Blekinge Tekniska Högskola

Kandidatarbete

Inblick i fenomenet webbskrapning

Av: Lars Andersson

Email: Lars_andersson4@hotmail.com

Abstrakt

Föreliggande kandidatarbete har till syfte att undersöka fenomenet Webskrapning.

Webbskrapnings-program (också kända som Web Wanderers, Crawlers, Spiders eller skrapare) är program som söker igenom webben automatiskt för att extrahera information från webbsidor.

Ett exempel på web skrapning är när ett företag samlar in data om prissättningar på en vara eller en tjänst och sen använder informationen för att producera billigare erbjudanden. Detta ger företaget en fördel så att de kan fokusera mera på att marknadsföra sin sida/tjänster. Utöver detta så blir de utsatta företagens servrar också hårt belastade med trafik (skrapning) från ”icke kunder”.

Efter att ha genomsökt både akademiska och allmänna källor via informationsinsamling, av denna information så dras slutsatsen att man inte fullt ut kan hindra skrapning av hemsidor. Detta på samma sätt som man inte fullt ut kan hindra någon IT-attack, det finns inga 100 % vattentäta system. Av utfallet ifrån informationssökningen var det bara ett akademiskt arbete, av de hundra, som genomsöktes som hade inriktat sig på att förhindra skrapningsbotar.

Nyckelord/Keywords: Scraping, Crawler, Spider, Skrapning, Spider trap, Sticky honeypot, Robots.txt, Bot, Webbscraping, Scraping methods.

Innehållsförteckning

1 Inledning

Syftet med detta examensarbete är att via informationsinsamling sammanställa relevant information om fenomenet webskrapning.

Detta examensarbete har initierats av Sentor AB, ett företag som arbetar med att ta hand om andra företags IT-säkerhetsfrågor och utföra säkerhetsanalyser.

Sentor AB har haft problem med att hitta tillräckligt med relevant information om webbskrapning och vilka sätt det finns att skydda sig mot fenomenet.

Webbskrapning är programvaror som automatiskt söker igenom hela webben och samlar in data som kan nås via nätet. Information som hämtas in kan vara allt från prisuppgifter på varor till företagshemligheter.

Sentor AB anger två huvudproblem; det ena är att man genom webbskrapning kan komma åt icke offentlig data, det andra är att webbskrapning kan orsaka överbelastning på kunders webbservrar.

Sentor AB:s önskemål var därför att examensarbetet skulle ta reda på, vilken information det fanns om webbskrapning på nätet och vilka akademiska artiklar som behandlar ämnet.

Ett tredje uttalat intresse var att få en inblick i vad lagstiftningen säger om webbskrapning.

2 Syfte och Mål

Syftet med examensarbete är att sammanställa information om webbskrapning. Samt att ge förslag på ett experiment som kan visa hur mycket datatrafik som webbskrapning alstrar.

2.1 Forskningsfrågor

Dessa frågeställningar kommer att besvaras i examensarbetet:

F1: Vad är Webbskrapning?

F2: Vilka tekniker används vid Webbskrapning?

F3: Är Webbskrapning ett problem?

F4: Hur kan man förhindra webskrapning?

F5: Hur kan man sätta upp en pilotstudie som undersöker trafikeffekt av webbskrapning?

2.2 Avgränsningar

Informationssökningen i arbetet avgränsas till sökningar på följande sökmotorer: www.google.se, www.scholar.google.se och summon@bth.se. Pilotstudien i fråga F5 kommer inte kunna testas i detta examensarbete på grund av att det ligger utanför arbetets tidsramar. Önskad analys av rättsläget ligger utanför ramen för utbildningen och kommer därför att endast översiktligt beröras i diskussionsdelen.

3 Metod

För att undersöka och besvara frågeställningarna i examensarbetet kommer Södertörns Högskolas informationssökningsprocess (Södertörn, 2012) (fig.1) att följas samt de databaser som beskrivs i avgränsningar (kap. 2.2). Som komplement till informationssökningen så avses genomföras intervjuer med representanter från Sentor AB. De olika metoderna i processtegen redovisas nedan:

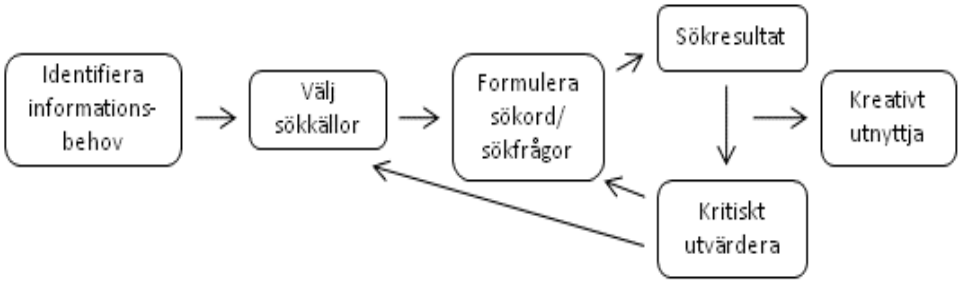

Fig.1 Illustrerar Södertörns högskolas sökprocess.

3.1 Identifiera informationsbehovet

För att identifiera informationsbehovet kommer ostrukturerade intervjuer med respondenter från Sentor AB att genomföras. När en ostrukturerad intervju genomförs ställs öppna kvalitativa frågor inom ett eller flera frågeområden. Intervjusituationen och respondentens svar genererar oftast fler följdfrågor. Utifrån den insamlade informationen kommer följande frågor beaktas:

- Vilken karaktär har arbetets informationsbehov?
- Hur mycket information behöver samlas in för att svara på frågeställningen?
- På vilken nivå ska informationen ligga? Vetenskapligt material?
- Vilket/vilka språk kommer informationen vara på?
- Vilken typ av material är relevant? Böcker, artiklar, statistik?

3.1.1 Välj sökkällor

Utifrån det identifierade informationsbehovet och de satta avgränsningarna kommer www.google.se, www.scholar.google.se och summon@bth.se att väljas som sökkällor.

3.1.2 Formulera sökord/sökfrågor

Det identifierade informationsbehovet som uttryckts i intervjuer med representanter från Sentor AB kommer att läggas till grund för examensarbetets relevanta sökord.

Det första steget i att hitta fler sökord görs via översiktliga sökningar i sökmotorn google.se. Därefter utvärderas sökorden utifrån dess relevans i förhållande till hur de kan besvara frågeställningarna (kap. 2.1). På de utvärderade sökorden görs nya sökningar i Summon och Google-Scholar för att undersöka sökordens akademiska relevans.

3.1.3 Sökresultat

Två sökstudier är planerade att genomföras, en med publika källor och en med akademiska källor. I sökstudien med publika källor är det planerat att använda synonymsökning, frassökning där orden omgärdas av citattecken (””) och filtersökning i sökmotorn google.se. Vid *Frassökning* anges sökningarna med citationstecken, t.ex. "etniska konflikter". En sådan sökning innebär att orden ska återges exakt så som du skrivit inom citationstecknen.

I den planerade akademiska sökstudien genomförs indexsökningar samt booleska operationer på de sökord som tidigare formulerats. Utöver det kommer sökhistorik att jämföras för att förfina sökresultat.

Index består oftast av nyckelord som författaren själv har tilldelat dokumentet. Dessa nyckelord försöker beskriva innehållet i dokumentet, detta gör att dokument som har liknande innehåll kan ha olika indexord beroende på vem som gör indexeringen.

3.1.4 Kritiskt utvärdera

Resultaten från de två studierna kommer att jämföras mellan varandra för att undersöka vart den mest relevanta informationen och artiklarna finns som kan besvara frågeställningarna. Informationen och artiklarna genomgår sedan en källkritisk granskning där följande utverderingskriterier används:

- **Tid** - Är informationen aktuell?
  - När uppdaterades webbsidan?
- **Beroende** - Förstahandsinformation eller beroende av annan källa?
  - Avskrivet, kopierat?
- **Äkthet** - Är webbsidan vad den utger sig för att vara?
  - Är upphovsmannen vad han/hon utger sig för att vara?
- **Tendens** - Är informationen vinklad?
  - Är det Propaganda?
  - Påverkas framställningen av upphovsmannens ideologi eller världsbild?
- **Trovärdighet** -Är webbplatsen välstrukturerad och informationsrik?
  - Hur är det med språk och stil?
  - Vilka har länkat till källan? Adress, domän?

