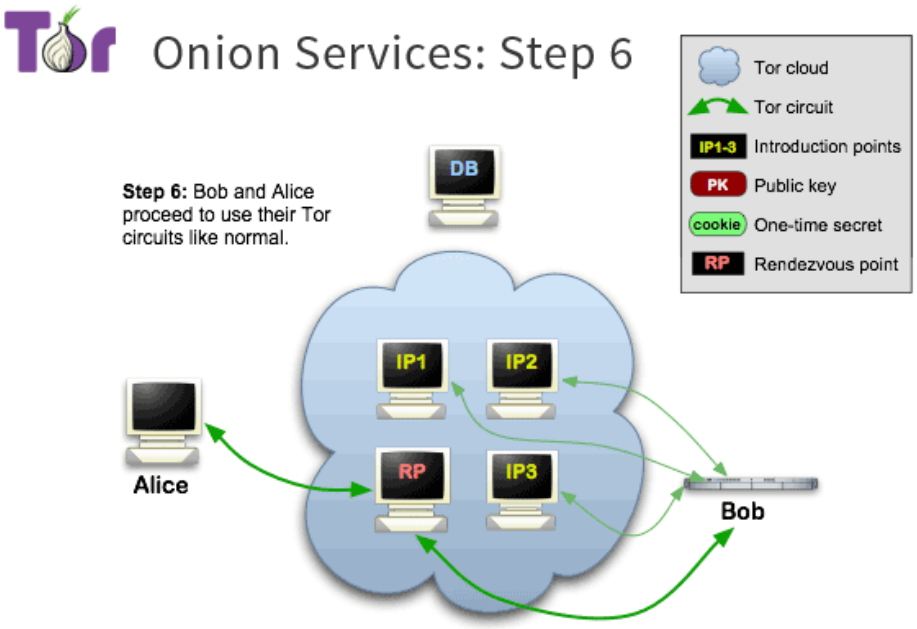
Vygenerováním velkého množství adres je možné nalézt takovou, která bude z části dávat smysl a bude tak snáze zapamatovatelná. Takovou adresu používá například **Facebook** (*facebookcorewwwi.onion*, zřízeno r. 2014 [27]) nebo **New York Times** (*nytimes3xbfgragh.onion*, zřízeno r. 2017 [28]).

1.2.3.2 Připojení ke skryté službě

Připojení ke skrytým službám je o poznání složitější než připojení ke *clearnetovému* serveru. Z důvodu jeho utajení není klientovi jeho IP adresa známa. Navázání spojení je řešeno pomocí *distribuované databáze*, tzv. **introduction pointů** a **randevous pointu**, ke kterému je klient i server připojen každý svým vlastním *Tor circuitem* a který pak zprostředkovává jejich komunikaci [5]. Mezi klientem a serverem je tak díky použití dvou *Tor circuitů* a jednoho *randevous pointu* sedm uzlů sítě Tor. Oba dva spolu díky tomu mohou komunikovat, aniž by jeden z nich znal identitu toho druhého. Detailní popis navázání spojení je zveřejněn na webu *Tor Projectu*, ze kterého je pro ilustraci převzat i obrázek 1.6.

[18] V září 2017 byla vydána nová verze protokolu, která používá adresy o 56 znacích a přináší vylepšení v oblasti bezpečnosti [24].

[19] Koncovka .onion slouží k označení, že se jedná o adresu sítě Tor. V roce 2015 byla oficiálně uznána jako doménové jméno pro speciální použití [25].



Obrázek 1.6: Navázané spojení mezi klientem (Alice) a serverem hostujícím skrytou službu sítě Tor (Bob) [5]

1.2.4 Přístup na síť Tor

Existuje několik způsobů, jak lze síť Tor používat, s čímž souvisejí i způsoby, jak se k ní připojit. Většina uživatelů se k Toru patrně připojuje za účelem prohlížení webových stránek, jak už *surface webu*, tak i dark webu, k čemuž je nejlepší nástroj bezpochyby webový prohlížeč. Jeho použití ale není vhodné pro automatické zpracování, kdy je zapotřebí programového přístupu. V této sekci jsou proto hlavní způsoby připojení k síti Tor shrnuty.

1.2.4.1 SOCKS proxy

Základním způsobem, jak se k síti Tor připojit a používat ji, je prostřednictvím **SOCKS proxy**. *Socket Secure* neboli *SOCKS proxy* se od *HTTP proxy* liší tím, že podporuje i jiný síťový provoz než jen HTTP a je ji tak možné použít i pro jiný síťový provoz, než jen pro procházení webu [29].

Tor při svém spuštění takovou *proxy* automaticky vytvoří. Pro přesměrování síťového provozu přes Tor pak uživateli stačí v aplikaci podporující *SOCKS* použití *proxy* nastavit.

Takto lze nastavit například i webový prohlížeč, který pak bude komunikovat přes Tor. Pro opravdu anonymní prohlížení internetu toto řešení ale

není příliš vhodné, jelikož běžné prohlížeče často umožňují odhalení identity uživatele i při použití Toru. Proto je jeho autory doporučováno použít *Tor Browser*, který je k prohlížení webu přes Tor přímo vyvinut.

1.2.4.2 Tor Browser

Nejběžnějším způsobem připojení k síti Tor je již zmíněný **Tor Browser**, což je webový prohlížeč vyvíjený *The Tor Projectem* speciálně k tomuto účelu. Jeho použití je velmi snadné, protože standardně není třeba nijak konfigurovat. Není dokonce nutná ani jeho instalace, což z něj dělá ideální nástroj pro použití v mnoha situacích i bez správcovského oprávnění k používanému systému.

Tor Browser je postavený na prohlížeči *Firefox*, do kterého je Tor integrován spolu s dalšími nástroji pro zvýšení bezpečnosti při prohlížení webu. Dále je *Tor Browser* nakonfigurován tak, aby minimalizoval možnost identifikace pomocí tzv. *fingerprintingu*.

Fingerprinting je technika identifikování prohlížeče na základě informací, které poskytne webové stránce, na kterou uživatel přistupuje. Jedná se například o informace o operačním systému, jazykovém nastavení, verzi prohlížeče a jeho nastavení [30]. Kombinací těchto informací lze od běžných prohlížečů získat i kolem **20 bitů** identifikující informace, čímž se zařízení stává jedinečným průměrně mezi *milionem* dalších. *Tor Browser* množství poskytované identifikující informace snižuje zhruba na **6,5 bitu**, čímž se stává jedinečným v průměru pouze mezi zhruba *90* dalšími zařízeními a dramaticky tak snižuje možnost sledování.

1.2.4.3 Stem

Stem je knihovna pro programovací jazyk *Python*, pomocí které je možné Tor ovládat. To je užitečné především pro automatizované úkony, například jeho start, změnu používaného *Tor circuitu*, přístup na stránku a podobně [31].

1.2.4.4 Orchid

Orchid je implementace Toru v jazyce *Java*. Její součástí je kromě Tor klienta i knihovna, která může být použita pro integrování Toru přímo do vyvíjené aplikace. Jde o otevřený software, který je součástí projektu *Subgraph OS* [32].

1.2.4.5 Torsocks

Pomocí **Torsocks** je možné obalit libovolnou aplikaci, která používá *SOCKS* protokol, a směrovat tak její síťový provoz skrz síť Tor. To je obzvláště užitečné, když aplikace neumožňuje nastavit *proxy*, přes kterou má komunikovat. Při spouštění zvolené aplikace ji stačí spustit jako parametr programu *torsocks*, tedy např. **torsocks telnet** pro spuštění *telnetu* komunikujícího přes

Tor. Takové obalení a přesměrování přes Tor se také označuje jako **torifikace** [33].

1.2.4.6 Tor2web

Tor2web je webová služba umožňující přistupovat na *onion services* i z prostředí *clearnetu*. Pro přístup na ni stačí v běžném webovém prohlížeči za koncovku **.onion** připojit ještě **.to** [34]. Z adresy *skryté služby* se tak stane doména třetího řádu na doméně **onion.to**, která patří právě službě *Tor2web* a plní funkci *proxy serveru*. Zmíněný *Dream Market* je tedy jejím prostřednictvím možné zobrazit na adrese *lchudifyeqm4ldjj.onion.to*[20].

Tor2web není jediná služba svého druhu, jeví se však jako nejstabilnější a nejtransparentnější ze všech podobných. Používáním takových služeb uživatel za své pohodlí vždy obětuje bezpečnost, což je nutné mít na zřeteli a používat je adekvátně jejich riziku. Navíc, ačkoliv se takový přístup ke *skrytým službám* může zdát pohodlný, odezva bývá příliš dlouhá a pro zasílání více požadavků tak nejsou služby tohoto druhu příliš vhodné.

1.2.5 Další projekty Toru

Dalšími užitečnými nástroji fungujícími na základě Toru jsou například:

- **Tails**, celým názvem The Amnesic Incognito Live System, je operační systém s integrovaným Torem. Systém je spouštěný z vyjímatelného média a nezanechává po sobě žádné stopy.
- **Tor Messenger** je zabezpečená šifrovací klient, jehož komunikace probíhá skrz Tor.
- **Orbot** je Tor klient pro platformu *Android* vytvářející *proxy* a umožňující použít Tor jako *VPN*.
- **Orlib** je *Android* knihovna pro integraci Toru do *Android* aplikací [35].

[20] Funkční k 28. 3. 2018

Kapitola 2

Analýza a návrh systému

Tato kapitola obsahuje návrh systému a analýzu projektu jeho implementace a nasazení. Analytická část se zabývá požadavky, přínosy, riziky a finanční stránkou projektu, stejně tak jako existujícími řešeními pro sběr dat, zejména v prostředí darknetu. Dále je navržena architektura systému jsou popsány funkce jeho jednotlivých komponent.

2.1 Popis a vymezení projektu

Cílem projektu je návrh a vývoj systému pro monitorování marketů na darknetu. Systém je poptáván **sekcí kybernetické kriminality** spadající pod **Národní centrálu proti organizovanému zločinu SKPV Policie ČR**. Ta si od něj slibuje pomoc při objasňování a potírání ilegální činnosti na internetu.

Policii ČR by měl systém sloužit k získávání informací o nabídce zboží, které je na darknetu momentálně dostupné. Bude díky němu mít přehled například o tom, jaké druhy drog jsou aktuálně nejvíce nabízeny nebo jaká je přibližná cena různého ilegálního zboží na černém trhu. Dále může systém pomoci při spojování prodejců a jimi nabízeným zbožím.

Ačkoliv by bylo možné polemizovat, že projekt může PČR ušetřit výdaje spojené s prohledáváním marketů jejími pracovníky, je na projekt pohlíženo jako na **neziskový**.

2.1.1 Požadavky na systém

Jak název napovídá, jádro systému má zajišťovat monitorování, tedy sledování *darknetových obchodů*. Základní funkcionalitou v tomto smyslu je schopnost připojit se k *darknetu* a stáhnout *nestrukturovaná* data o nabízeném sortimentu z webových stránek obchodů. Tato data by následně měla být transformována do *strukturovaných* dat, ve kterých bude možné **vyhledávat**.

Dalším aspektem zmíněného monitorování je opakované stahování dat a **sledování změn**, jako například nové nabídky zboží, změna jeho ceny či jiných parametrů. O těchto změnách by měl umět uživatele **notifikovat**, např. prostřednictvím e-mailu nebo jiné komponenty systému určené k informování o dění na sledovaném trhu.

2.1.1.1 Funkční požadavky

Systém bude umět:

- připojit se k obchodům;
- stáhnout stránky se zbožím, *vyparsovat*[21] z nich data a ta uložit do databáze;
- vyhledávat v získaných datech;
- opakovaně stahovat data a aktualizovat je v databázi;
- notifikovat uživatele o změnách.

2.1.1.2 Nefunkční požadavky

Systém bude:

- modulární,
- multiplatformní,
- uživatelsky přívětivý.

2.2 Přínosy projektu pro Policii ČR

Jak již bylo uvedeno, projekt nebude realizován za účelem zisku, veškeré jeho přínosy jsou tedy nefinančního charakteru. Hlavní přínosy projektu pro PČR jsou:

- Systém pomůže **zmapovat situaci** na *darknetových* marketech, umožní získat:
  - přehled o prodávaném zboží,
  - přehled o ceně a dalších parametrech,
  - informace o prodejcích, jako je počet nabízeného zboží a doba působení [36].

[21] Parsování je extrakce dat z nestrukturovaného textu pomocí definovaných pravidel.

- Systém poskytne **čerstvé informace** o změnách ve zboží:
  - notifikacemi může upozornit na nové druhy drog, podvodů či na únik dat.
- Systém může pomoci v boji proti nelegálním aktivitám a zboží:
  - při spojování zboží s jeho prodejci,
  - při celních kontrolách.
- Získaná data bude možné použít pro statistické účely.
- Pokud používání systému přinese výsledky může tak přispět ke zvýšení reputace PČR.

2.3 Rizika projektu

Fakt, že se jedná o projekt pro veřejný sektor s sebou přináší některá specifika. Tím hlavním je financování, v praxi totiž například platí, že při soutěžení veřejné zakázky je hlavním kritériem co nejnižší cena. To se mnohdy zákonitě podepíše na kvalitě zpracování, ať už se jedná o projekt z libovolného oboru. V případě navrhovaného systému tedy může nastat problém při příliš **vysoké ceně** projektu nebo například při financování jeho dalším rozvoje a údržby.

Rizikem pro projekt je také **špatný návrh architektury**, kterou bude nutné z nějakého důvodu měnit v průběhu projektu. Opatřením pro jeho snížení je pečlivý návrh a dostatečná rezerva v harmonogramu projektu na případné změny.