3.1.5 Kreativt utnyttja

Under denna rubrik syntetiseras informationen från sökprocessen, något av de problem eller behov som identifieras under informationsinsamlingsprocessen kommer väljs ut att undersökas vidare. Valet av problem eller behov görs utifrån författarens egen bedömning. För att kunna sätta upp en studie och mäta en effekt kommer utgångspunkten vara att den skall kunna genomföras empiriskt kvantitativ med rimliga medel.

4 Uppsatsens struktur

Examensarbetet kommer att följa Blekinges tekniska högskolas kandidatuppsatsmall med det undantaget att då detta till stor del är en litteraturstudie så kommer rubriken Experimentdesign ändrats till Genomförande. Genomförandekapitlet kommer att beskriva hur informationssökningsprocessen gått till.

Resultatdelen kommer att svara på frågeställningarna samt beskriva hur det går att utforma en pilotstudie.

I diskussion och slutsats diskuteras metodval, arbetet som sådant, samt svaren på frågeställningarna.

Harvards referenssystem kommer att användas i examensarbetet.

4.1 Genomförande

I detta kapitel beskrivs hur intervjuer och informationssökningar genomförts i detta examensarbete enligt den process som redovisats i metodkapitlet (se kap. 3).

4.1.1 Identifiera informationsbehovet

Tre ostrukturerade intervjuer genomfördes med två anställda på Sentor AB. De båda anställda var män i 30-40 årsåldern med en lång erfarenhet av IT-säkerhet. Första intervjun genomfördes på plats på Sentor AB's Malmökontor och varade i ca 1½ timma. De två följande intervjuerna gjordes över Skype och varade i ca 45 minuter vardera.

De frågeområden som behandlades var: ämnet webbskrapning, vilka problem webbskrapning kan ställa till med och hur Sentor AB skyddar sig själv och sina klienter mot webbskrapning.

Intervjuerna med anställda på Sentor AB indikerade följande två huvudproblem; det ena är att man genom webbskrapning kan komma åt icke offentlig data, det andra är att webbskrapning kan orsaka överbelastning på kunders webbservrar. Intervjuerna gav sökorden *"webbskrapning"* och *"web scraping"*.

4.1.2 Val av sökkällor

Den idag mest spridda och använda sökmotorn är Google, därför valdes den som publik sökkälla (Wikipedia, 2013). De akademiska webbsökmotorer som fanns till förfogande var BTH´s egna artikeldatabas, summon@bth.se och google.scholar därför valdes dessa som akademiska sökkällor. Anledningen till uppdelningen mellan sökkällorna är för att kunna belysa likheter och skillnader mellan källorna. Vid sökningar av tryckta källor i BTH´s biblioteksdatabas fanns en 10 år gammal bok som dömdes vara inaktuell.

4.1.3 Formulering av sökord

Utifrån intervjuerna med anställda på Sentor AB framkom det att *"webbskrapning"* och *"web scraping"* var relevanta ord att utgå ifrån för att hitta fler sökord. Sökningar på *"web scraping"* i googles sökmotor

gav fler synonymer samt uppslag till andra för ämnet relaterade ord t.ex. *Web Wanderers, Crawlers, Spiders, web indexering, Spider trap, Sticky honeypot, Robots.txt, Bot, Webbscraping, Scraping methods, harversters.*

4.1.4 Sökresultat och kritiskgranskning av resultaten

Sökningar och granskningar skedde antingen parallellt eller i en iterativ följd. För att kunna undersöka om det fanns en diskrepans mellan publik källa och akademiska källor (se sökkällor) delades sökstudien upp i två separata studier. Sökstudien i den publika källan där användes synonymsökning, frassökning (””) och filtersökning. Både med de direkta sökorden som *web scraping* samt *webbskrapning* och de mer indirekta sökorden som *Crawlers, spider traps* etc. (Se formulering av sökord).

Urvalet av hemsidor sorterades efter när informationen var uppdaterad (datum), relevans mot frågeområdet, att det inte var en akademisk artikel och mängden relevant information. Utifrån information i de relevanta hemsidorna så gjordes utökade sökningar.

Vid sökningar i de publika källorna var det ordet *”web scraping”* som gav mest information och som bäst besvarade frågeställningarna. Därför valdes det ordet för att gå vidare med vid sökningen i de akademiska källorna. I den akademiska sökstudien genomfördes först sökningar i summon@bth.se och därefter genomsöktes scholar.google.se.

Följande kriterier har använts:

Web scraping, utan filter, utan citationstecken, från det öppna nätet.

Web scraping, utan filter, med citationstecken, från det öppna nätet.

Web scraping, med filter, utan citationstecken, från det öppna nätet.

Web scraping, med filter, med citationstecken, från det öppna nätet.

Web scraping, utan filter, utan citationstecken, från student inloggning.

Web scraping, utan filter, med citationstecken, från student inloggning.

Web scraping, med filter, utan citationstecken, från student inloggning.

Web scraping, med filter, med citationstecken, från student inloggning.

Med det *öppna nätet* menas sökningarna gjorda hemifrån med en privat dator utan att vara inloggad på BTH. *Från student inloggning* menas att författaren genomförde sökningar och var inloggad som student på BTH för

att få tillgång till de 95 artikeldatabaser som högskolan prenumererar på. Filtret som användes på summon@bth.se var ”Artiklar från vetenskapliga publikationer”, inklusive ”vetenskapligt granskade”. Utan filter och citattecken samt från det ”öppna nätet” gav sökningarna 36 264 träffar, med citattecken och filter gav de 29 träffar, se fig.4.

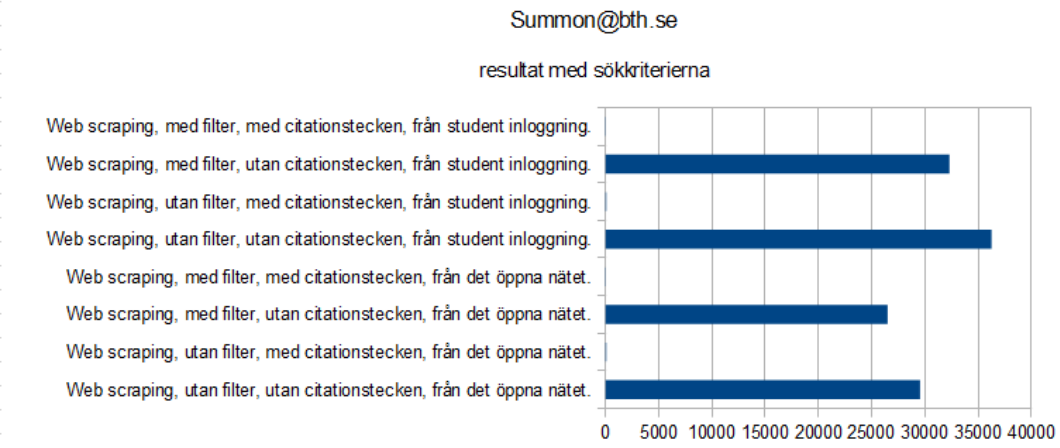

Fig 4. Visar antalet träffar i formen av ett stapeldiagram

Sökfunktionen Ctrl+f i Adobe Acrobat och sökordet ”web scraping” användes för att kontrollera vad artiklarna säger om ämnet samt i vilket sammanhang det förekom. Mer än hälften av de 29 träffarna var dock fortfarande irrelevanta artiklar om t.ex. medicinsk skrapning, arbeten om skrapning av is på betongytor eller patentansökningar som inte gav någon relevant information.

Efter summon@bth.se sökningarna gjordes sökningar i scholar.google.se, de genomfördes på samma sätt som i summon@bth.se med tillägg att även ”patent” och ”citat” filtrerades bort i sökningarna. De första 100 av totalt 491 träffar på scholar.google.se söktes igenom med Ctrl+f funktionen och med samma sökord.