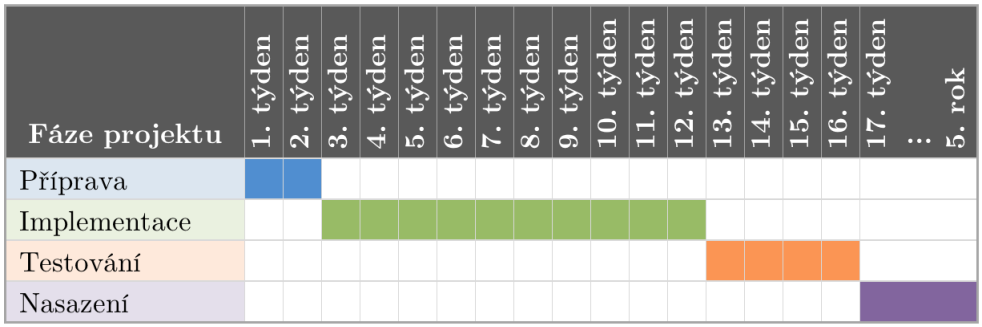
Dalším rizikem, které může nastat, je aktivní **blokování systému** ze strany marketů. Protože se při sběru dat chování systému bude zákonitě lišit od chování běžného uživatele (uživatelé typicky stráví na jednotlivých stránkách určitý čas čtením a neprohlížejí velké množství stránek), mohou toto markety detekovat. Protože není v jejich zájmu pouštět na své weby roboty (ať už kvůli ochraně svých uživatelů nebo pro zabránění zbytečného vytěžování serverů), je možné, že se je budou snažit blokovat. Blokování by přineslo další finanční náklady pro vývoj mechanizmů k jeho obcházení.

Rizikem, které by znamenalo další náklady, jsou i **časté změny struktury marketu** na kterém by systém již fungoval. V případě, že na marketu proběhne změna ve struktuře stránky, bude nutné upravit i způsob parsování informací. Protože ale markety historicky měnily strukturu webu velmi zřídka, je toto riziko nízké.

Protože bude systém závislý na službě třetí strany při řešení *CAPTCHA* ochrany, je nutné uvažovat situaci, kdy tato služba přestane fungovat. V tom případě bude nutné za službu sehnat náhradu. Podobný problém by nastal v případě, že by market implementoval lepší ochranu než pouze opisování obrázku, kterou by externí služba neuměla vyřešit.

2.4 Harmonogram projektu

Byl stanoven harmonogram projektu (obrázek 2.1) začínající po fázi návrhu. Harmonogram má sloužit hlavně jako podpůrný nástroj pro analýzu nákladů na projekt, počítá proto se 40 hodinovým pracovním týdnem a neberou se v úvahu žádné dny volna. Harmonogram také není vztažen k žádnému konkrétnímu datu zahájení.



Obrázek 2.1: Harmonogram projektu

2.4.1 Příprava

Přípravná fáze bude trvat 2 týdny a jednotliví pracovníci se během ní budou seznamovat s projektem samotným a s technologiemi, které v něm budou použity. Dále bude fáze sloužit jako prostor pro případné změny či upřesnění v návrhu.

2.4.2 Implementace

V průběhu implementační fáze bude probíhat hlavní vývoj systému. Implementace je plánována na 10 týdnů. V případě, že bude implementace spět ke konci dříve, bude možné začít systém částečně testovat v jejím průběhu.

2.4.3 Testování

Během testování bude spuštěn pilotní provoz systému. Systém bude v této fázi testován na 2 největších marketech a sběr dat bude přizpůsoben tak, aby byl co nejrychlejší, ale aby se zároveň minimalizovalo blokování markety. Testování je plánováno na 4 týdny.

2.4.4 Nasazení

Nasazení je finální fází projektu a je v něm zahrnut provoz systému do 5 let od spuštění. Během těchto pěti let bude systém podléhat údržbě a rozvoji podle

aktuálních potřeb. Základem v tomto ohledu bude úprava konfigurace pro získávání dat při změně struktury marketů a vývoj modulů pro nové markety, které se objeví na trhu.

2.5 Projektové role

Pro realizaci projektu bude zapotřebí obsadit několik rolí projektového týmu. Mapování jednotlivých rolí na fáze projektu ve kterých je nutné, aby byli přítomni je znázorněno v tabulce 2.1 spolu s jejich počtem a hodinovou mzdou. Fáze projektu jsou označeny jejich počátečními písmeny.

Tabulka 2.1: Projektové role, jejich mapování na jednotlivé fáze projektu a hodinová mzda

Role	**Počet**	**Fáze projektu**	**Hodinová mzda**
Projektový manažer	1	P, I, T	400 Kč
Vývojář	2	P, I	300 Kč
DevOps inženýr[23]	1	P, I, T, N	400 Kč
Tester	1	T	200 Kč

2.6 Analýza nákladů

V této sekci jsou stanoveny náklady na systém, které jsou rozděleny dle jejich charakteru na náklady na **vývoj**, **provoz, údržbu** a **rozvoj**. Z těchto údajů jsou dále vypočítány *celkové náklady na vlastnictví* (TCO).

2.6.1 Náklady na vývoj

Náklady na vývoj systému zahrnují **lidské** náklady vynaložené od zahájení projektu do dokončení jeho testování a jsou vyčíslené v tabulce 2.2.

Tabulka 2.2: Lidské náklady na vývoj projektu

Týmová role	**Počet MD**[25]	**Sazba na MD**	**Suma**
Projektový manažer	80	3 200 Kč	256 000 Kč
Vývojáři	180	2 400 Kč	432 000 Kč
DevOps	80	3 200 Kč	256 000 Kč
Tester	20	1 600 Kč	32 000 Kč
Celkem			**976 000 Kč**

2.6.2 Náklady na provoz

Provozní náklady v sobě zahrnují náklady vynaložené na běh systému během testovací fáze a fáze nasazení systému v běžném provozu na 5 let. Při zjišťování

těchto nákladů hraje velkou roli, zda systém poběží *lokálně* nebo *v cloudu*. Pro snížení nákladů se počítá s využitím bezplatného softwaru, do nákladů na provoz tedy nejsou zahrnuty žádné licence na software. Stejně tak do nákladů není zahrnuta cena služby pro řešení CAPTCHA, protože je zanedbatelná.

2.6.2.1 Lokální nasazení

Když systém běží lokálně, znamená to, že veškerá infrastruktura, na které systém běží, je ve vlastnictví a pod kontrolou společnosti, která systém provozuje.

Hlavní, často uváděnou výhodou provozování systému lokálně, je kontrola nad daty, se kterými systém pracuje. Stinnou stránkou tohoto řešení jsou náklady ve formě nákupu hardware a jeho správy a údržby. Nákup hardware je také nutné dimenzovat odpovídajícím způsobem s výhledem do budoucna a s ohledem na zastarání komponent. Standardně se tedy hardware kupuje spíše výkonnější, aby nějakou dobu „vydržel“. Výčet jednorázových nákladů pro lokální nasazení navrhovaného systému je v tabulce 2.3, roční náklady pak v tabulce 2.4.

Tabulka 2.3: Jednorázové náklady na lokální nasazení systému

Položka	**Náklady**
Server - 4x CPU, 32 GB RAM	40 000 Kč
Diskové pole - 4 TB	15 000 Kč
Instalace serveru	2 400 Kč
Nasazení systému	6 400 Kč
Celkem	**63 800 Kč**

Tabulka 2.4: Roční náklady na provoz systému lokálně

Položka	**Náklady**
Administrace serveru 2 h/týdně	41 600 Kč
Konektivita	1 200 Kč
Elektrická energie	4 200 Kč
Celkem	**47 000 Kč**

2.6.2.2 Nasazení v cloudu

Nasazení systému do cloudu znamená pronájem výpočetních prostředků od jednoho z poskytovatelů cloudových řešení a provozování systému u něj.

To přináší řadu výhod, mezi které patří především zásadní snížení vstupních nákladů při pořizování a instalaci infrastruktury, snadná

škálovatelnost[26] a přesnější výpočet nákladů na provoz. Nevýhodou je, že nad systémem a jeho daty není přímá kontrola, což může být problém při práci s citlivými daty. Zde je ale na místě zvážit, zda poskytovatelé cloudu, kteří do zabezpečení investují ohromné prostředky, nedokáží nakonec data ochránit lépe, než jejich zákazníci s o mnoho menším rozpočtem.

Srovnání předních hráčů na trhu v tomto odvětví podle společnosti Gartner je znázorněno na obrázku 2.2. Z něj vyplývá, že těmi nejvyspělejšími jsou **Amazon Web Services**, **Microsoft Azure** a **Google Cloud Platform**. Proto je při výpočtu nákladů vycházeno z jejich cenových modelů. Protože pro provoz systému v cloudu není třeba žádných investic na zakoupení hardware, jednorázové náklady pro jeho nasazení v tomto případě sestávají jen z lidské práce. Ta byla stejně jako při lokálním nasazení stanovena na **6 400 Kč**. Ceny ročního provozu systému na jednotlivých platformách jsou pak porovnány v tabulce 2.5.

Tabulka 2.5: Srovnání ročních nákladů pro provoz systému v cloudu u jednotlivých poskytovatelů

Položka	**Amazon Web Services**	**Microsoft Azure**	**Google Cloud Platform**
Virtuální server - 4 CPU, 16 GB RAM (hodinová sazba)	4,90 Kč	3,26 Kč	3,64 Kč
Celkem ročně (8760 h)	**42 924 Kč**	**28 558 Kč**	**31 886 Kč**

[26] Škálovatelností se rozumí možnost pružně přizpůsobovat výkon systému aktuálním potřebám.

Obrázek 2.2: Srovnání předních hráčů v oblasti cloudových platforem dle společnosti Gartner [6]

2.6.3 Náklady na údržbu a rozvoj

Během doby provozu systému může být nutná jeho úprava. Jedním z důvodů může být nutnost opravy scrapovací části systému z důvodu změny ve struktuře marketu nebo ve způsobu přihlašování. Dále může nastat potřeba systém rozvíjet o nové funkcionality. Jako v každém systému bude navíc pro jeho správnou funkčnost a možnost rozvoje nutné provádět údržbu formou aktualizací jednotlivých komponent.

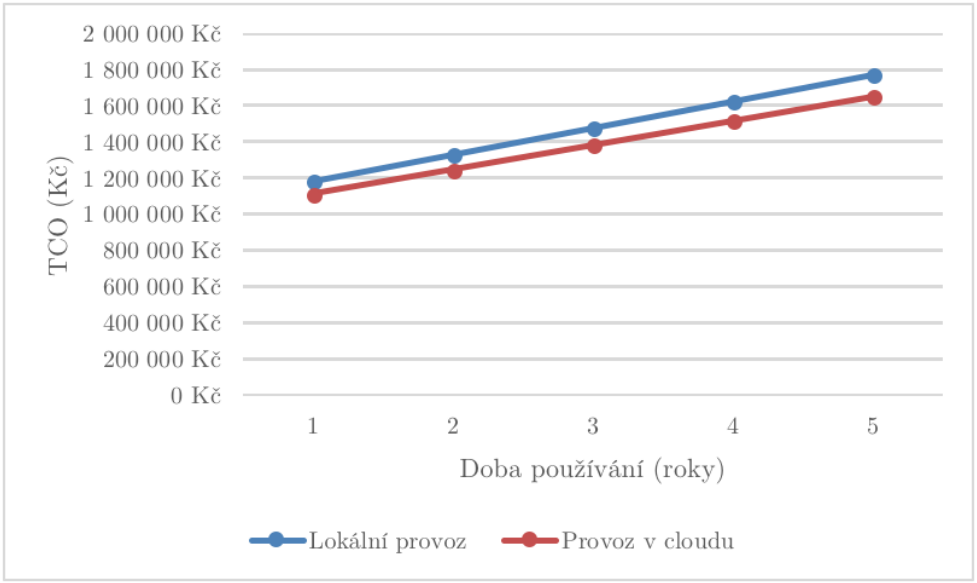
Tyto práce na systému bude standardně provádět DevOps inženýr, potažmo vývojář. V případě většího zásahu a nutnosti koordinace se úprav systému zúčastní i projektový manažer s případným větším projektovým

týmem. Na údržbu a rozvoj bude ročně vyhrazeno 100 000 Kč, což pokrývá přibližně 30 man-dayů práce DevOps inženýra. Velké zásahy bude nutné řešit navýšením rozpočtu.

2.6.4 Total cost of ownership

Total cost of ownership (TCO) se do českého jazyka překládá jako „celkové náklady na vlastnictví“, jde tedy o souhrn všech nákladů vynaložených k tomu, aby byl systém funkční a v provozu. Tento údaj je ze své podstaty velmi důležitým při rozhodování o realizaci projektu.

Srovnání TCO při provozu systému lokálně a v cloudu dle počtu let využívání systému je zaznamenané v tabulce 2.6 a znázorněné v grafu na obrázku 2.3. Z obojího je vidět, že způsob provozu má na celkové náklady vzhledem k ostatním položkám poměrně nízký vliv. Při stanovování provozních nákladů byl však uvažován provoz *24/7/365*, který bude zřejmě reálně o mnoho nižší. Z toho důvodu budou skutečné náklady za cloudové řešení výrazně nižší, jelikož většina velkých cloudových platforem nabízí účtování po minutě nebo podobně krátkých časových úsecích.



Obrázek 2.3: Srovnání TCO při provozu systému lokálně a v cloudu v závislosti na počtu let jeho používání

Tabulka 2.6: Srovnání TCO při provozu systému lokálně a v cloudu dle počtu let jeho používání

Druh nákladů	**Lokální provoz**	**Provoz v cloudu**
Vývoj systému	976 000 Kč	976 000 Kč
Nasazení systému	63 800 Kč	6 400 Kč
Roční provoz systému	47 000 Kč	34 456 Kč[28]
Roční údržba a rozvoj	100 000 Kč	100 000 Kč
TCO na 1 rok	**1 186 800 Kč**	**1 116 856 Kč**
TCO na 2 roky	**1 333 800 Kč**	**1 251 312 Kč**
TCO na 3 roky	**1 480 800 Kč**	**1 385 768 Kč**
TCO na 4 roky	**1 627 800 Kč**	**1 520 224 Kč**
TCO na 5 let	**1 774 800 Kč**	**1 654 680 Kč**

Z výše uvedeného vychází z hlediska nákladů nasazení systému do cloudu jako výhodnější varianta. V případě jeho provozu Policií ČR by ale přesto bylo *nutné*, aby systém běžel lokálně. Dle vyjádření zástupců PČR totiž musí mít policie data pod kontrolou, přestože se jedná pouze o monitorování třetí strany.