Resultaten av Ctrl+f sökningarna grupperades sedan i fyra grupper, se nedan:

1. Artiklar som tillämpar "web scraping" vid datainsamling.
2. Artiklar som beskriver metoder för "web scraping"
3. Betalning krävs för publikationsåtkomst.
4. Övriga artiklar.

De arbeten som grupperades enligt kriterierna 1 och 2 valdes ut för ytterligare granskning.

Grupp 4, innefattade de artiklar som bara nämner att de i framtida arbeten kan använda sig av ”web scraping” eller de som bara nämner ”web scraping” och sedan inte tog upp något mer om metoden.

Artiklar om medicinsk skrapning (gynekologi) togs också bort. Grafen (fig.5) nedan visar att största delen av de 100 genomsökta artiklarna antingen bara tillämpar webbskrapning utan att beskriva fenomenet, inte är relevanta eller att de kräver betalning för publik åtkomst. Bara 10 av artiklarna beskriver metoder för webbskrapning.

Fig 5. Resultatet efter genomgång av de 100 första länkarna i google scholar (med sökkriteriet ”Web scraping, med filter, med citationstecken, från student inloggning”)

Efter att ha jämfört de olika sökningarna gjordes bedömningen att artiklarna från summon@bth.se inte hade tillräcklig information eller samma information som scholar.google om webbskrapning.

De var antigen dubbletter med scholar.google.se-sökningarna eller handlade om någon annan typ utav skrapning, som beskrivits tidigare.

Förutom information från publika källor så valdes därför de 10 identifierade relevanta vetenskapliga artiklarna ut ur scholar.google.se för att ligga till grund för besvarandet av frågeställningarna. En vidare undersökning av de 10 artiklarnas referenser visar att 7 av 10 inte anger någon tidigare referens om webbskrapning, medan de återstående använder wikipedia som referens.

4.1.5 Kreativt utnyttja

Information ur litteraturstudien samt intervjuer av anställda på Sentor AB har används för att besvara frågeställningarna. Av de tio utvalda artiklarna och utifrån sökningarna i det öppna nätet upptäcktes det att det fanns en outforskad aspekt inom området webbskraping. Ingen har undersökt den

trafikmängd som webbskrapningsbotar genererade.

Det som fanns beskrivet var hur man kan *använda* skrapning för att samla in information samt hur man kan *förhindra* hur olika botar samlar in information. Man har inte undersökt om botarna verkligen stoppas eller hur stor belastning de kan tänkas utgöra för en enskild hemsida.

Avsaknaden av information användes som utgångspunkt vid skapandet av frågställning F5:

> *Hur kan man sätta upp en pilotstudie som undersöker trafikeffekt av webbskrapning?*

Idén var att undersöka om robots.txt (förklaras i kap. 5.4) efterlevs av de stora sökmotorernas botar och undersöka hur stor trafikmängd de olika botarna genererar.

5 Resultat

I resultatkapitlet sammanställs den information som har hittats under informationssökningsprocessen om ämnet webbskrapning. För att organisera informationen har frågeställningarna använts som rubriker.

5.1 Vad är webbskrapning?

Enligt den information som insamlats om ämnet så dras samma slutsats som, (Jennings & Yates, 2009) att det inte finns någon universell definition av *skrapning* i IT-säkerhetssammanhang. Begreppet omfattar ett antal olika metoder för att erhålla data från en webbplats eller en databas. Detta i regel genom att använda ett för syftet utformat datorprogram. Dataskrapning kan beskrivas utifrån de följande två huvudmetoderna:

- **Skärmskrapning (Screen scraping):** här extraherar skrapningsprogramen bara den information på hemsidor som kommer att visas på slutanvändarnas egen skärm. Detta är en gammal metod och användes främst när användargränssnitten var textbaserade och då oftast mot terminaler. När en ”skrapare” idag tillämpar metoden så fångas skärmdata in i bitmap format och körs sedan genom en OCR-programvara.
- **Webb-skrapning:** Detta innebär användning av en ”skrapare” för att extrahera alla de uppgifter som rör den underliggande strukturen i HTML dokument som används för att på skärmen skapa webbplatsen, (och inte bara den data som visas för besökaren via dess skärm). Sådana program benämns ofta på engelska som: *”bots”, ”web-bots”,” Web Wanderers”,” Spiders” och ”crawlers”*, och på svenska som: *"skördare", ”skrapare” eller ”spindlar”* (Jennings & Yates, 2009), (Wikipedia, 2012).

Då skärmskrapning inte är lika utbrett och kräver mycket mer datahantering, kommer fokus att ligga på att utveckla och synliggöra likheterna och olikheterna mellan webbskrapning och webbindexering.

Webbindexering är nära besläktat med webbskrapning. Båda går ut på att snabbt och automatiskt söka igenom stora mängder data i hemsidor och extrahera relevant information för att sedan kunna leta efter informationsmönster. Vad informationen används till och hur lätt det är för den utsatte att kunna skydda sin information är det som skiljer dem åt. Både teknikerna bakom webbindexering och webbskrapning kommer att förklaras djupare i kap. 5.2 - 5.4, men generellt så går webbindexering lättare att stoppa med en deklarativ fil på webservern och används främst

för att hjälpa en sökande användare att finna den genomsökta hemsidan i en sökmotor.

Webbskrapor söker i regel efter specifik information så som e-postadresser, prisuppgifter eller foruminlägg genom att tolka dessa och extrahera ut datapunkter i förhållande till deras HTML eller XHTML struktur (Alba et al., 2009), (Wikipedia, 2012). ”Skraparen” kan då komma åt icke formaterad data från en hemsida utan ägarens tillåtelse (Wikipedia, 2012). Informationen skalas normalt av ifrån all immaterialrätt och kan därför lätt användas igen genom att t.ex. återges som att den tillhör en annan avsändare, i programvaror, via ett tjänsteerbjudande eller säljas vidare till tredje part (Poggi et al., 2007).

5.2 Vilka tekniker används vid webbskrapning?

Ur den genomförda litteraturstudien framkom det att det finns flera olika skrapningstekniker som är olika mycket automatiserade. För att undgå upptäckt kan webbskrapningsprogrammen simulera en riktig människas sätt att söka på en hemsida tillexempel fördröjningar mellan frågor, länkar som utgår ifrån startsidan etc. Genom att exempelvis implementera ett skript[α] eller genom att maskera förfrågningen så ser det ut som att förfrågningen kommer från en allmänt använd webbläsare som t.ex. Internet Explorer eller Mozilla Firefox. Här nedan följer de 6 vanligaste sätten som webbskrapning kan utförs på (Wikipedia, 2012):

1) HTML-programmering:

Statiska och dynamiska webbsidor kan hämtas genom att skicka HTTP-förfrågningar till Webbservrar.

2) Reguljära uttryck:

Ett kraftfullt sätt är att använda reguljära uttryck. Dessa uttryck används för att söka fram mönster i texter. Började användas i UNIX-kommandon och texteditorer så som *grep* och *ed* för att filtrera text. Idag så stöds reguljära uttryck i de flesta programmerings- och skriptspråk. De implementeras i programmeringsspråk som till exempel Perl eller Python.

3) DOM-tolkning:

Genom att bädda in en riktig webbläsare som Internet Explorer eller Mozilla och låta de hämta det dynamiska innehållet som skapas av klientsidans skript. Dessa webbläsare kontrollerar också tolkningarna av

α eller ge är ett litet program som tillåter kontroll över en eller flera applikationer.

webbsidor i ett DOM-träd[α]. Baserat på vilka program som används så kan olika delar av webbsidorna hämtas.

4) HTML-tolkning:

Vissa halvstrukturerade[β] data och frågespråk[γ] som XQuery och HTQL kan användas för att tolka HTML-sidor och för att hämta och omvandla webbinnehåll.

5) Webbskrapningsprogramvara:

Dessa program kan automatiskt känna igen datastrukturen för en sida eller ge ett gränssnitt för webbinspelning som tar bort behovet av att manuellt skriva webbskrapningskod. Funktioner kan användas för att utvinna och omvandla webbinnehåll och databasgränssnitt som kan lagra den skrapade datan i lokala databaser. Här följer exempel på några typiska webskrapningsprogramvaror:

Sitescraper lär sig automatiskt, XPath-baserade, mönster för att identifiera var en användarbestämd lista av strängar återfinns på en viss webbsida. För att träna Sitescraper ges en liten uppsättning exempel av webbadresser, från en given hemsida, och strängar som användaren önskar att skrapa från. Denna information används sedan för att generera en XPath förfrågan som beskriver var de önskade strängar finns och som kan tillämpas när en webbsida med en liknande struktur skall skrapas (Penman, Baldwin & Martinez, 2010).