2.7 Metriky pro měření úspěšnosti projektu

Projekt lze zpravidla označit za úspěšný v případě, že jsou naplněny všechny jeho cíle. Protože takové cíle mohou být obtížně měřitelné, je vhodné stanovit **metriky**, které lze kvantitativně ohodnotit, a určit hodnoty požadované pro to, aby bylo projekt možné označit za úspěšný.

Pro potřeby posouzení úspěšnosti projektu monitorovacího systému byly metriky stanoveny dle tabulky 2.7.

Tabulka 2.7: Metriky pro měření úspěšnosti projektu

Měřená metrika	**Požadovaná hodnota**
Podíl zpracovaných položek vůči celkovému počtu položek na marketu	> 50 %
Doba funkčnosti systému bez potřeby zásahu	> 1 měsíc
Počet relevantních dat	> 80 %
Skutečná pracnost nakonfigurování systému pro sledování nového marketu	< 3 MD
Skutečná pracnost úpravy konfigurace v případě změny na marketu	< 2 MD
Překročení plánovaného rozpočtu na vývoj	< 10 %

2.8 Analýza struktury webů darknetových obchodů

Tato sekce obsahuje analýzu struktury *marketů* a *vendor shopů* na darknetu, které budou souhrně označovány jako **darknetové obchody**.

Skryté služby jsou většinou jednoduché, bez množství grafických prvků a efektů, které by zbytečně zpomalovaly už tak poměrně pomalý přenos dat na Toru. To platí i o darknetových obchodech. Jejich webové stránky jsou orientované hlavně na funkčnost a zabezpečení.

2.8.1 Markety

Jak bylo naznačeno v sekci 1.1.3.1, *darknetové markety* se v mnohém podobají svým protějškům na clearnetu. Market slouží jako místo pro zveřejňování zboží a služeb prodejci a poskytuje mechanizmy k jejich prodeji. V sortimentu je většinou možné vyhledávat a bývá rozdělený do kategorií, podle kterých ho lze filtrovat.

Každá položka má typicky dva způsoby zobrazení. Jedním z nich je její miniatura v seznamu mezi ostatními, kdy jsou zobrazeny jen nejdůležitější informace. Druhým způsobem zobrazení je její detail, kdy je zobrazen celý popis zboží, který je často doplněn recenzemi o jeho prodejci.

Na detailu konkrétní položky je standardně možné zboží vložit do *nákupního košíku* a posléze ho zakoupit. Jako platidlo se zpravidla používají *kryptoměny* jako jsou *Bitcoin*, *Monero* nebo *Bitcoin Cash*. Pro platbu se používá jejich zůstatek na uživatelově účtu na marketu, který si tedy musí nejdříve nabít. Markety často nabízí platbu pomocí *escrow*, kdy jsou peníze do potvrzení přijetí zboží zákazníkem drženy marketem.

Díky integrovanému platebnímu systému a systému recenzí je tedy pro nákup na marketu nutné, aby byl uživatel přihlášen ke svému uživatelskému účtu. Většina marketů dokonce požaduje přihlášení už před možností prohlížení zboží, kdy je během něj navíc nutné vyplnit CAPTCHA test, kterým se weby chrání proti přístupu robotů. Založení uživatelského účtu však bývá snadné, pro zachování anonymity totiž zpravidla stačí jen zvolit uživatelské jméno a heslo, nebývá vyžadována ani e-mailová adresa.

Velikost sortimentu na těch největších z marketů se pohybuje v řádech tisíců až desesetitisíců položek. Níže je seznam těch historicky nejpopulárnějších[29] [37]:

[29]Protože se bezpečnostní složky snaží obchodu s ilegálním zbožím zamezit a markety zavírají (odpojují a zabavují jejich servery), seznam z principu časem zastará. Markety bez poznámky o jejich uzavření byly na svých adresách funkční k datu 1. 4. 2018.

- Silk Road *(**uzavřen 2013**)*,
- Silk Road 2.0 *(**uzavřen 2014**)*,
- Agora *(agorahooawayyfoe.onion, **uzavřen 2015**)*,
- AlphaBay *(pwoah7foa6au2pul.onion, **uzavřen 2017**)*,
- Hansa *(hansamkt2rr6nfg3.onion, **uzavřen 2017**)*,
- Dream Market *(lchudifyeqm4ldjj.onion)*,
- Wallstreet Market *(wallstyizjhkrvmj.onion)*,
- Point Free Market *(pointgg344ghbo2s.onion)*.

2.8.2 Vendor shopy

Vendor shopy jsou obchody, které pochází pouze od jeho provozovatele, tedy obdobně, jako v případě *clearnetových* e-shopů. Nabídka na nich bývá oproti marketům velmi malá, protože se prodejci většinou specializují na úzkou skupinu zboží.

Proces prodeje zboží probíhá většinou „na dobré slovo“, kdy si zákazník ujasní s prodejcem objednávku prostřednictvím e-mailové komunikace, odešle mu domluvenou částku, a prodejce mu následně odešle zboží. Protože se zpravidla jedná o ilegální zboží, bývá pravidlem používat při komunikaci šifrování *PGP*[30]. Stejně tomu je i při komunikaci s prodejci na marketech.

Zdá se, že provozovatelé *vendor shopů* většinou nabízejí své zboží i na marketech. Svůj vlastní obchod tak zřejmě používají pouze jako kanál pro prodej bez provize marketu, případně ke vzbuzení dojmu věrohodnosti a profesionality. To v kombinaci s malým sortimentem a nízkou frekvencí změn v nabídce snižuje motivaci vkládat úsilí a prostředky do monitorování *vendor shopů*.

2.8.3 Analýza Dream Marketu

Dream Market funguje od roku *2013* a se zbožím čítajícím více než *100 000* položek aktuálně je největším z fungujících *darknetových* marketů. Jeho hlavní adresa je *lchudifyeqm4ldjj.onion*, ale v případě její nefunkčnosti je dostupný i na několika dalších adresách, které jsou uvedeny na jeho webu.

Pro přístup na *Dream market* je vyžadována registrace a přihlášení je chráněno proti přístupu robotů pomocí CAPTCHA. Po přihlášení je uživateli zobrazena nabídka zboží, které je rozděleno do kategorií, je jím možné filtrovat dle kritérií a vyhledávat v něm, viz. obrázek C.1. Hlavními kategoriemi jsou:

[30]Pretty good privacy, zkráceně PGP, je šifrovací standard používaný hlavně k šifrování e-mailové komunikace.

- digitální zboží,
- drogy,
- drogové příslušenství,
- služby,
- ostatní.

Jednotlivé položky sortimentu marketu jsou zobrazeny v mřížce jako miniatury a obsahují všechny informace, které jsou pro zákazníka důležité. Jsou to:

- **název** zboží;
- miniatura **obrázku**;
- **cena** v měně, kterou je možné nastavit v uživatelském profilu;
- uživatelské **jméno** prodávajícího;
- počet úspěšných transakcí a **hodnocení** prodávajícího (část recenzí je převzata z ostatních marketů, které si prodávající spároval se svým účtem);
- **země**, ze které prodejce zboží odešle a do kterých ho doručuje (může být důležité z důvodu celních prohlídek);
- informace o použití **escrow**.

Po kliknutí na položku je zobrazeno *javascriptové*[31] okno, obsahující ty samé informace a tlačítko pro přechod na stránku s úplným popisem zboží. Toto okno lze přeskočit vypnutím *javascriptu* ve webovém prohlížeči. Kliknutím na název zboží se pak uživatel dostane přímo na jeho stránku. Tady se kromě detailních informací o produktu nalézají i recenze prodávajícího, a konečně i tlačítko pro vložení do nákupního košíku, spolu s výběrem kryptoměny, ve které si uživatel přeje zaplatit.

[31] Javascript je programovací jazyk hojně používaný na webových stránkách, často pro dynamické zobrazování obsahu nebo pro ovládání prvků stránky.

2.8.4 Analýza Wallstreet Marketu

Wallstreet Market je oproti *Dream Marketu* co do počtu zveřejněných nabídek řádově menší, v době psaní práce jich bylo pouze lehce přes *8 000*. Dalšími zajímavými čísly, která market zveřejňuje je *2 800* prodejců a *362 000* zákazníků. Ve srovnání s *Dream Marketem* působí *Wallstreet Market* modernějším dojmem, protože je postavený na *Bootstrapu*[32].

Rozdíl je i v přístupu na market. Uživatel Wallstreet Marketu rovněž musí být pro prohlížení jeho obsahu přihlášený, CAPTCHA je zde ale použita hned dvakrát. Poprvé ji musí návštěvník vyplnit už při načtení úvodní stránky marketu a podruhé pak během přihlašování.

Po přihlášení se uživateli zobrazí stránka podobná obrázku C.2, na které si může začít prohlížet nabídku zboží. *Wallstreet Market* nabízí podobné možnosti filtrování a vyhledávání jako *Dream Market*, jen rozdělení do kategorií se zdá být podrobnější. Shodně na tom konkurenti jsou i v množství informací zobrazovaných v miniaturách. Po kliknutí na miniaturu konkrétní nabídky se uživateli zobrazí přímo její stránka bez jakýchkoliv vyskakovacích oken.

Na Wallstreet Marketu bohužel nelze nastavit měna, ve které je cena zboží zobrazována. Zdá se, že prodejce si během zadávání ceny zboží zvolí i zobrazovanou měnu. Ta je převedena na jednu z kryptoměn až při platbě. Některé položky tak mají svou cenu zobrazovanou v amerických dolarech a některé v eurech, což může činit potíže při zpracování dat připravovaným systémem.

2.9 Analýza způsobu získávání dat

Prvním krokem pro monitorování *darknetových obchodů* je získání dat o jejich produktech. Žádný z prozkoumávaných obchodů bohužel neposkytuje způsob, jakým získat strukturovaná data o jeho produktech, například pomocí *API*[33], často se tomu dokonce aktivně brání používáním CAPTCHA ochrany. Z toho důvodu je nutné se uchýlit k získávání informací z nestrukturovaných dat, konkrétně z *HTML* dokumentů obchodů.

Informacemi o zboží, které by bylo vhodné získávat jsou:

- adresa stránky se zbožím,
- ID zboží v obchodě,
- kategorie zboží,

[32] Bootstrap je populární framework používaný pro vývoj responzivních webových stránek, tedy takových, které se přizpůsobují velikosti displeje, na kterém jsou právě zobrazovány.

[33] Application programming interface (API) je rozhraní aplikace, které programátorovi umožňuje s aplikací komunikovat, ovládat ji a získávat od ní data.

- název zboží,
- popisek zboží,
- cena zboží,
- jméno prodávajícího,
- název obchodu.

Dále jsou diskutovány dva hlavní způsoby získávání dat.

2.9.1 Manuální získávání dat

Manuálním získávání dat se rozumí extrakce dat ze zdroje, která je vykonávána člověkem. V našem případě se tedy jedná o procházení webových stránek *darknetového obchodu*, identifikace informací o nabízeném zboží a jejich ukládání do databáze, ve které je pak možné data vyhledávat, filtrovat je a dále s nimi pracovat.

Je patrné, že manuální přístup není vhodný pro opakované získávání dat z velkých zdrojů, protože vyžaduje velké množství práce. Zde je na místě úkony prováděné při sběhu dat zautomatizovat.

Získávat data manuálně bývá rešením u zdrojů s malým množstvím dat, která navíc nejsou příliš často měněna. Vývoj automatického řešení se zde většinou nevyplatí, což se zdá jako případ automatického monitorování *vendor shopů*, které zpravidla mají velmi omezenou nabídku a navíc existuje poměrně vysoká šance, že se jejich nabídka objeví i na *marketech*, který je pro automatické monitorování již o mnoho zajímavější.

2.9.2 Automatické získávání dat

Jak již bylo zmíněno, automatizace slednování nabídky zboží se jeví jako zajímavé hlavně u *darknetových marketů*. Hlavním důvodem pro automatizaci je velké množství nabídek, které navíc mají standardizovaný tvar.

Základními pojmy v oblasti automatizace získávání dat na webu jsou *web crawling* a *web scraping*. Ačkoliv se často tyto pojmy při užívání zaměňují, exaktně je lze odlišit následovně [38]:

Jak již bylo zmíněno v sekci 1.1.1 zabývající se *surface webem*, jako **web crawler** se označuje program, který navštěvuje webové stránky a prochází odkazy, které se na nich vyskytují. Objevuje tak další stránky, ze kterých postupuje dále a mapuje tak, tak jsou mezi sebou propojeny.

Dá se říci, že úloha **web scraperu** je co se týče získávání informací opačná. Pohybuje se pouze v rámci jedné webové aplikace a získává z ní co nejvíce užitečných dat, která převádí do strukturovaných dat – typicky je ukládá do databáze. Protože bývá struktura webů různá, často je nutné web scraper k použití na konkrétním z nich adekvátně nakonfigurovat. Typicky se jedná

o definování, ve kterých elementech *DOM modelu*[34] jsou data, která má *web* scraper získat.

Protože má navrhovaný systém sbírat data v rámci jednotlivých marketů, jejichž adresy budou předem známy, je zřejmé, že jeho jádrem bude právě web scraper. Ten bude muset být nakonfigurovaný podle jejich struktury. Výzvou, kterou bude muset web scraper řešit, je přístup k marketům chráněných přihlášením a CAPTCHA.

2.10 Analýza existujících řešení

Tato sekce se věnuje existujícím nástrojům pro sběr dat z webu. Takových nástrojů je velké množství a jejich porovnávání se věnuje mnoho článků i dedikovaných webových stránek[35]. Výběr ideálního řešení pro připravovaný systém tak není snadný.

Na základě průzkumu na internetu bylo vybráno několik programů a služeb, které byly k podobným projektům v minulosti použity nebo se jeví být vhodnými kandidáty. Níže jsou popsány jejich základní vlastnosti.