AgentMat är utformat för att på ett effektivt sätt utvinna stora mängder data från webbsidor. Webbskrapningsspindeln är baserat på ett XML-språk. Enligt artikelförfattarna skrapar spindeln först all text på hemsidan, och inte hemsidestrukturen (eng. site maps), i sin helhet. När nästa skrapning genomförs på hemsidan så skrapas bara ny information och förändringar. Detta gör att AgentMat blir effektivare ur ett tids- och datamängds perspektiv för att spindeln inte behöver skrapa all information på hemsidan. (Beiio, Misek & Zavoral, 2009).

SharpSpider är ett C# baserat webbskrapningsprogram som är konstruerad för att hantera förändringar som skalning, decentralisering och kontinuitet hos en hemsida. Till stor del liknar webbskrapningsprogrammet AgentMat i sin funktionalitet men SharpSpider har tillskillnad från AgentMat ett användarvänligt API och distribueras gratis. SharpSpiders API möjliggör för en användare att lätt kunna skräddarsy sin skrapningsspindel (Moody & Palomino, 2003).

[α] **DOM:** Document Object Model är en specifikation för ett programmeringsgränssnitt från W3C som tillåter att program och skript får uppdatera innehåll, struktur och stil HTML och XML-dokument.

[β] **Halvstrukturerat:** beroende på vad som hittas så ändras reglerna i formuläret.

[γ] **Data frågespråk:** är programspråk för att söka eller modifiera data lagrad i databaser.

6) Analys av metadata:

Webbsidor kan omfatta metadata eller semantiska markeringar/anteckningar som kan utnyttjas för att lokalisera specifika databitar, se fig. 2. Om anteckningarna är inbäddade i sidorna som Mikroformat[α] så kan denna metod ses som ett specialfall av DOM tolkning (se punkt 3 ovan).

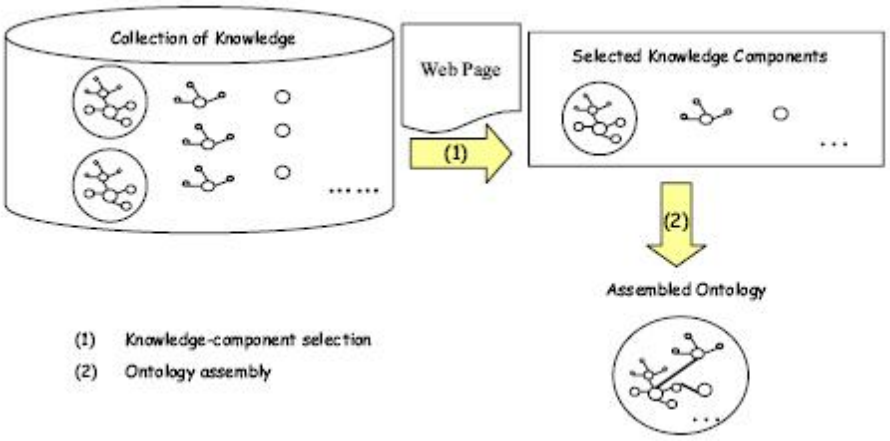

Fig. 2. Visar analys av metadata; (1) är samlingen av all metadata på hemsidan. (2) visar att bara viss åtråvärd data samlas in samt sätts sedan ihop i en ny lista.

[α] **Mikroformat**: Mikroformat är en utökad semantik, det vetenskapliga studiet av språklig betydelse, som åstadkommer i (X)HTML möjligheten att märka upp information.

5.3 Är webbskrapning ett problem?

Det finns både för- och nackdelar med webskrapning. Enligt Enck et al. (2005) är den vanligaste användaren av webbskrapningsprogramvaror just en spammare. De använder webbskraparen till att på ett effektivt och automatiserat sätt samla in e-postadresser som publicerats på olika webbsidor. Men de kan även använda tekniken för att samla in information om potentiella mål.

Informationsstöld på nätet är ett verkligt problem för många informations- och näthandelsföretag (Poggi et al., 2007). Användningen av webbskrapning skiftar från enkla plagiat av hemsidor till att i vinstdrivande syfte använda webbskrapning till att automatiserat samla in och använda skyddad information.

De företag som drabbas värst är ofta de som gratis tillhandahåller tjänster åt sina användare som t.ex. webbaserade katalogföretag, köp-och-säljsidor, dejtingtjänster eller resebyråer. Ideella informationsplatser som Wikipedia har också varit särskilt utsatta för webbskrapningar av data (Poggi et al., 2007).

Andra negativa problem uppstår av själva webbskrapningsprogrammen, de kan överbelasta de angripna företagens IT-infrastruktur genom att de vid sina skrapningar genererar massiv Bot-trafik på företagens servrar. De hårt överbelastade företagen förlorar då kunder, reklamintäkter vartill kommer att de även i vissa fall förlorar kontrollen över sitt datainnehåll genom att en annan aktör har en kopia på datainnehållet (Jennings & Yates, 2009), (Zadel & Fujinaga, 2004).

Webbskrapning kan även användas som ett aggressivt konkurrensmedel. Ett webbföretag skrapar konkurrentens sida i realtid och slipper själv skapa nytt innehåll. Genom t.ex. bannerreklam tjänar det skrapande företaget pengar utan att behöva lägga ner kostnader på att skapa innehåll. Kunderna får även svårare att urskilja vem som är avsändare av tjänsten, vilket påverkar det utsatta företagets varumärke. (IDG, 2005) Det angripna företaget får också betala för den bandbreddstrafik som webbskrapningsprogrammen genererar, vilket innebär att de förlorar pengar på trafik de inte kan utnyttja.

Ur ett integritetssäkerhetsperspektiv är för den enskilde användaren okontrollerad webbskrapning ett problem. Personlig användarinformation som skall vara skyddad hos en e-tjänsteleverantör kan genom webbskrapning bli fritt tillgänglig för ”skraparen”.

Positiva aspekter med webskrapning är att för en informationssökande person kan webbskrapning samla in information från hela nätet om ett visst ämne och personen kan sen sammanställa all data på en för ändamålet specialiserad hemsida. Personen behöver då inte själv lägga ner tid på att

hitta, utvärdera och sortera relevant information. Han/hon kan tillexempel hitta de billigaste flygresorna genom att enkelt jämföra alla flygbolags reseerbjudanden på en och samma webbplats. (Poggi et al. 2007).

> *”Som ett exempel tas online reseförsäljningsindustrin. Onlineresebyråer har avtal med Global Distribution Systems (GDS) som under specifika SLA och ”look-to-book” förhållanden (antalet sökningar per bokning). När en användare gör en sökning på en flygresa så skickas begäran via olika webbtjänster till GDS, som i de flesta fall skickar begäran vidare till flygbolag så att de kan ta fram den slutliga tillgängliga flygresan. Nyligen har flygjämförelsesajter dykt upp som i realtid skrapar flera resesajter och kombinerar resultaten på sin egen hemsida. Trots att det kan vara till nytta för användarna, så blir det ett problem för de riktiga resebyråerna och resten av leveranskedjan, eftersom varje ny sökning är resurskrävande och kostsamt. Flygjämförelsewebbplatserna ökar också look-to-book förhållandet vilket gör att kostnaderna av den ökade trafiken belastar resebyråerna, medan flygjämförelsesiterna har i princip ingen kostnad. I allmänhet så upptäcks sådana platser manuellt av administratören och lösningen har vart att manuellt blockera deras IP-adress. Men precis som i fallet med spammare så inför webbskrapare ständigt nya tekniker för att kringgå upptäckt, såsom IP ombud eller IP pooler.”*
>
> *Citat fritt översatt av författaren från Poggi et al. (2007).*

5.4 Hur kan man förhindra webskrapning?

Man kan dela in webbskrapningsprogram i de som respekterar instruktioner i innehållet i filen robots.txt (fig. 3) och de som inte gör det. Detta är en fil som ligger i roten på en hemsida och ger instruktioner om vad sökspindlar får och inte får indexera. Ofta finns det en uppsättning sidor på en hemsida som inte utomstående skall kunna hitta via en sökmotor, till exempel administratörsinloggningssidor eller medlemsforum. Då anger man detta i robots.txt filen. Om denna textfil inte existerar så ser sökspindlarna hela hemsidan som tillgänglig att indexera (Wikipedia ref 2, 2012). Som tagits upp innan är grundtanken med sökspindlar enkel: Allt som går att indexera, kommer att indexeras.