2.10.1 Webhose.io

Webhose.io je společnost, která se zabývá sběrem dat pomocí crawlování a scrapování webu. Získaná data pak nabízí prostřednictvím svých *endpointů*, kterých se její uživatelé mohou dotazovat. Data jsou pomocí základních filtrů rozdělena na e-commerce, publicistiku, audiovizuální média, recenze, a nově i dark web [40]. *Webhose.io* je tak ojedinělou komerční službou poskytující data z tohoto zdroje. Při základní, bezplatné licenci umožňuje uživateli zaslat 1 000 dotazů na každý z endpointů.

2.10.2 Octoparse

Octoparse je scrapovací nástroj konfigurovatelný přes grafické rozhraní. Základní licence je zdarma a limituje množství exportovaných dat na 10 000 záznamů. *Octoparse* dále nabízí i *cloudové* řešení s užitečnými funkcemi jako je rotace IP adres, ze kterých je *scrapováno*, či přístup pomocí *API* [41].

[34]DOM je objekt, do kterého je převeden HTML dokument při načtení webovým prohlížečem. Jde o stromovou strukturu tvořenou jednotlivými prvky dokumentu, se kterými je možné pracovat, například získávat jejich hodnoty nebo je měnit [39].

[35]Viz. například `https://www.ijser.org/researchpaper/Comparison-of-Open-Source-Crawlers--A-Review.pdf`, `http://www.scraping.pro`, `http://bigdata-madesimple.com/top-50-open-source-web-crawlers-for-data-mining/` nebo `https://www.capterra.com/data-extraction-software/`

2.10.3 Dexi.io

Služba **Dexi.io** (dříve CloudScrape) je komplexní *crawlovací a scrapovací* cloudový nástroj obsahující prémiové funkce, jako je rozpoznávání CAPTCHA pomocí *OCR*[36], vyplňování formulářů, integraci s externími službami, či použití externí proxy. Služba je placená a na vyzkoušení je k dispozici hodina běhu robota [42].

2.10.4 Apify

Stejně jako *Octoparse*, **Apify** nabízí cloudové řešení pro crawlování a scrapování webu. Uživatelovi pak data poskytuje prostřednictvím *API* a v základní, bezplatné licenci umožňuje shromáždit data z 5 000 stránek měsíčně.

2.10.5 Scrapy

Scrapy je framework v jazyce *Python* pro vytváření *scriptů*[37] pro extrakci dat z webu. Jde o *otevřený software*, který je možné provozovat lokálně nebo jako cloudové řešení. Pro uživatele je ke spouštění vytvořených *scrapy scriptů* k dispozici placená platforma **Scrapycloud**, která však nabízí běh jedné instance zdarma a podporuje *Elastic Search*, což je *fulltextový* vyhledávací *engine*. Pomocí nástroje **Scrapyd** je pak možné tyto scripty spouštět i z vlastního serveru [43].

Scrapy je v prostředí *Toru* používán např. v projektu **Fresh Onions**[38], který je zaměřen na monitorování funkčnosti *skrytých služeb* na *Toru*.

2.10.6 Webscraper.io

Webscraper.io je scrapovací nástroj skládající se ze dvou částí. Jednou z nich je *rozšíření* pro prohlížeč *Google Chrome*, který slouží k vytvoření konfigurace podle konkrétní webové stránky, ze které si uživatel přeje získat data [44].

Druhou částí je samotný scraper, což je placená cloudová služba, do které se vytvořená konfigurace nahraje a scrapování se spustí.

[36] Jako Optical Character Recognition (OCR) software se označují programy, které dokáží rozpoznat text v obrázku a na výstupu ho poskytnout uživateli. Často se tato technologie používá při zpracování skenovaných dokumentů.

[37] Jako scripty bývají označovány jednoduché programy určené k provádění automatizované činnosti.

[38] Zdrojový kód Fresh Onions je zveřejněn na GitHubu na adrese `https://github.com/dirtyfilthy/freshonions-torscraper`, běžící služba je pak dostupná jako skrytá služba na adrese `http://zlal32teyptf4tvi.onion`.

2.10.7 Apache Nutch

Apache Nutch je software v jazyce Java a bývá označován za zlatý standard scrapování [45]. Jsou dostupné dvě řady *Nutche*, které jsou vyvíjeny nezávisle. Řada *1.x* je standardní vyspělý crawler používající *Apache Hadoop* k ukládání dat. Řada *2.x* oproti tomu nabízí možnost škálovatelnosti s použitim *NoSQL* úložiště. *Apache Nutch* je modulární a poskytuje rozhraní pro podporu *plug-inů*[39], například pro *parsování* pomocí *Apache Tika* a pro *indexování* pomocí *Apache Solr* nebo *Elastic Search* [46].

Pro použití *Nutche* je nejdříve třeba nakonfigurovat jeho *crawlovací* část. Zde je třeba nastavit počáteční URL, na které *Nutch* začne sbírat odkazy na další stránky. Rozsah těchto odkazů nutné omezit jen na ty žádoucí, což jsou v případě scrapování jednoho webu typicky odkazy vedoucí pouze na jeho doménu. Toto omezení se provádí *regulárním výrazem*. Dále je možné zakázat některé další odkazy, například pokud by se kliknutím na něj robot sám odhlásil. *Nutch* dále automaticky *blacklistuje*[40] odkazy, na které již v minulosti přistoupil, aby tím zamezil zacyklení.

Po stažení stránek *Nutch* použije *plug-in*, pro extrakci požadovaných dat ze stránky a uloží je do indexu.

Nutch byl pro crawlování na *Toru* použit v projektu[41], jehož cílem bylo indexování *skrytých služeb*. Autor ho provozoval na operačním systému *Ubuntu* a používal nástroje jako kontejnerovou platformu *Docker*, fulltextový index *Elastic Search*, nástroj *Kibana* pro vizualizaci dat a program pro střídání proxy za účelem zamezení blokování ze strany serverů při detekci přístupu robota.

2.10.8 Knihovny Request a BeautifulSoup

Request a **BeautifulSoup** jsou knihovny pro programovací jazyk *Python*. **Request** je knihovna sloužící k posílání požadavků na server, která si mimo jiné dokáže poradit i s přihlášením a držením *cookie*[42]. **BeautifulSoup** umožňuje *parsovat* text, což je jedna ze základních věcí potřebných při scrapování. Kombinací těchto knihoven je tak možné si naprogramovat vlastní crawler v jazyce *Python* [47].

2.10.9 Wget

Jako způsob pro crawlování webových stránek lze použít i nástroj **wget**, který je důvěrně známý většině uživatelům příkazové řádky na *unixových*

[39] Plug-in je program malého rozsahu naprogramovaný pro použití s jiným, již existujícím programem, do kterého je integrován.

[40] Blacklistováním se rozumí přidání na seznam zakázaných položek, se kterými aplikace nemá pracovat.

[41] Postup při realizaci projektu je popsán na `http://www.boredhackerblog.info/2017/07/crawling-tor-using-apache-nutch.html`.

[42] HTTP cookie, česky sušenka, jsou data, která webový prohlížeč posílá serveru pro svoji identifikaci.

operačních systémech. Jde o program, který je určený pro stahování souborů prostřednictvím protokolů *HTTP*, *HTTPS*, *FTP* a *FTPS*. *Wget* funguje pouze v příkazové řádce (nemá grafické uživatelské rozhraní) a není interaktivní. Z toho důvodu je vhodný pro použití při automatizovaných činnostech ve *scriptech* a lze ho použít i při crawlování [48].

Wget byl jako *crawlovací* nástroj použit pro projekt výzkumníka vystupujícího pod pseudonymem *Gwern Branwen*, který ho spolu s dalšími nástroji v letech 2013 – 2015 téměř každodenně používal ke sběru dat z *darknetových* marketů a přidružených diskuzních fór. Jedná se o nejrozsáhlejší známý projekt svého druhu. Získaná data o velikosti *1,6 TB* byla spolu s kompletním popisem jejich sběru zveřejněna[43] na Gwernových webových stránkách.

Jak na svém webu Gwern dále uvádí, *wget* má chybu v implementaci *blacklistování*. Ta spočívá v tom, že soubor i přes jeho zařazení na blacklist stáhne a následně ho smaže. Přestože se jedná o chybu zjištěnou již v roce 2003, dosud nebyla opravena. Pro dosažení správného chování je tedy třeba stránky *blacklistovat* mimo *wget*, například na použitém *proxy serveru* [36].

2.10.10 Nástroje pro řešení CAPTCHA

Jak již bylo v práci zmíněno, markety bývají dostupné pouze po přihlášení, které bývá proti přístupu robotů chráněno pomocí CAPTCHA. Existuje několik možných způsobů, jak se s přítomností ochrany CAPTCHA vypořádat:

1. Vyplnit ji **ručně** kdykoliv, kdy bude třeba.
2. Použít na její řešení **OCR** software.
3. Využít některou ze **služeb** pro řešení CAPTCHA. Takové služby nabízejí *API*, na které stačí odeslat CAPTCHA obrázek a služba vrátí její text. Tyto služby jsou placené a pro zaručení správnosti zpravidla používají levnou lidskou sílu z rozvojových zemí. Příklady takových služeb jsou **anti-captcha.com**, **deathbycaptcha.com** a **bypasscaptcha.com**.
4. Při odesílání požadavků na server používat **cookie** z již přihlášeného sezení. Tento přístup při scrapování zvolil například *Gwern*.

2.11 Výstupy analýzy

Z analýzy v rešeršní části práce vyplynulo, že data lze z marketů získávat pouze v nestrukturované podobě jako *HTML dokumenty*. Tyto dokumenty je nutné nejprve stáhnout pomocí **crawleru** a následně zpracovat pomocí **scraperu**. Ten z nich bude procházením jednotlivých prvků extrahovat předdefinovaný

[43] Získaná data jsou, spolu s popsaným postupem sběru, použitým softwarem a diskuzí, dostupná na `http://www.gwern.net/DNM-archives`.

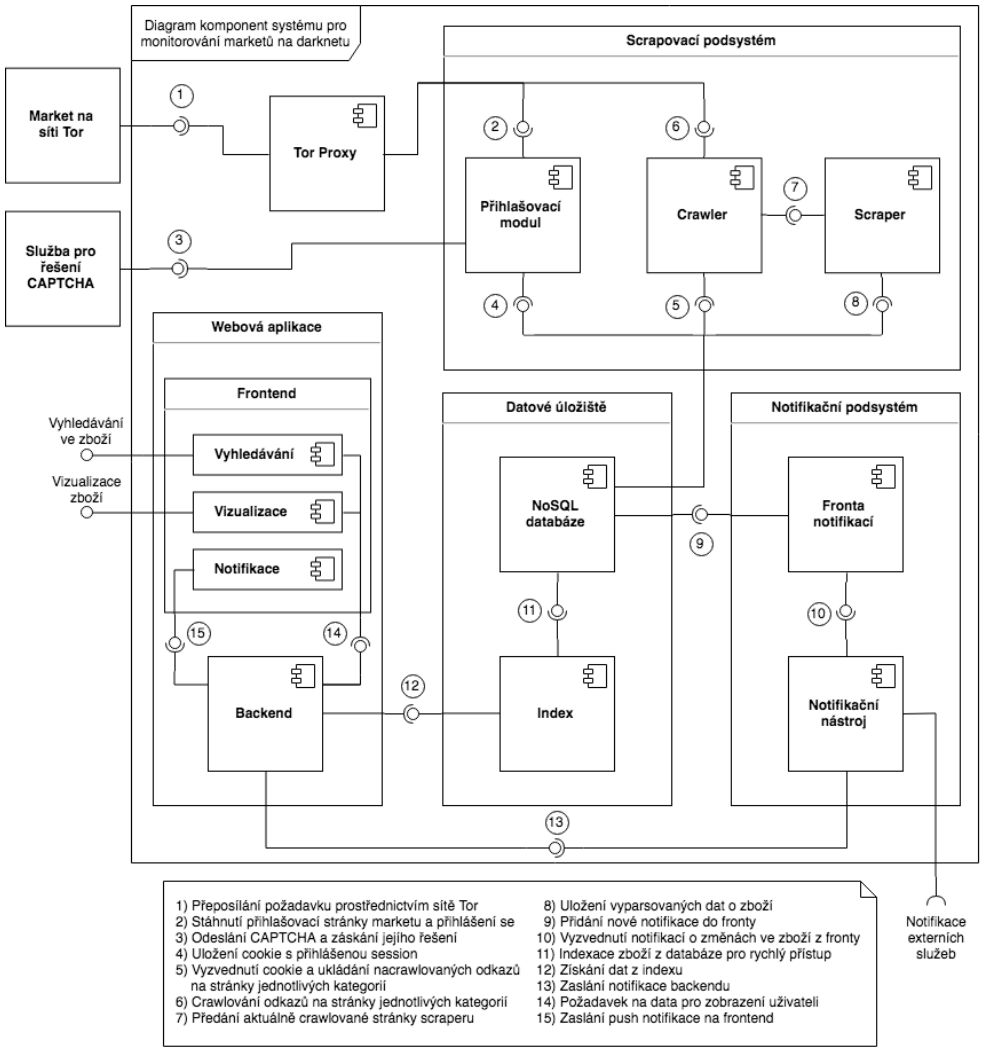
obsah, jako je název položky, její cena, apod. Jako crawler a scraper lze použít některé z existujících nástrojů, případně naprogramovat vlastní řešení s použitím dostupných knihoven.

Proto, aby bylo možné z marketu dokumenty stáhnout, je nutné se k němu nejdříve připojit a přihlásit. K marketu se je nutné připojovat pomocí *Toru*, který k tomuto účelu používá *SOCKS5 proxy*, použitý crawler tedy tento protokol musí podporovat. Další možností je mezi crawler a *Tor proxy* umístit ještě jeden *proxy server*, který bude umožňovat překlad *HTTP* na *SOCKS5*.

Protože se markety snaží automatizovanému sběru dat z jejich stránek zabránit, vyžadují pro přistupu k jejich obsahu přihlášení, které je zpravidla chráněné pomocí CAPTCHA a někdy i *obfuskací* přihlašovacího procesu. Ke spolehlivému automatickému řešení CAPTCHA ochrany je vhodné použít některou ze služeb, které k jejich řešení používájí lidskou sílu. Kvůli *obfuskaci* je vytvoření univerzálního řešení pro přihlášení se k různým darknetovým marketům značně ztíženo, a zdá se, že pro každý market je nutné vytvořit samostatný přihlašovací modul.

2.12 Architektura systému

Byl navržen modulární systém, který je znázorněn v diagramu komponent na obrázku 2.4.



Obrázek 2.4: Diagram komponent systému pro monitorování marketů na darknetu s popisem jednotlivých rozhraní

2.13 Systémové komponenty

V této části jsou jednotlivé součásti systému podrobně popsány.

2.13.1 Tor proxy

Jako **Tor proxy** je označena *SOCKS5 proxy*, která je součástí instalace Toru. Prostřednictvím této proxy **přihlašovací modul** a **crawler** přistupuje na stránku marketu na darknetu, pro který jsou nakonfigurovány. Pro použití této proxy tak musí přihlašovací modul i crawler podporovat protokol *SOCKS5*. V případě, že protokol alespoň jeden z nich nepodporuje a používá pro komunikaci pouze protokol *HTTP*, je třeba navíc použít některý z nástrojů pro překlad HTTP na SOCKS5 a předřadit ho před Tor proxy.

2.13.2 Přihlašovací modul

Protože je pro přístup k marketům vyžadováno přihlášení, je zapotřebí, aby součástí podsystému, který bude mít na starosti scrapování dat, byl **přihlašovací modul**.

Tento modul je zodpovědný za přihlášení k systému a uložení *cookie* s přihlášenou *session* do **databáze**. Tu si z ní následně vyzvedne **crawler**, který se díky ní bude moci po marketu pohybovat jako přihlášený uživatel. Je možné, že bude uživatelský účet po zaslání velkého množství požadavků zablokován. Pro snížení této šance je vhodné, aby modul přihlásil takových účtů více a crawler je střídal.

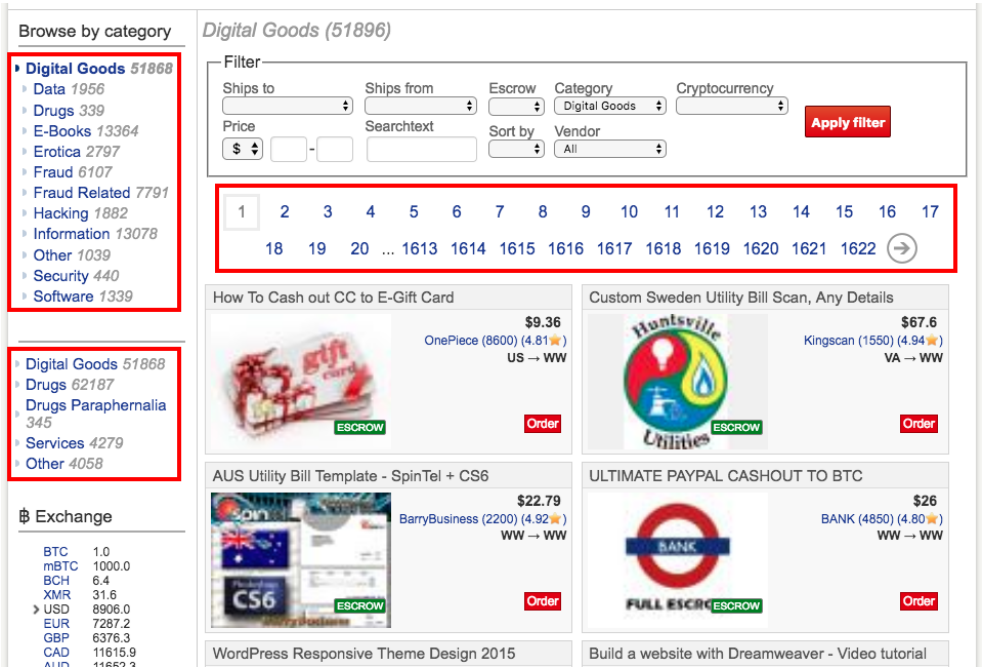
Pro přihlášení uživatelského účtu je nezbytné zadat jeho uživatelské jméno a heslo. Seznam těchto údajů od již vytvořených účtů bude přihlašovacím modulem načten z jeho konfiguračního souboru. Druhou variantou je vytvoření uživatelských účtů na marketu samotným modulem a jejich následné přihlášení.

Protože se darknetové markety snaží automatizovanému sbírání informací bránit, je při přihlašování nutné splnit CAPTCHA, tedy většinou správně opsat zdeformovaný řetězec písmen a číslic z obrázku. Tuto ochranu lze obejít využitím některé ze služeb pro řešení CAPTCHA. Tyto služby k řešení používají lidskou sílu a poměrně spolehlivě v řádu jednotek až desítek sekund vrátí řetězec formou textu. CAPTCHA obrázek je možné takové službě zaslat pomocí její *API*, čehož přihlašovací modul bude využívat.

Markety jako druhotnou ochranu proti automatizaci přístupu *obfuskují* zdrojový kód přihlašovací stránky. To celý proces přihlašování značně znepřehlední a vede k nutnosti vyvinout přihlašovací modul „na míru“ konkrétnímu marketu.

2.13.3 Crawler

Úlohou **crawleru** je procházet *HTML* dokumenty a hledat v nich podle předem definovaných pravidel odkazy na další stránky s obsahem. V případě crawlování marketů bude třeba, aby crawler našel stránky s jednotlivými kategoriemi zboží. Protože je na marketech nabízeného zboží velké množství, je jeho zobrazení rozděleno na více stran (v případě *Dream Marketu* až 4 000 stran po 32 položkách), které bude crawler sbírat. Příklad odkazů pro sběr je znázorněn na obrázku 2.5.



Obrázek 2.5: Odkazy ke sběru crawlerem na stránce Dream Marketu (vyznačeny červeně) [7]

Crawler po svém spuštění z **databáze** získá *cookie* s přihlášenou *session*, kterou pro něj připravil **přihlašovací modul** a díky níž se bude moci po marketu pohybovat jako přihlášený uživatel. Výchozí bod pro pohyb crawleru se nazývá *seed*[44]. Jde o adresu stránky se zbožím, na které crawler nalezne odkazy na další stránky, uloží je do databáze a předá HTML dokument **scraperu**, aby z něj získal data o jednotlivých položkách.

Stejný postup crawler aplikuje na všechny získané adresy. Do databáze se každá adresa ukládá pouze jednou, crawler tedy při ukládání nově nacrawlo-

[44]Seed (česky semeno) je v informatice obecně chápán jako počáteční hodnota, která slouží k výpočtu dalších hodnot. V tomto kontextu se jedná o počáteční adresu stránky, ze které crawler získá další adresy.

vaných adres kontroluje, zda již v databázi neexistují. Po prohledání každé stránky se do databáze navíc uloží časový údaj o jejím posledním úspěšném zpracování. Aby se předešlo zacyklení a zároveň byla co největší šance na nalezení nového zboží, crawler vždy jako další adresy zpracovává ty s nejstarším datem poslední návštěvy. Crawler navíc typicky zpracovává několik stránek najednou, z databáze si tedy v jednom kroku vyzvedne například 20 nejstarších adres.

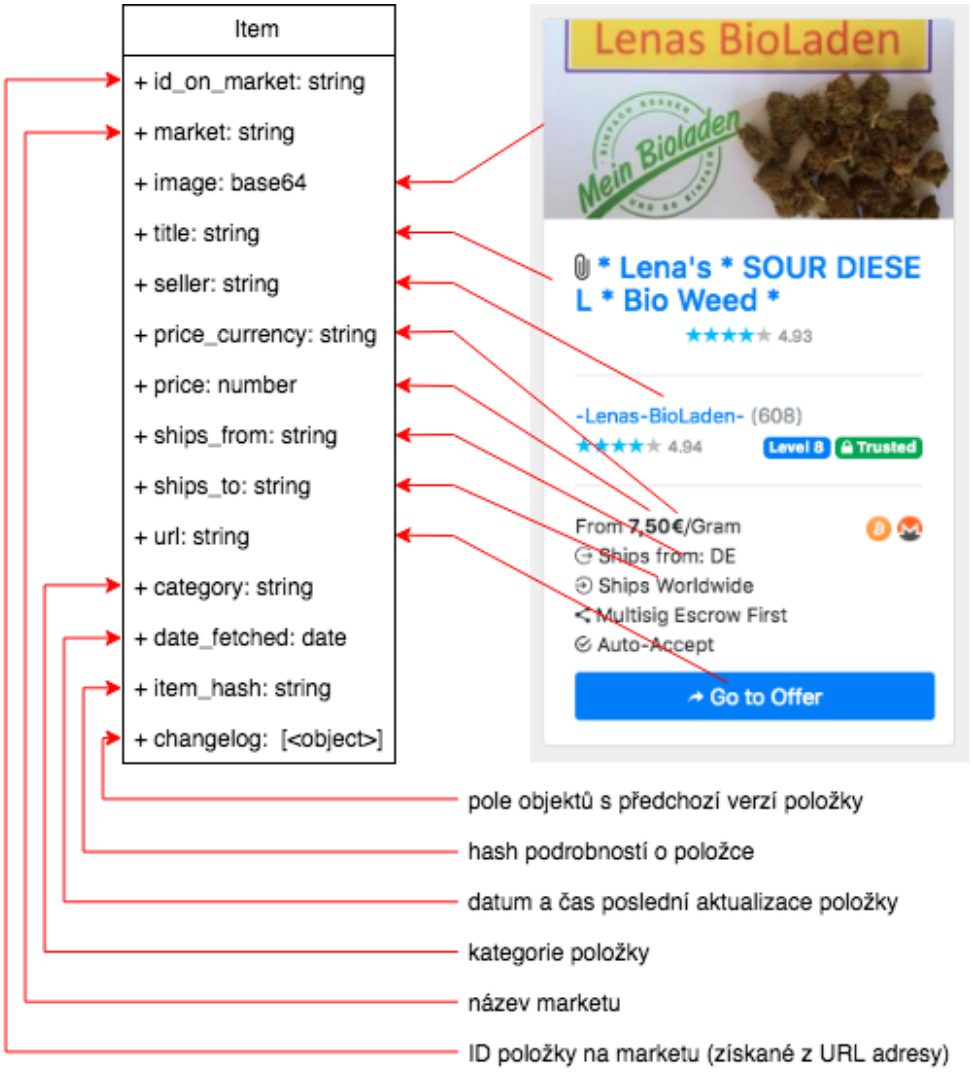
Pro správnou funkci musí být crawler před svým spuštěním nakonfigurován pro konkrétní market. Tato konfigurace bude načtěna z konfiguračního souboru, který bude obsahovat zmíněný seed s adresou počáteční stránky a pravidla pro získávání dalších odkazů.

Pro crawlování je buď možné vytvořit vlastní software nebo využít již existujících řešení, které často obsahují i modul pro scrapování. Příklady takových řešení jsou **Apache Nutch**, **Pavuk**, **Scrapy** nebo **wget**.

2.13.4 Scraper

Po obdržení *HTML dokumentu* od **crawleru** začne **scraper** či **scrapovací modul** s jeho zpracováním, tedy s extrakcí informací o jednotlivých položkách zboží. Protože je *HTML dokument* na straně serveru marketu dynamicky generován, bývá zpravidla rozumně strukturovaný. scraperu tedy stačí extrahovat data z *DOM modelu* bez složitého *parsování*. Cestu ke správným elementům DOM modelu získá podobně jako v případě crawleru z konfiguračního souboru. Tato konfigurace bude muset být opět přizpůsobena na míru pro konkrétní market.

Extrahovaná data scraper následně porovná s kolekcí již nasbíraných dat v **databázi**. V té je každá položka jednoznačně identifikována názvem marketu, odkud pochází, a svým identifikátorem, pod kterým je vedena na marketu (identifikátor je součástí adresy položky). Schéma kolekce dat v databázi s jejich mapováním na data získávaná z detailů položek je znázorněno na obrázku 2.6.