Sökleverantörer som Google, Yahoo och Bing använder sig av en webbindexeringsbot för att indexera webbsidor. Men med hjälp av textfilen med namnet robots.txt, som ligger i rotkatalogen på webbservern, exkluderas vissa kataloger/sidor som ägaren inte vill att sökleverantörernas webbindexeringsbot ska söka igenom.

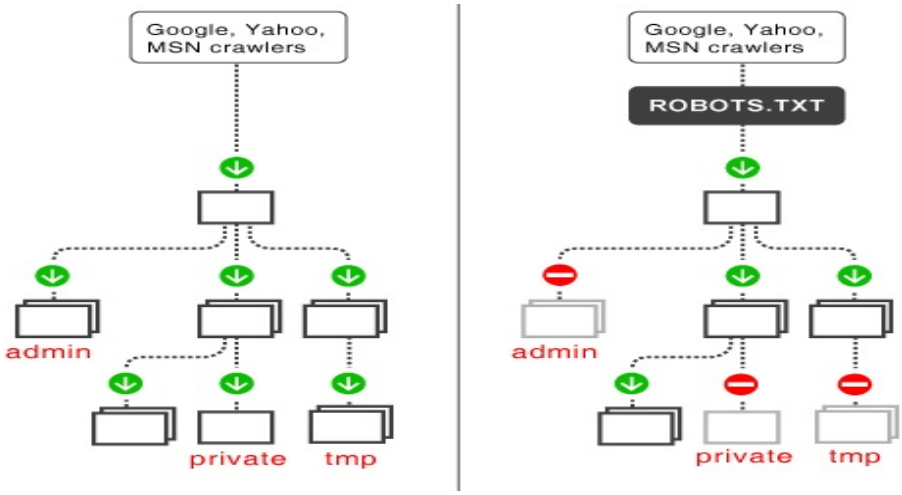

Fig.3. Till vänster en hemsida utan robots.txt, till höger en hemsida med robots.txt

Dessa botar växlar även sina förfrågningar mellan olika sidor och begär inte dokument från samma server mer än någon gång under en period på några sekunder, vilket innebär att företags IT-system påverkas i mycket mindre grad än när mindre nogräknade webbskrapningsspindelar konstant skickar förfrågningar.

Här är ett exempel på syntaxen i filen robots.tx (Robotstxt, 2012), (Backendmedia, 2005):

User-agent: Namn på sökspindel

Disallow: /katalog/

User-agent anger namnet på sökspindeln. Google har exempelvis namnet googlebot. Disallow anger en fil eller katalog som inte ska indexeras.

Några exempel på robots.txt:

Ge instruktioner till sökspindlar att *inte* indexera en fil och en katalog:

User-agent: *

Disallow: /filnamn.html

Disallow: /katalog/

Säg åt Google att *inte* indexera katalogen /hidden/:

User-agent: googlebot

Disallow: /hidden/

Säg åt Google och Yahoo att *inte* indexera bilder som finns i katalogen /bilder/, men däremot alla andra bilder:

User-agent: Googlebot-Image

User-agent: Yahoo-MMCrawler

Disallow: /bilder/

Säg åt alla sökspindlar att *inte* indexera någonting på hela webbplatsen:

User-agent: *

Disallow: /

Det finns även ytterligare några direktiv som det går att kontrollera sökspindlarna med men de stödjs inte av alla sökleverantörer. De beskrivs nedan (Wikipedia ref 2, 2012).

Crawl-delay direktivet

Flertalet stora sökrobotar stödjer en crawl-fördröjningsparameter, som sätts till det antal sekunder som sökspindeln ska vänta mellan successiva förfrågningar till samma server:

Till exempel säg åt alla sökspindlar att vänta 10 sekunder mellan förfrågningarna:

User-agent: *

Crawl-delay: 10

Allow direktiv

Vissa större sökspindlar stödjer även ett Allow direktiv som kan motverka ett Disallow direktiv. Detta är användbart när man förbjuder (”Disallow”) en hel katalog, men ändå vill att några specifika HTML-dokument i den katalogen ska genomsökas och indexeras. Standardgenomförandet av ”Disallow” i robots.txt ”vinner” alltid över ett likadant matchande ”Allow” i robots.txt, men Googles genomförande skiljer sig

genom att tillåta Allow mönster med lika eller fler tecken i direktivet att vinna över ett matchande ”Disallow” mönster.

Bing använder det ”Allow” eller ”Disallow” direktiv som är mest signifikant. För att vara kompatibel med alla godartade sökspindlar, och om man vill tillåta enstaka filer i en annars otillåten katalog, är det nödvändigt att placera ”Allow” direktivet först, följt av ett ”Disallow” direktiv.

Detta exemplet kommer att förbjuda: /katalog/ *utom* /katalog/minfil.html, eftersom den skulle matchas först, i händelse av Googles sökrobot så är ordningen inte viktig.

Allow: /katalog/minfil.html

Disallow: /katalog/

Sitemap

Vissa sökspindlar stödjer även ett Sitemap[α] direktiv, vilket gör att flera sitemaps kan beskrivas och skrivas in på samma robots.txt.

Till exempel:

Sitemap: http://www.gstatic.com/s2/sitemaps/profiles-sitemap.xml

Sitemap: http://www.google.com/hostednews/sitemap_index.xml

Universal "*" Match

Den första versionen av Robot Exclusion Standard nämner inte något om "*"-tecknet i ”Disallow:” uttalandet. Vissa sökspindlar som Googlebot och Slurp erkänner strängar med "*", medan MSN-bot och Teoma tolka det på olika sätt.

Om en webskrapningsspindel ändå skulle följa robots.txt behöver du också veta User-Agent namnet för att kunna skapa ett fall som kan utesluta den specifika webbskrapningsspindeln i en robots.txt fil.

Däremot har företaget Augures upptäckt att webbskrapnings-botar aldrig köper något (Poggi et al., 2007). Genom att prioritera användarna på en webbhandelsplats genom deras köpbeteende och anta webbhandelsplatsens förväntade intäkter på användarna upptäckte de att webbskrapnings-botar systematiskt tilldelades en mycket låg prioritet och då kan brandväggen avbryta deras sessioner men detta sker bara när systemets resurser blir för är knappa.

[α] **Sitemap**: är en lista över sidor på en webbplats tillgänglig för crawlers eller användare.

Andra sätt att förhindra webbskrapning är att konfigurera servern till att göra autentiseringar[β], och konfigurera lämpliga tillstånd. Man kan i så fall använda sig av till exempel Sprite bilder[γ]/captcha[δ], för att få åtkomst till den informationen. Moderna innehållshanteringssystem stödjer ofta åtkomstkontroll för enskilda sidor och samlingar av resurser.

Ett annat sätt är att använda sig av RID/”Error seeding” en teknik för att uppsåtligen lägga till fel i HTML-koden som gör att den skrapade datan blir oanvändbar för botar men vanliga användare blir inte drabbade (Measurement and Instrumentation, 2011), (IEEE International Conference on Web Services).

Om man vill fånga en skrapningsspindel eller försöka krascha den så kan man gillra en ”spindelfälla” (eng. crawler trap) i en oändlig omdirigeringsslinga. Det är en uppsättning av webbsidor som avsiktligt används för att orsaka en spindel att göra ett oändligt antal förfrågningar. Detta leder i sin tur till att dåligt konstruerade spindlar får slut på minnesresurser och kraschar. Spindelfällor kan även skapas för att "fånga" spamrobotar[ε], eller andra spindlar som använder en webbplats bandbredd. Det finns idag ingen algoritm för att detektera alla spindelfällor. Vissa klasser av fällor kan upptäckas automatiskt och en del spindlar kan undvika att begära resurser som har ett ”?” i sig (Wikipedia ref 3., 2012). Detta indikerar ofta att det är en dynamisk sida och på så sätt undviker en webbspindel att ladda ner oändligt många URL:er från samma hemsida.

Sajter med spindelfällor brukar ha en robots.txt implementerad som berättar för sökspindlar att inte gå i fällan, så att en indexerande spindel inte fastnar i fällan, medan en spindel som bortser från robots.txt inställningarna påverkas däremot utav fällan. Några möjliga tekniker som kan används för att skapa ”spindelfällor” är:

1) att skapa ett stort djup på katalogens strukturer som tillexempel http://bla.com/oop/bip/oop/bip/oop/bip/oop/... Detta för att spindelns resurser ”äts” snabbt upp om den måste travesera djupt ner i en katalogstruktur.

2) Dynamiska sidor som exempelvis kalendrar kan kontinuerligt producera ett oändligt antal sidor som kan få en spindel att besöka nästa sida i all oändlighet och detta är resursödande för en spindel att följa..

3) Sidor fyllda med ett stort antal tecken som kraschar den lexikaliska analysatorn, en process att konvertera en sekvens av tecken till en

β **Autentisering**: kontrollera identiteten på användare.

γ **Sprite bilder**: en datorgrafisk komponent som kan flyttas runt på skärmen oberoende av annan grafik.

δ **Captcha**: ett test som antas vara lätt att lösa för människor, men inte för automatiska datorprogram.

ε **Spamrobotar**: robotar som skickar mängder av oönskad e-mail reklam/virus.

sekvens av tokens[α].

4) Använda sig av en Tarpit/Sticky honeypot /kvicksandshål. Till exempel LaBrea, detta program utför en Denial-of-service-attack mot boten (Lorgor, 2013).

Administratören av en webbplats kan även använda sig av olika åtgärder för att stoppa eller bromsa en skrapningsspindel. Här är några tekniker:

1) Som beskrivits tidigare så kan man hindra en webindexeringsspindel genom att lägga till segment i robots.txt som kommer att följas om den håller sig till standarden.

2) Blockera en IP-adress i nätverksbrandväggen. Detta kommer att blockera alla trafik som kommer från den adressen.

3) Spindlar förklarar ibland vilka de är och kan på så sätt blockeras, "Googlebot" är ett exempel på en spindel som talar om vem den är. Men vissa spindlar gör ingen skillnad mellan sig och en ”mänsklig” webbläsare.

4) Spindlar kan blockeras med hjälp av att övervaka de trafikmönster den genererar, skapas det onormalt mycket trafik eller att sidor besöks enligt ett ologiskt mönster från en viss adress så kan denna spindel blockeras.

5) Kommersiella anti-bot tjänster: Här finns det företag som Sentor som analyserar och blockerar trafik.

6) Använda sig av en webbapplikationsbrandvägg, några av dessa har även en begränsad möjlighet att upptäcka spindlar.

7) Hitta spindlar med en ”honeypot”/spindelfälla för att automatiskt identifiera IP-adressen och sedan blockera denna.

8) Använda sig av CSS sprites[α] för att visa sådana uppgifter som telefonnummer eller e-mail adresser vilket gör informationen svårskrapad.

[α] **Tokens:** En ”biljett” används som en substitut valuta, som poker chips.

[α] **CSS sprites**: CSS-sprite är möjligheten att utnyttja olika delar i en bild och minska förfrågningarna mot servern.

9) Ändra HTML-taggar och strukturen på hemsidan regelbundet. Detta för att försvåra det för en återvändande skrapningsspindeln att hitta informationen.

Det man bör tänka på är att spindelfällor också kan skapas oavsiktligt av kalendrar som använder sig av dynamiska sidor med länkar som ständigt pekar till nästa dag eller år. Detta kan göra att indexeringsspindlar som du vill skall indexera din hemsida kan få problem.

5.5 Hur kan man sätta upp en pilotstudie som undersöker trafikeffekten av webbskrapning?

Inga av de undersökta artiklarna eller arbetena innehöll någon information om vilken trafikmängd webbskrapningsbotar genererar när de under ett dygn kontinuerligt skrapar en hemsida. Däremot fanns det en akademisk artikel (Poggi et al., 2007), som beskrev hur man kan förhindra webbskrapningsbotar.

Därför formulerades ett experiment som förhoppningsvis ska kunna utröna om man kan mäta trafikmängden av webbskrapningsbotar.

5.6 Pilotstudien

Antaganden: att kunna empiriskt kvantitativt undersöka hur mycket trafikmängd olika typer av spindlar skapar på en hemsida under ett dygn och hur ofta de återvänder till samma hemsida.

Design av pilotstudien

Experimentet kommer att gå ut på att jämföra två hemsidors trafikdata och trafikmängd under 24 timmar en vanlig arbetsdag. Båda skall vara helt nya och aldrig ha varit indexerade eller besökta av vanliga användare. En ny domänadress skall användas med två identiska hemsidor där försöksledaren har fullkontroll över hemsidornas loggar.

Experimenthemsidorna skall vara oberoende av varandra och inte vara länkade till någon annan hemsida samt att ingen utomstående hemsida heller skall vara länkad till experimentsidorna.

Den ena hemsidan har en robots.txt som förbjuder alla webbskrapningsspindlar att indexera hemsidan. Den andra hemsidan är helt ”öppen” för all trafik.

Den loggade trafiken på hemsidorna jämförs sedan empiriskt kvantitativt. De båda hemsidornas loggar jämförs och man räknar hur många botar som följer robots.txt och hur många som inte följer robots.txt. Man räknar också hur mycket trafik som genereras av olika sorters webbskrapningsbotar.

Analysfrågor på resultat

1) Skiljer sig trafiken mellan sidorna?
2) Skiljer det sig mycket mellan tidpunkten de olika hemsidorna upptäckts?
3) Följs robots.txt?
4) Går det att skilja de olika spindlarna åt?

Förväntat resultat av pilotstudien

Trafiken som genereras på sidan med robots.txt bör enbart genereras av inte indexerade spindlar och den trafik som genereras av sidan utan robots.txt bör bestå av trafik från båda sorterna av spindlar.

Man bör kunna undersöka hur lång tid det tar innan sidorna får besökare och om det skiljer sig mycket åt mellan hemsidorna. Hemsidan med robots.txt borde få mycket mindre trafik.

6 Diskussion

6.1 Metoddiskussion

Som stöd för informationsinsamlingsprocessen har Södertörns Högskolas informationssökningsprocess använts. Detta för att när kandidatarbetet genomfördes fanns den, för arbetets syfte, vara en pedagogisk och lättöverskådlig mall och därför bättre anpassad till denna litteraturstudie än Blekinge Tekniska Högskolas.

De ostrukturerade intervjuerna som genomfördes med anställda på Sentor AB gav en bra utgångspunkt för de sökord som användes. Detta var ett bra sätt att snabbt få en inblick i området och få en bra utgångspunkt för vidare informationsinsamling, men med den nackdelen att man kan bli påverkad av respondenternas egna uppfattningar.

Val av sökkällor har skett med kriteriet att på effektivaste sätt och med tillräcklig täckning av publicerade artiklar finna relevant information. Google.se och scholar.google.se är den i världen mest använda sökmotorn (Wikipedia, 2013) och Blekinge Tekniska Högskolas biblioteksdatabas bedömdes ha tillräcklig täckning av böcker samt akademiska tidskrifter.

Det var en fördel att lägga upp informationssökningen som två separata studier och inte använda samma sökmotorer i varje studie. Det gav då ett större informationsunderlag samt att man kunde jämföra sökresultaten och kontrollera materialet dem emellan.

Vid analys av den framkomna informationen om ämnet formulerades det en idé som sedan låg till grund vid formulerandet av frågeställning F5 och den pilotstudie som redovisades i kap. 5.5. Tyvärr gick inte studien att genomföras inom ramen för detta arbete. Pilotstudien kommer därför att redovisas som ej genomförd.