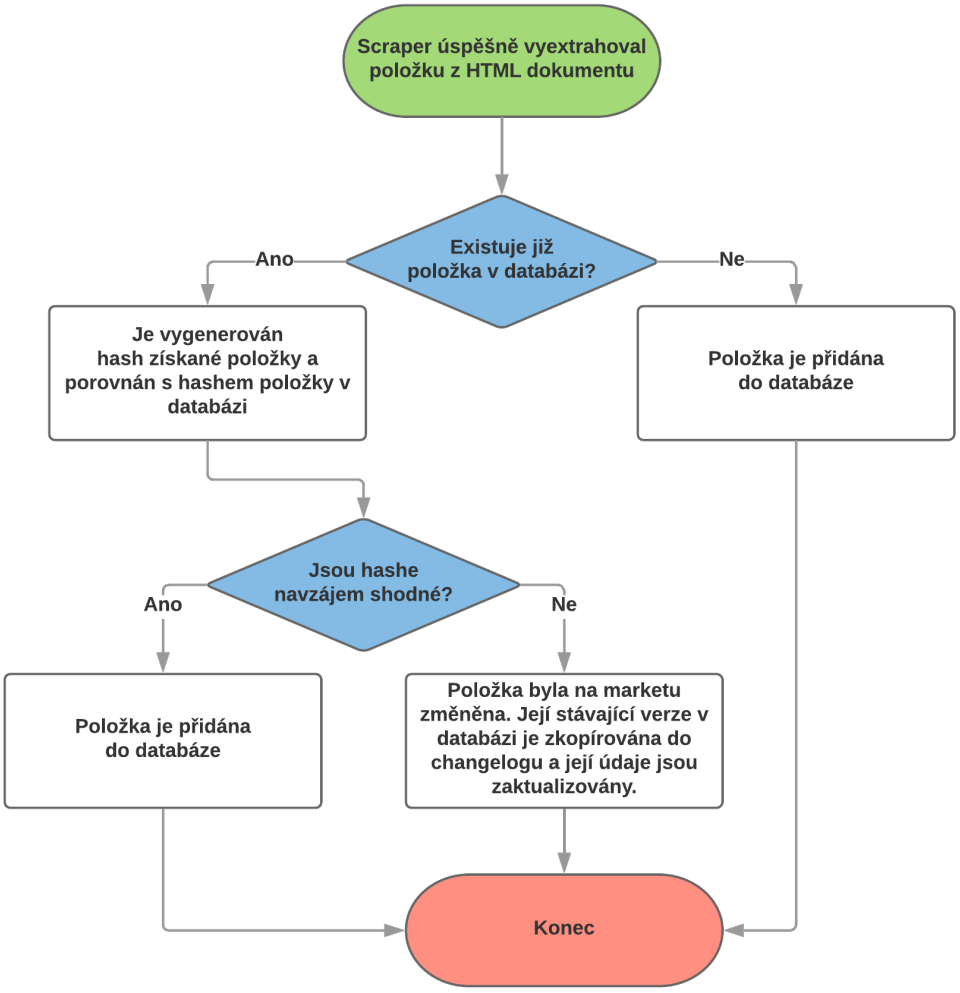


Obrázek 2.6: Mapování dat z miniatury položky na marketu (konkrétně Wall-street Marketu) na kolekci dat v databázi [8]

Pokud nejsou uvažovány různé chyby při zpracování položek, může během něj dojít celkem ke třem různým situacím. Pokud se právě zpracovávaná položka v databázi ještě nenachází, scraper ji do ní přidá. Pokud se položka v databázi již nachází, porovná se její uložený *hash*[45] s hashem aktuální položky. V případě, že hashe sedí, nebyla položka na webu od poslední

[45]Hashovací funkce (funkce generující hashe) mají v informatice mnoho uplatnění. Mezi jejich hlavní vlastnosti patří, že jednoznačně a jednosměrně generují z libovolného množství vstupních dat krátký řetězec, který se i při nepatrné změně vstupních dat významně změní.

návštěvy změněna a v databázi se tak nachází její aktuální verze. V tomto případě scraper pouze aktualizuje časovou značku poslední aktualizace v databázi. Pokud hashe nesedí, znamená to, že položka na webu byla oproti té uložené v databázi změna. V tomto případě se data o položce v databázi zkopírují do jejího *changelogu*[46] a stávající data se nahradí jejich novou verzí. Rozhodovací proces při ukládání dat je znázorněn ve *workflow diagramu* na obrázku 2.7. Všechny zápisy do databáze se pro rychlejší zpracování provádějí po dávkách, nikoliv právě v okamžik, kdy změna nastane.



Obrázek 2.7: Workflow diagram scraperu při ukládání položky zboží do databáze

[46] Changelog je záznam o změnách, které se udály například v systému nebo dokumentu.

Stejně jako v případě crawleru je možné i **scraper** vyvinout jako vlastní řešení, například za pomocí knihoven pro procházení *DOM*, jako jsou **BeautifulSoap** pro *Python* nebo **jsoup** pro *Javu*.

2.13.5 Datové úložiště

Jako datové úložiště bude použita **NoSQL databáze**. Hlavními důvody pro její zvolení oproti *relační databázi* jsou flexibilní datový model, snadná implementace a škálovatelnost a možnost notifikací ostatních komponent při změnách.

Jak už bylo naznačeno v popisu předešlých komponent, databáze bude sloužit jako úložiště celkem 3 kolekcí dat:

1. **přihlášené cookies** ukládané *přihlašovacím modulem* a využívané crawlerem k procházení marketu
2. **nacrawlované odkazy** na jednotlivé stránky s výpisem zboží získané crawlerem a využívané scraperem
3. **nascrapovaná data** zboží získaná scraperem

Protože se na získaná data bude dotazovat **backend**[47] webové aplikace, je třeba, aby byla rychle přístupná a zbytečně se neprodlužovala doba, jakou bude muset uživatel čekat na jejich zobrazení. Pro zrychlení přístupu k datům je v návrhu zahrnut **index**, který bude data držet v operační paměti.

Dalším kanálem, kterým přijde uživatel s daty do styku, jsou notifikace. Zde bude využito funkce databáze zaslat změny jiné komponentě. Díky tomu se nebude nutné pro zjištění změn databáze opakovaně dotazovat v předem stanovených intervalech.

Vhodnými kandidáty pro databázi jsou **MongoDB** nebo **RethinkDB**, jako index pak **Elasticsearch** či **Apache Solr**.

2.13.6 Notifikační podsystém

O vytváření notifikací o změnách se budou starat dvě komponenty: **fronta notifikací**, která bude přijímat změny od **databáze**, a **notifikační nástroj**, který bude frontu zpracovávat a po dávkách uživatele informovat o změnách zvoleným způsobem. Základní formou informování uživatele v rámci systému budou *push notifikace*[48] do **webové aplikace**. Dále bude možné z notifikačního nástroje zasílat reporty elektronickou poštou či na něj napojit externí služby jako jsou **Slack** nebo **Twitter**.

[47] Jako backend se označuje část aplikace, která obsahuje programovou logiku a přistupuje k datům. Výsledky pro zobrazení uživateli pak předává frontendu.

[48] Push notifikace je zpráva nebo upozornění, které je uživateli zasláno v moment, kdy situace nastala, bez nutnosti dotazovat se serveru [49].

Výhodou rozdělení podsystému na frontu a samotný notifikační nástroj je možnost notifikace agregovat do dávek, kdy například dojde ke zpracování fronty každý týden. Další výhodou je, že v případě výpadku nástroje nezpracované notifikace zůstanou ve frontě a nedojde k jejich ztrátě.

Mezi nejpoužívanější software zajišťující funkci fronty patří **RabbitMQ** a **Apache Kafka**.

2.13.7 Webová aplikace

Hlavní způsob prohlížení dat uživatelem je prostřednictvím webové aplikace. Samotná aplikace bude rozdělená na dvě části: **backend**, který se bude starat o aplikační logiku a získávání dat, a **frontend**[49], který zajistí jejich správné zobrazení uživateli.

2.13.7.1 Backend

Backend zpracovává požadavky od frontendu a vrací mu data, která mají být zobrazeny uživateli. Další úlohou, kterou backend řeší, je příjem notifikací od **notifikačního nástroje**. Tyto notifikace rovněž předává frontendu pro jejich zobrazení.

2.13.7.2 Frontend

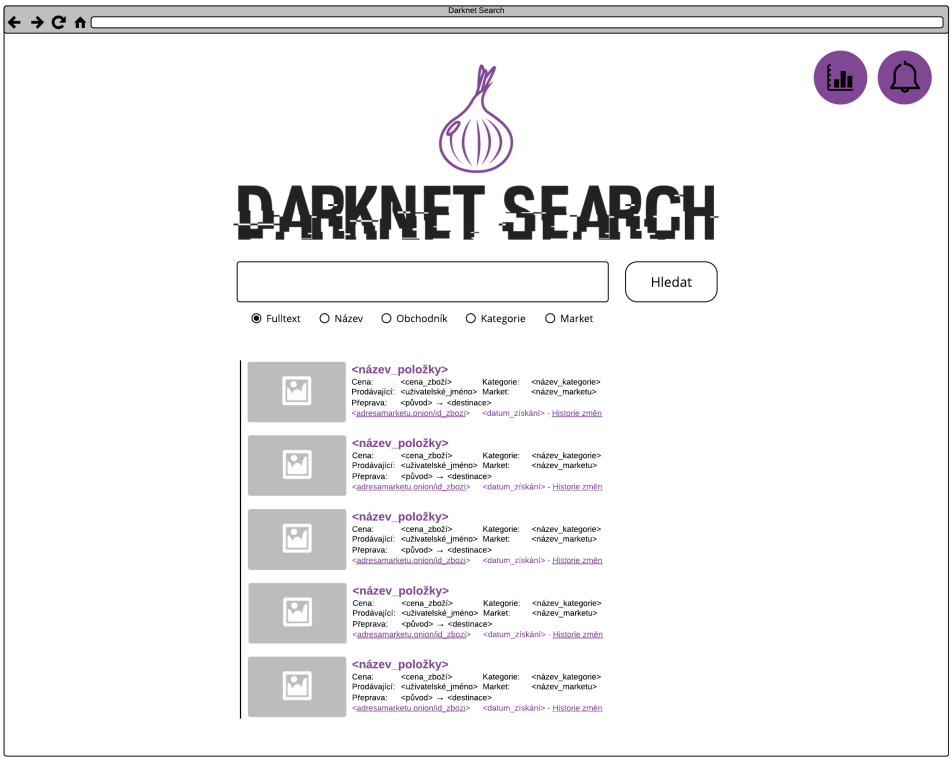
Frontend je *prezentační* vrstvou, pomocí které uživatel zadává požadavky na zobrazení dat. Data obdržená od backendu pak frontend uživateli vhodným způsobem zobrazuje ve webovém prohlížeči. Celkem se skládá ze 3 modulů, jsou jimi:

- vyhledávací modul,
- vizualizační modul,
- notifikační modul.

Navržené uživatelské rozhraní je inspirováno u tradičních vyhledávačů typu **Google** nebo **Seznam** a jeho wireframe[50] je vyobrazen na obrázku 2.8.

[49] Jako frontend se označuje část systému nebo aplikace, prostřednictví které uživatel interaguje se systémem a která mu z něj tak zpřístupňuje data. Jeden z hlavních požadavků na frontend je tak uživatelská přívětivost.

[50] Wireframe je schématickou kostrou uživatelského rozhraní, slouží k jeho návrhu a znázorňuje rozmístění jednotlivých prvků na obrazovce.



Obrázek 2.8: Wireframe navrženého uživatelského rozhraní

Kapitola 3

Implementace

Tato kapitola popisuje způsob a průběh implementace prototypu systému, který byl vytvořen pro demonstraci jeho návrhu. Prototyp je schopný se připojit k darknetovému marketu, přihlásit se a získat z něj definovaný obsah, který uloží do databáze. Systém byl implementován a nasazen lokálně na platformě MacOS a sběr dat byl testován na *Dream Marketu*.

3.1 Použité technologie a nástroje

Při výběru nástrojů pro implementaci prototypu se vycházelo z provedené analýzy dostupných řešení, z informací o podobných projektech dohledaných ve volně dostupných zdrojích na internetu a v neposlední řadě také z vlastních zkušeností.

Scrapovací podsystém Protože crawler/scraper tvoří jádro systému, má jeho výběr zásadní vliv na funkčnost a případná omezení celého systému. Apache Nutch pro crawlování pro mnoho programátorů první volbou, po které sáhnou. V tomto případě se však jako nástroj pro scrapování několika předem definovaných webů jevil jako příliš komplexní.

Nakonec byl zvolen **Scrapy**, o kterém byla zmínka v sekci 2.10.5 na straně 35 a který je v komunitě také velmi oblíbený. Protože jde spíše o scrapovací framework pro jazyk *Python*, je snadno rozšiřitelný. To se hodí zejména při přihlašování na market. Jeho nevýhodou je, že nepodporuje protokol *SOCKS*.

Tor proxy Protože Scrapy pouze protokol HTTP a ne SOCKS, který Tor používá, je nutné navzájem tyto dva protokoly překládat. K tomu je použita další proxy **Polipo**. Ta je umístěna mezi Scrapy a Torem a obstarává překlad jejich komunikace.

CAPTCHA Pro řešení CAPTCHA ochrany byla vybrána služba **Anti-Captcha**, která nabízí moderní rozhraní a nízkou cenu, která se pohybuje okolo 2 haléřů za obrázek. Obrázky k řešení přijímá prostřednictvím

Datové úložiště O uložení dat se starají celkem dvě komponenty: databáze a index. Jako databáze byla zvolena **MongoDB**, která je velmi rozšířená a umí notifikovat o změnách prostřednictvím funkce **Change Streams**.

3.2 Tor proxy

Jak bylo zmíněno, pro komunikaci s marketem je kromě samotného **Toru** nutné použít ještě **Polipo**, které bude komunikaci překládat z HTTP na SOCKS a obráceně. Jeho konfigurace nebyla příliš složitá, ačkoliv tento software již není aktivně udržován. Problém dělalo pouze načtení konfiguračního souboru na platformě MacOS a při každém spuštění tak bylo nutné Polipo znovu nakonfigurovat. Na platformě Linux, kde systém bude běžet v reálném provozu, by tento problém neměl nastat.

3.3 Získání obsahu

V čem se scrapování darknetových marketů liší od sběru dat z většiny e-shopů a tržišť na běžném internetu je to, že markety na darknetu se aktivně snaží zamezit robotům jejich data sbírat. Průměrný e-shop snaží robotům crawling usnadnit, aby v jeho zboží bylo možné vyhledávat i mimo jeho web, což mu přinese nové zákazníky. Typický darknetový market naopak schovává veškerý obsah za přihlašovací obrazovku s CAPTCHA ochranou.

3.3.1 Přihlášení

Aby bylo možné získat stránky s jednotlivými položkami prodávaného zboží, musí se uživatel nejdříve registrovat a přihlásit. Protože indikací přihlášeného uživatele je pro webový server marketu cookie v hlavičce požadavku, je úkolem **přihlašovacího modulu** tuto cookie získat a předat ji crawleru, kterému pak díky ní budou stránky se zbožím přístupné.

V návrhu popisujícím architekturu systému z obecnějšího hlediska se o její získání stará přihlašovací modul, který cookie předává crawleru prostřednictvím databáze. Protože ale byl jako crawler zvolen Scrapy, bylo možné přihlašování integrovat přímo do něj a Scrapy tak cookie získá přímo voláním přihlašovací funkce na začátku crawlování.

Automatizace přihlašování na Dream Market má několik nástrah. Jednou z nich je vyřešení CAPTCHA. Ačkoliv je použití **Anti-captcha** poměrně snadné, stává se poměrně často, že bývá vyřešena nesprávně. V některých případech je toto detekovatelné ještě před jejím vyplněním do přihlašovacího

formuláře, a to její špatnou délkou. Ta je na Dream Marketu při správném vyřešení přesně 4 znaky. Protože jsou v obrázku přítomné i znaky, které nemají být opsány, stává se, že se i ty vrátí v odpovědi od Anti-captchy jako řešení. V tom případě je stejný obrázek posílán službě znovu.

Další překážkou, kterou je nutné pro přihlášení na Dream Market překonat, jsou skrytá formulářová pole v přihlašovacím formuláři. V tomto případě se jedná o několikanásobně duplikovaná pole pro vyplnění uživatelského jména a hesla. Z těchto polí je opravdovému uživateli skutečně viditelná pouze jedna dvojice, do které musí být vyplněny přihlašovací údaje. Ostatní jsou předvyplněny dlouhými vygenerovanými řetězci se kterými pro úspěšné přihlášení nesmí být manipulováno a slouží pouze pro „zmatení“ robota - jde o tzv. honeypoty[51]. Všechna tato předvyplněná pole musí být spolu s přihlašovacími údaji odeslána serveru.

Ze zmíněného příkladu je jasně patrné, že přihlašování musí být, minimálně v případě Dream Marketu, připraveno každému marketu na míru. Protože se ale většina obchodu na darknetu soustřeďuje na několika největších marketech, neměla by mít příprava samostatného modulu pro každý market zvlášť zásadní dopad na finanční náročnost projektu.

3.3.2 Crawlování a scrapování

Po úspěšném přihlášení roli přebírá crawler a scraper. V případě **Scrapy** jsou obě tyto funkce velice úzce provázány.

Crawler začíná pracovat s adresami, které má v seedu. Z těchto adres sbírá data definovaná na obrázku 2.6 na straně 43. Data extrahovaná ze stránky opatřuje jejich hashem a časovým razítkem a ukládá je do databáze v závislosti na jejich předchozí existenci v databázi podle obrázku 2.7 na straně 44.

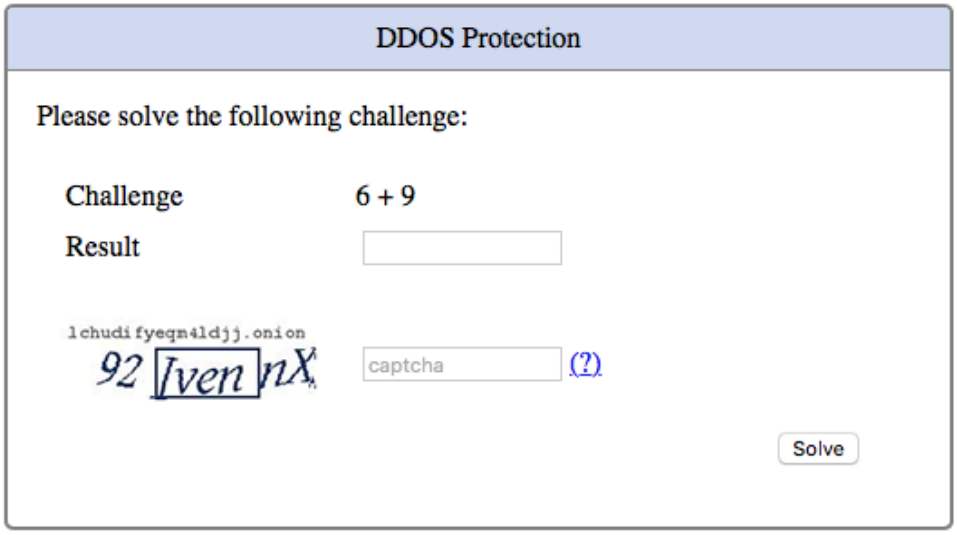
Po zpracování dat na stránce získává crawler adresy na další stránky s daty, které ukládá do databáze, pokud v ní již nejsou. Z databáze pak načítá adresy, které nebyly dlouho scrapovány a celý proces scrapování opakuje.

Problémem, který se čas od času vyskytl, byla blokace marketem pro podezření z *DDoS útoku*[52]. V tomto případě market zobrazil stránku (obrázek 3.1), na které pro pokračování vyžadoval vyřešení snadného matematického příkladu a vyplnění CAPTCHA.

Toto je řešeno opětovným přihlášením, čímž je vygenerována nová cookie. V ideálním případě je vhodné použít i nový uživatelský účet.

[51] Honeypot je prvek stránky, který se tváří jako že je určený k uživatelskému vstupu. Ve skutečnosti uživateli ale není vyditelný a jedná se pouze o past k identifikaci robotů a jejich blokování.

[52] DoS, celým názvem denial-of-service, je druh útoku na webové služby, při kterém se útočník snaží službu dočasně vyřadit z provozu posíláním velkého množství požadavků dokud nedojde k jejímu zahlcení, čímž se stane nedostupná i pro ostatní uživatele. Distribuovaná verze tohoto útoku se označuje jako DDoS, která se od klasického DoS útoku liší tím, že požadavky přicházejí z mnoha zdrojů, často prostřednictvím botnetů.



Obrázek 3.1: Stránka zobrazená Dream Marketem jako ochrana proti DDoS útoku

Kromě nastavení proxy serveru a naprogramování crawlování a extrakce dat ze stažených stránek bylo také nutné ve Scrapy vypnout respektování souboru *robots.txt*, který Dream Market obsahuje. Tento soubor totiž obsahuje direktivy pro zákaz jakéhokoliv pohybu robotů po jeho stránkách.

3.4 Databáze

Jako úložiště byla použita databáze **MongoDB**. Pro její instalaci byl použit virtualizační nástroj **Docker**, který umožňuje instalaci software jako *kontejnery* nezávisle na použitém operačním systému.

3.5 Průběh a výsledky testování

Prototyp systému byl testován na stahování dat z **Dream Marketu**. V testovacím běhu bylo crawlování omezeno na jednu z největších kategorií marketu – **Digital goods**. V okamžiku zahájení stahování čítala tato kategorie dle údajů na stránkách 49 263 položek. Pro účely testování byl na marketu vytvořen jeden uživatelský účet, na který se systém pomocí přihlašovacího modulu přihlašoval a který crawler využíval k pohybu v rámci marketu.

Běh systému trval přibližně **45 minut** a podařilo se mu během něj nasbírat **46 657** položek. Rychlost sběru dat tedy dosáhla hodnoty přibližně **62 209** položek za hodinu. V průběhu získávání dat byl systém blokován *DDoS* ochra-

nou přibližně každých 30 požadavků, tedy každých 960 položek (32 položek na stránku).

V případě, že lze údajům na webu Dream Marketu důvěřovat a uvádí tedy počet položek zboží, který se na něm skutečně nachází, dosáhl systém úspěšnosti **94,71 %**.

Závěr

Téma práce Tématem práce a jejím hlavním úkolem bylo seznámit se s problematikou marketů na darknetu, provést na toto téma rešerši a na jejím základě navrhnout systém, který bude tyto markety monitorovat. Dále měla být provedena kalkulace náročnosti vývoje takového systému a vytvořen jeho prototyp. Podnětem pro práci byla poptávka Policie ČR po tomto systému.

Rešeršní část Byla definována a vysvětlena terminologie týkající se darknetu a pojmů s ním spojených. Dále byla popsána technologická stránka fungování sítě Tor a obsah, který se na této síti nachází.

Analýza a návrh systému Byl navržen systém pro sběr dat o zboží prodávaném na darknetových marketech. Systém poskytuje svým uživatelům vhodné rozhraní pro prohledávání získaných dat a notifikuje je o nových změnách v nabízeném zboží. Návrh jeho architektury se nachází v sekci 2.12 na straně 39.

Dále byla vypracována analýza projektu pro vývoj tohoto systému pro sekci kybernetické kriminality Policie ČR. Analýza kromě jiného zahrnuje kalkulaci nákladů na vývoj i provoz systému, jeho přínosy a metriky pro měření úspěšnosti projektu.

Získané poznatky Během návrhu i implementace prototypu systému bylo zjištěno, že úplná univerzálnost systému pro jeho použití na různých marketech není dosažitelná a některé jeho součásti je nutné vyvinout „na míru“ konkrétním marketům. Je tomu tak hlavně proto, že markety používají opatření pro znesnadnění přístupu robotů na jejich stránky, jako jsou používání *CAPTCHA*, obfuskace přihlašování a detekce vysokého počtu přístupů.

Přínosy práce Darknetové markety a získávání jejich obsahu nejsou příliš prozkoumaným tématem. Částečně zřejmě proto, že první z nich byl otevřen

teprve v roce 2011. Ačkoliv bezpečnostní složky mnoha států své úsilí sběru informací z jistě věnují, veřejných zdrojů tohoto druhu je jen hrstka.

Práce v tomto směru přináší ucelené shrnutí informací týkajících se fungování darknetových marketů a nabízí způsob, jakým monitorovat jejich dění.

Práce byla konzultována se zástupci sekce kybernetické kriminality PČR, kterými byla přijata jako přínosná a reflektující realitu.

Doporučení pro další rozvoj V rámci dalšího rozvoje by bylo systém vhodné vybavit mechanismem, který by stěžoval detekci jeho činnosti cílenými markety. Vhodnými způsoby jak systém maskovat by mohla být například rotace různých uživatelských účtů, pravidelné změny používaných uzlů Toru či simulace uživatelského chování na webu.

Další oblastí, ve které má systém také prostor pro rozvoj, je vytěžování informací ze získaných dat. Toto úzce souvisí s dalším druhem dat, o jejichž sběr by měla policie také zájem, kterým je obsah darknetových diskuzních fór. Spojení dat z těchto fór s účty na marketech by podle ní přineslo velkou přidanou hodnotu.

Literatura

[1] Bergman, M. K.: White paper: the deep web: surfacing hidden value. *Journal of electronic publishing*, ročník 7, č. 1, 2001. Dostupné z: http://dx.doi.org/10.3998/3336451.0007.104

[2] The Tor Project: *Tor Logo [online]*. 2011, [cit. 2018-03-21]. Dostupné z: https://media.torproject.org/image/official-images/2011-tor-logo-flat.svg

[3] Wikimedia Commons: *SVG Diagram of the „Onion Routing“ Principle [online]*. 2008, [cit. 2018-03-17]. Dostupné z: https://commons.wikimedia.org/wiki/File:Onion_diagram.svg

[4] Goodin, D.: *Scientists detect „spoiled onions“ trying to sabotage Tor privacy network [online]*. Ars Technica, 2014, [cit. 2018-03-22]. Dostupné z: https://arstechnica.com/information-technology/2014/01/scientists-detect-spoiled-onions-trying-to-sabotage-tor-privacy-network

[5] The Tor Project: *Tor: Onion Service Protocol [online]*. 2018, [cit. 2018-03-26]. Dostupné z: https://www.torproject.org/docs/onion-services.html.en

[6] Gartner: *Magic Quadrant for Cloud Infrastructure as a Service, Worldwide [online]*. 2017, [cit. 2018-05-03]. Dostupné z: https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519

[7] Dream Market: *Dream Market [online]*. 2018, [cit. 2018-04-01]. Dostupné z: http://lchudifyeqm4ldjj.onion

[8] Wallstreet Market: *Wallstreet Market [online]*. 2018, [cit. 2018-04-01]. Dostupné z: http://wallstyizjhkrvmj.onion

[9] Greenberg, A.: *Hacker Lexicon: What Is the Dark Web? [online]*. Wired Magazine, 2014, [cit. 2018-03-03]. Dostupné z: https://www.wired.com/2014/11/hacker-lexicon-whats-dark-web

[10] WhoIsHostingThis.com: *Tor & The Deep Web: The WhoIsHostingThis.com Guide [online]*. 2014, [cit. 2018-03-03]. Dostupné z: https://www.whoishostingthis.com/blog/2017/03/07/tor-deep-web

[11] WhoIsHostingThis: *Everything you need to know about the Deep Web in one simple infographic [online]*. Business Insider, 2015, [cit. 2018-03-03]. Dostupné z: http://www.businessinsider.com/everything-you-need-to-know-about-the-deep-web-in-one-simple-infographic-2015-2

[12] Epstein, Z.: *How to find the Invisible Internet [online]*. Boy Genius Report, 2014, [cit. 2018-03-03]. Dostupné z: http://bgr.com/2014/01/20/how-to-access-tor-silk-road-deep-web

[13] ICANN: *What Does ICANN Do? [online]*. 2012, [cit. 2018-03-03]. Dostupné z: https://www.icann.org/resources/pages/what-2012-02-25-en

[14] Piscitello, D.: *The Dark Web: The Land of Hidden Services [online]*. ICANN, 2017, [cit. 2018-03-03]. Dostupné z: https://www.icann.org/news/blog/the-dark-web-the-land-of-hidden-services

[15] Černý, J.: *Tři druhy webu: povrchový, hluboký a temný [online]*. Informační gramotnost, 2017, [cit. 2018-03-04]. Dostupné z: https://www.informacnigramotnost.cz/tri-druhy-webu

[16] UC San Diego Library: *Web Sources [online]*. 2018, [cit. 2018-03-06]. Dostupné z: https://ucsd.libguides.com/mmw122/websources

[17] European Police Office: *Internet Organised Crime Threat Assessment*. 2016, doi:10.2813/275589, [cit. 2018-03-14]. Dostupné z: https://www.europol.europa.eu/iocta/2016/resources/iocta-2016.pdf

[18] The Tor Project: *Tor Metrics - Users [online]*. 2018, [cit. 2018-03-15]. Dostupné z: https://metrics.torproject.org/userstats-relay-country.html?start=2017-12-15&end=2018-03-15&country=all&events=off

[19] The Tor Project: *Users of Tor [online]*. 2018, [cit. 2018-03-23]. Dostupné z: https://www.torproject.org/about/torusers.html

[20] The Tor Project: *Tor FAQ [online]*. 2018, [cit. 2018-03-15]. Dostupné z: https://www.torproject.org/docs/faq.html

[21] Syverson, P.: *Onion Routing: History [online]*. 2005, [cit. 2018-03-20]. Dostupné z: https://www.onion-router.net/History.html