6.2 Resultatdiskussion

Här nedan diskuteras det resultat som redovisades i resultatkapitlet.

6.2.1 Vad är Webbskrapning?

Den genomförda informationsinsamlingen gav resultatet att det inte finns någon klar akademisk definition av vad webbskrapning är samt att det var väldigt få artikelförfattare som ens försökte definiera begreppet. När begreppet väl refererades till hänvisades författaren till Wikipedias definition av webbskrapning, dvs. att med hjälp utav programvaror utvinna information ur hemsidor. Därför har den definitionen legat till grund för besvarandet av frågeställningen.

Att det ser ut på detta sätt kan bero på att webbskrapning är ett ganska okänt fenomen och har i majoriteten av artiklarna bara använts för att beteckna insamling av stora mängder information till diverse experiment. Artikelförfattare har därför inte varit fokuserade på att vidareutveckla definitionen av webbskrapning.

6.2.2 Vilka tekniker används vid Webbskrapning?

Av litteraturstudien framkom det att det finns flera olika skrapningstekniker som var olika mycket automatiserade. För att undgå upptäckt simulerar vanligtvis webbskrapningsprogrammen en människas sökbeteende på en hemsida. Antingen genom att implementera ett skript eller genom att maskera förfrågningen så att det ser ut som att den kommer från en allmänt använd webbläsare som t.ex. Internet Explorer eller Mozilla Firefox. Ingen av de 10 artikelförfattarna gick in i detalj på hur de implementerat detta med sökfunktionen i boten eller hur allt hängde ihop inne i boten. Däremot gav informationsinsamlingen en bra överblick av de vanligaste protokoll och script som man kan använda vid webbskrapning.

6.2.3 Är Webbskrapning ett problem?

Det beror på vem man frågar, är det företag och innehållsleverantörer så upplever de nog att det är ett problemen med informationsstöld, internetmissbruk, serveröverbelastning, förlust av reklamintäkter, förlust av eller devalvering av innehållet. Vilket gör hemsidan mindre unik och minskar dess immateriella värde (eller varumärkesvärde). Medan personer som kan få ta del av den skrapade informationen, som exemplet med

flygindustrin och köp av flygbiljetter, nog är positiva till webbskrapning fast utan att veta om hur det går till. Men som beskrevs i resultatdelen så är det främst spammare som utnyttjar tekniken (Enck, et al., 2005). Webbindexering däremot är vida använt av sökmotorer som google.se. Merparten av författarna i de tio artiklarna som analyserats hade utvecklat egna programvaror för att utföra skrapningar. Ingen av författarna reflekterade över om deras skrapningsprogram följer lagar och regler för upphovsrättsliginformation eller om de kunde bryta mot personuppgiftslagar. Slutsatsen som drogs ur informationsinsamlingen var att webbskrapning ligger i en gråzon och att de orsakar problem för de företag eller organisationer som utsätts för detta.

Troligtvis upplevs webbskrapning på samma sätt som DDoS-attackerna i mitten/slutet av 90-talet. Så länge det var relativt okänt för allmänheten så förblev det en attack som låg i gråzonen. Detta ända tills attackerna blev mer och mer offentliga och politiker och allmänheten fick inblick i vilka problem denna typ av attack kunde skapa. Efter detta så har DDoS-attacker blivit ett allvarligt brott som i vissa länder och fall getts hårda straffrättsliga påföljder.

De angivna problemen var informationsstöld i olika former samt att vid överbelastning så kunde företagen förlora pengar. Detta för att de inte kunde utnyttja den bandbredd som de betalat för eller att de betalar för mycket bandbredd som bara används av skrapningsbotar.

Den informations- och immaterialrättsstöld som webbskrapning medför gör att utförare av skrapning är måna om att hålla det så hemligt som möjligt samt att undvika att belysa vilka problem skrapning kan skapa. Den skulle nämligen kunna användas som en vidareutvecklad variant av DDoS med den skillnaden att inte bara överbelasta servrarna även stjäla företagsdata.

Hur utbrett detta problem är finns det ingen riktig statistik på. Pilotstudien var tänkt att ge en första indikation på trafikmängden och omfattningen av botar som söker igenom nätet.

Ett verkligt exempel på detta är:

> *AllaAnnonser.se är en sida som systematiskt skrapar hemsidor som Blocket.se och har varit inblandad i en rättslig tvist med Blocket.se.*
>
> *Detta slutade med att Blocket.se och AllaAnnonser.se gjorde en förlikning och ingick ett samarbete. Detta efter att tingsrätten givit Blocket.se ett vite på 200 000 kr.*
>
> *Då det finns företag som rättsligen försöker hindra konkurrenter att utnyttja deras information är skrapning uppenbart ett problem.*
>
> *Ur IDG (2005)*

6.2.4 Hur kan man förhindra webskrapning?

Efter att ha gått igenom både akademiska och allmänna källor så är slutsatsen att man inte kan fullt ut hindra skrapning av hemsidor. Detta på samma sätt som man inte fullt ut kan hindra någon IT-attack, det finns inga vattentäta system. Vill någon skrapa en hemsida och har tillräckligt med resurser så kommer de att lyckas. Men man kan försvåra processen att införskaffa informationen så att priset för att genomföra skrapningen blir så dyrt att det inte längre är ett attraktivt tillvägagångssätt. Detta kan man uppnå genom att antingen köpa in en tjänst eller ett verktyg från ett IT-säkerhetsföretag som övervakar trafiken dygnet runt och fångar eller blockerar webbskrapningsspindlar (Sentor, 2011). Alternativt skapar man ett eget system som uppnår samma mål.

Andra sätt är att blockera kända spindlars IP-adresser som går att få tag på ur ”svartalistor” från t.ex. www.projecthoneypot.org.

Det man dock ska komma ihåg är att det som är lagligt i ett land kan vara olagligt i ett annat land. Den lagstiftning som gäller beror på var ”attacken” utfördes och om landet som attacken var riktad emot har utlämningsavtal.

Webbindexerare som följer robots.txt är däremot lätta att förhindra men det kan finnas en viss nackdel med att använda filen. När man listar sidor eller kataloger i robots.txt filen kan man bjuda in till oavsiktligt tillträde genom att man visar vart man inte vill att spindlar skall genomsöka. Det finns två sätt att se på detta.

Det första är att man lägger alla filer du inte vill att spindlar ska besöka i en separat underkatalog, gör sedan denna katalog ”olistbar” på webben (genom att konfigurera servern att inte lista den katalogen), placera sedan dina filer där och lista bara katalognamnet i robots.txt.

Men om man istället arbetar efter det antagandet att spindlarna arbetar från unika IP-adresser så blir det möjligt att blockera tillgången till dessa i din webbserver, via serverns konfiguration verktyg eller i nätverksbrandväggen. Men om kopior av spindeln verkar från flera olika IP-adresser, som kan vara kapade datorer som ingår i ett stort botnet[α], blir det svårare. Det bästa alternativet kan då vara att använda avancerade brandväggsregler som konfigureras till att automatiskt blockera IP-adresser som gör för många anslutningar, men det kan även slå bort webbindexerare.

[α] **Botnet**: är ett nätverk av datorer infekterade av datavirus eller trojanska hästar. Dessa datorer ansluter till en central styrande nod där de får uppgifter att utföra.

6.2.5 Pilotstudien

Pilotstudien borde vara intressant att genomföra då inga hittade källor har undersökt hur mycket trafikmängd ”webbskrapare” eller ”webbindexerare” generar på en hemsida. Det kan också vara intressant att få reda på hur ofta de återväder till sidan och vilka ”webbskrapare” och ”webbindexerare” som är mest frekventa. Man bör också kunna säga något om hur väl robots.txt följs.

Pilotstudier eller explorativa studier är ett sätt att få en inblick i en frågeställning eller idé utan att vara helt säker på dess utkomst. De är ett första steg till att genomföra mer kontrollerade studier. En sådan studie bör nog planeras som en långtidsstudie alternativt att man gör flera kortare studier under en längre period för att utröna om t.ex. trafikflödet fluktuerar eller hur ofta olika ”webbskrapare” återkommer.