[22] The Tor Project: *Tor Donor FAQ [online]*. 2018, [cit. 2018-03-20]. Dostupné z: https://donate.torproject.org/donor-faq

[23] The Tor Project: *Tor: Bridges [online]*. 2018, [cit. 2018-03-24]. Dostupné z: https://www.torproject.org/docs/bridges.html.en

[24] nickm: *Tor 0.3.2.1-alpha is released, with support for next-gen onion services and KIST scheduler [online]*. The Tor Project, 2017, [cit. 2018-03-25]. Dostupné z: https://blog.torproject.org/tor-0321-alpha-released-support-next-gen-onion-services-and-kist-scheduler

[25] Appelbaum, J.; Muffett, A.: *The „.onion“ Special-Use Domain Name [online]*. 2015, [cit. 2018-03-25]. Dostupné z: https://tools.ietf.org/html/rfc7686

[26] WikiLeaks: *Tor [online]*. 2018, [cit. 2018-03-25]. Dostupné z: https://www.wikileaks.org/wiki/WikiLeaks:Tor

[27] Muffett, A.: *Making Connections to Facebook more Secure [online]*. 2014, [cit. 2018-03-25]. Dostupné z: https://www.facebook.com/notes/protect-the-graph/making-connections-to-facebook-more-secure/1526085754298237/

[28] Sandvik, R.: *The New York Times is Now Available as a Tor Onion Service [online]*. 2017, [cit. 2018-03-25]. Dostupné z: https://open.nytimes.com/https-open-nytimes-com-the-new-york-times-as-a-tor-onion-service-e0d0b67b7482

[29] Didsoft: *Socks Proxy VS HTTP Proxy [online]*. 2012, [cit. 2018-03-27]. Dostupné z: https://www.my-proxy.com/blog/socks-proxy-http-proxy

[30] Panopticlick: *About Panopticlick [online]*. 2018, [cit. 2018-03-27]. Dostupné z: https://panopticlick.eff.org/about#browser-fingerprinting

[31] The Tor Project: *Stem Docs - Frequently Asked Questions [online]*. 2018, [cit. 2018-03-28]. Dostupné z: https://stem.torproject.org/faq.html

[32] Subgraph: *Orchid [online]*. 2018, [cit. 2018-03-28]. Dostupné z: https://subgraph.com/orchid/index.en.html

[33] The Tor Project: *Torsocks [online]*. 2018, [cit. 2018-03-28]. Dostupné z: https://trac.torproject.org/projects/tor/wiki/doc/torsocks

[34] Hermes Center: *Tor2web: Browse the Tor Onion Services [online]*. 2018, [cit. 2018-03-28]. Dostupné z: https://www.tor2web.org

[35] The Tor Project: *Software & Services [online]*. 2018, [cit. 2018-03-29]. Dostupné z: https://www.torproject.org/projects/projects.html.en

[36] Branwen, G.: *Dark Net Market archives, 2011-2015 [online]*. 2015, [cit. 2018-04-11]. Dostupné z: https://www.gwern.net/DNM-archives

[37] Lnal: *List of onion sites seized/deanonymised [online]*. Reddit, 2018, [cit. 2018-04-01]. Dostupné z: https://redd.it/7zdius

[38] Pro Web Scraping: *Web Scraping Vs Web Crawling [online]*. 2018, [cit. 2018-04-08]. Dostupné z: http://prowebscraping.com/web-scraping-vs-web-crawling/

[39] W3schools: *The HTML DOM Document Object [online]*. 2018, [cit. 2018-04-03]. Dostupné z: https://www.w3schools.com/jsref/dom_obj_document.asp

[40] Webhose.io: *Crawled Web Data at Scale — Open and Dark Web API [online]*. 2018, [cit. 2018-04-08]. Dostupné z: http://www.webhose.io

[41] Octoparse: *Web Scraping Tool & Free Web Crawlers for Data Extraction [online]*. 2018, [cit. 2018-04-09]. Dostupné z: https://www.octoparse.com

[42] Dexi.io: *Dexi - web data extraction tool for professionals [online]*. 2018, [cit. 2018-04-09]. Dostupné z: https://dexi.io/product

[43] Scrapinghub: *Scrapy Cloud [online]*. 2018, [cit. 2018-04-10]. Dostupné z: https://scrapinghub.com/scrapy-cloud

[44] Web Scraper: *Web Scraper [online]*. 2018, [cit. 2018-04-10]. Dostupné z: http://webscraper.io/service

[45] Kofler, R.: *Scraping the Web with Nutch for Elasticsearch [online]*. 2015, [cit. 2018-04-10]. Dostupné z: https://qbox.io/blog/scraping-the-web-with-nutch-for-elasticsearch

[46] LewisJohnMcgibbney: *Nutch Wiki - FrontPage [online]*. Apache Software Foundation, 2018, [cit. 2018-04-10]. Dostupné z: https://wiki.apache.org/nutch/FrontPage

[47] Brody, H.: *I Don't Need No Stinking API: Web Scraping For Fun and Profit [online]*. 2018, [cit. 2018-04-10]. Dostupné z: https://blog.hartleybrody.com/web-scraping/

[48] Free Software Foundation, Inc.: *GNU Wget 1.18 Manual [online]*. 2015, [cit. 2018-04-11]. Dostupné z: `https://www.gnu.org/software/wget/manual/wget.html`

[49] PC Magazine: *PC Magazine Encyclopedia [online]*. 2018, [cit. 2018-04-27]. Dostupné z: `https://www.pcmag.com/encyclopedia/term/63334/push-notification`

Příloha **A**

Seznam použitých zkratek

PČR Policie České republiky

IP Internet Protocol

DNS Domain Name Service

WWW World Wide Web

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart

Tor The Onion Routing

Tails The Amnesic Incognito Live System

VPN Virtual Private Network

SSL Secure Sockets Layer

PGP Pretty Good Privacy

MD man-day

DOM Document Object Model

HTML Hypertext Markup Language

OCR Optical Character Recognition

TCO Total Cost of Ownership

DDoS distributed-denial-of-service

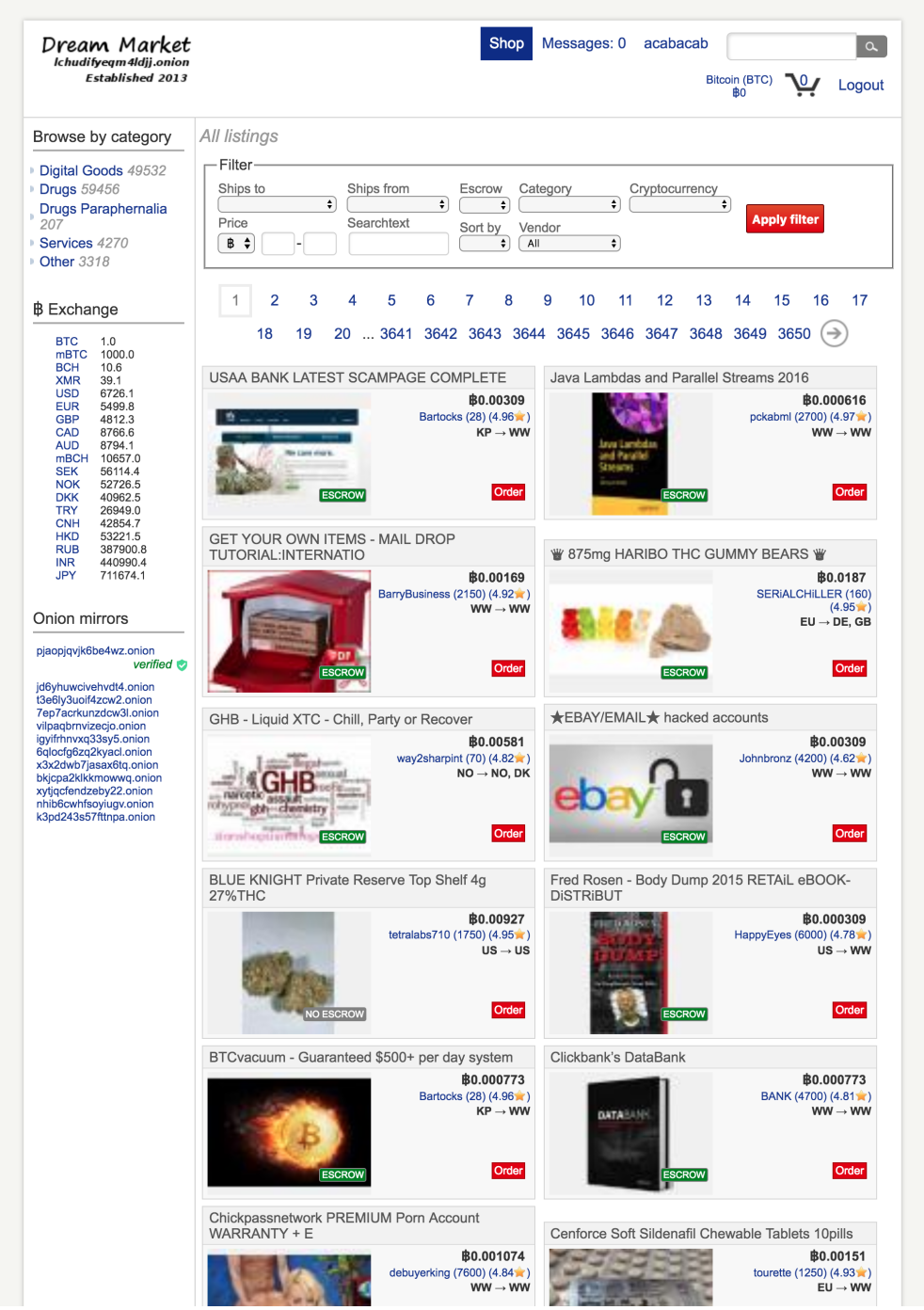
Příloha B

Obsah přiloženého CD

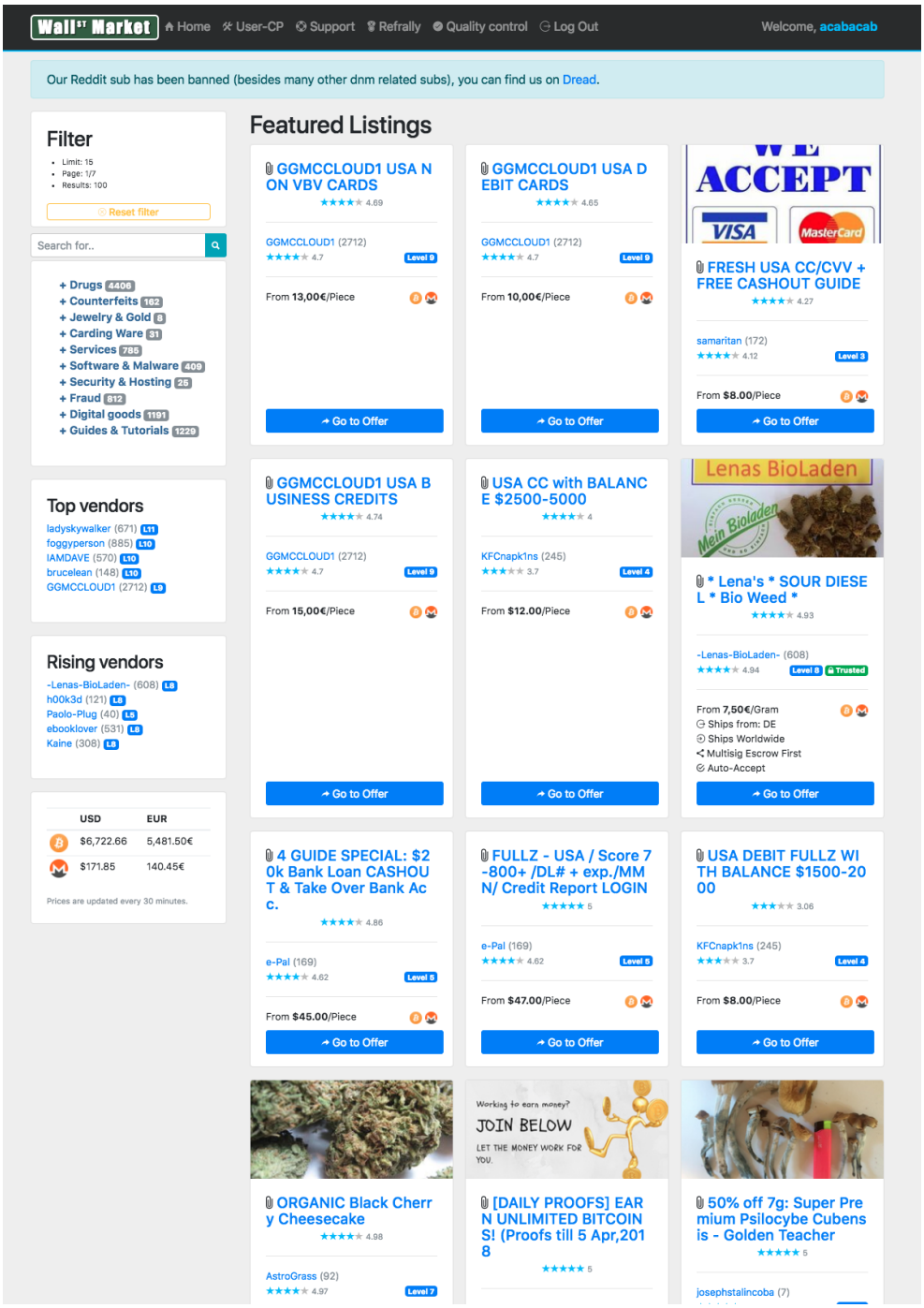
- `readme.txt` ..... stručný popis obsahu CD
- `data` ..... adresář s daty staženými z marketů
- `src`
  - `source.zip` ..... zdrojové kódy implementace
  - `thesis.tex` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF
  - `thesis.ps` ..... text práce ve formátu PS

Příloha C

Obrazová příloha



Obrázek C.1: Náhled obrazovky se zbožím na Dream Marketu [7]



Obrázek C.2: Náhled obrazovky se zbožím na Wallstreet Marketu [8]