7 Slutsats

Förekomsten av webbskrapning ökar av flera skäl: enkelheten att simulera mänsklig navigering, svårigheten att hålla robotar isär från människor, det gråa området på den rättsligastatusen och, viktigast av allt, lönsamhet i verksamheten (Poggi et al., 2007).

S1: Vad är webbskrapning?

Webskrapning är när en programvara automatiskt söker igenom och fångar information på hemsidor utan att fråga ägaren av informationen om lov. Skraparen använder sedan oftast informationen för egna ändamål och kan direkt eller indirekt skada den ursprungliga ägaren av informationen. En variant av webbskrapning är skärmskrapning som kopierar vad som visas på en användares skärm, ofta i bitmap format. Webindexering kan tekniskt sett likställas med webbskrapning men uppsåtet med webindexering är inte att direkt skada informationsägaren utan att underlätta för informationssökare att nå informationsägarens hemsida.

S2: Vilka tekniker används vid webbskrapning?

För att undgå upptäckt simulerar vanligtvis webbskrapningsprogrammen en riktig människas sökningar på en hemsida. Antingen genom att implementera ett skript eller genom att maskera förfrågningen så att det ser ut som att förfrågningen kommer från en allmänt använd webbläsare som t.ex. Internet Explorer eller Mozilla Firefox. De 6 vanligaste sätten att utföra webbskrapning redovisas i kap. 5.4.

S3: Är webbskrapning ett problem?

Både ja och nej. Webbskrapning är ett problem för personer, företag och organisationer som vill skydda eller ha kontroll över vilka som använder deras data. Webbskrapning kan även användas till att överbelasta servrar precis som vid DDos-attacker. Men det finns även positiva aspekter av webbskrapning. Personer, organisationer eller företag som vill jämföra information och olika erbjudanden är hjälpta av att kunna söka informationen på en hemsida som tillämpar webbskrapning.

S4: Hur kan webbskrapning förhindras?

Det går inte att hindra webbskrapning av hemsidor om de inte lyder direktiven i robots.txt. Man kan blockera skrapavsändares IP-adresser eller skapa fällor (s.k. spindelfällor) som i vissa fall kan sätta webbskrapare i en oändlig loop som gör att de kraschar. Tyvärr så går det lätt att åtgärda genom att byta IP-adresser eller använda Bot-nät vid webbskrapningsattacker.

S5: Hur kan man sätta upp en pilotstudie som undersöker trafikeffekten av webbskrapning?

Pilotstudien beskrivs i kap. 5.5 förväntas ge en indikation om man kan mäta trafikmängd, hitta vilka webbskrapare som frekvent sökerigenom internet och vilka som följer direktiven i robots.txt.

8 Framtida arbete

Man bör utföra pilotstudien för att utröna om metoden i studien ger det förväntade resultatet. Jurister eller juridikstudenter borde undersöka vilka rättsliga övertramp en webbskrapning kan medföra så att privatpersoner, organisationer och företag kan ta del av sina rättigheter och skyldigheter rörande ämnet.

9 Ordlista

Autentisering	kontrollera identiteten på användare.
Botnet	är ett nätverk av datorer infekterade av datavirus eller trojanska hästar. Dessa datorer ansluter till en central styrande nod där de får uppgifter att utföra.
Captcha	att test som antas vara lätt att lösa för människor, men inte för automatiska datorprogram
Cookies	är en liten textfil som webbläsare använder sig av.
CSS sprites	CSS-sprite är möjligheten att utnyttja olika delar i en bild och minska förfrågningarna mot servern.
Databas	är en samling information som är organiserad på ett sådant sätt att det är lätt att söka efter och hämta information.
DOM	Document Object Model är en specifikation för ett programmeringsgränssnitt från W3C som tillåter att program och skript får uppdatera innehåll, struktur och stil HTML och XML-dokument.
DoS	Denial of Service är en överbelastnings attack.
Mikroformat	Mikroformat är en utökad semantik, den vetenskapliga studiet av språklig betydelse , som åstadkommer i (X)HTML möjligheten att märka upp information.
Server	är ett datorsystem som betjänar andra system, klienter, ofta över ett datornätverk. Server kan syfta på en fysisk dator eller en viss programvara den kör
Sitemap	är en lista över sidor på en webbplats tillgänglig för crawlers eller användare.
Skript	är ett litet program som tillåter kontroll över en eller flera applikationer.
Sprite bilder	en datorgrafisk komponent som kan flyttas runt på skärmen oberoende av annan grafik.
Spamrobotar	robotar som skickar mängder av oönskad e-mail reklam/virus.

10 Tack

Skulle vilja tacka mina mentorer: Martin Boldt, PhD IT-säkerhet, BTH, och David Borin, Konsultchef, Sentor. För all hjälp de har bidragit med.

11 Referenslista

Alfredo Alba, Varun Bhagwan, Tyrone Grandison, Daniel Gruhl, Jan Pieper. 2009. Change Detection and Correction Facilitation for Web Applications and Services. *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 USA.*

Richard Baron Penman, Timothy Baldwin och David Martinez. 2010. Web Scraping Made Simple with SiteScraper. *The University of Melbourne Victoria, Australia.*

Miloslav Beiio, Jakub Misek, Filip Zavoral. 2009. AgentMat: Framework for Data Scraping and Semantization. *Department of Software Engineering Charles University in Prague, Czech Republic.*

William Enck, Patrick Traynor, Patrick McDaniel, och Thomas La Porta. 2005. Exploiting Open Functionality in SMSCapable Cellular Networks. *Systems and Internet Infrastructure Security Laboratory Department of Computer Science and Engineering The Pennsylvania State University University Park.*

Frank Jennings och John Yates. 2009. Scrapping over data: are the data scrapers' days numbered? *Journal of Intellectual Property Law & Practice*.

Nicol´as Poggi, Josep Lluis Berral, Toni Moreno, Ricard Gavald`a och Jordi Torres. 2007. Automatic Detection and Banning of Content Stealing Bots for E-commerce. *Universitat Politecnica de Catalunya Barcelona Spain*.

Ken Moody och Marco Palomino. 2003. SharpSpider: Spidering the Web through Web Services. *Computer Laboratory University of Cambridge Cambridge, CB3 0FD.*

Mark Zadel och Ichiro Fujinaga. 2004. WEB SERVICES FOR MUSIC INFORMATION RETRIEVAL. *Faculty of Music McGill University Montr´eal, QC H3A 1E3.*

Internet referenser

Backendmedia. 2005. Vad skriver man i Robots.txt? http://www.backendmedia.se/2005/12/01/vad-skriver-man-i-robotstxt/ (Hämtad 2012-03-24)

IDG. 2005. Tingsrätten ger Alla Annonser rätt att googla Blocket. http://internetworld.idg.se/2.1006/1.54076 (Hämtad 2012-03-24)

IEEE International Conference on Web Services. 2009. Deactivation of Unwelcomed Deep Web Extraction Services through Random Injection. (Hämtad 2012-03-24)

Lorgor. 2013. LaBrea: "Sticky" Honeypot and IDS. http://labrea.sourceforge.net/labrea-info.html (Hämtad 2013-07-18)

Measurement and Instrumentation. 2011. Theory and Application, sidan 303, Error seeding Mills 1972, ISBN: 9780123819604, 2011-09-26, (Hämtad 2012-03-24)

Robotstxt. 2012. About /robots.txt. http://www.robotstxt.org/robotstxt.html. (Hämtad 2012-03-24)

Sentor. 2011. ASSASSIN. http://www.sentormss.com/assassin.html (Hämtad 2012-03-24)

Södertörns Högskola. 2012 Guide till informationssökning och sökteknik. http://webappo.web.sh.se/p3/ext/content.nsf/aget?openagent&key=guide_till_informationssokning_och_sokteknik_1311062250300 (Hämtad 2012 -12-27)

Wikipedia. 2013. Google. http://sv.wikipedia.org/wiki/Google (Hämtad 2013-07-18).

Wikipedia. 2012. Web scraping. http://en.wikipedia.org/wiki/Web_scraping (Hämtad 2012-03-24)

Wikipedia ref 2. 2012. Robots exclusion standard. http://en.wikipedia.org/wiki/Robots_exclusion_standard (Hämtad 2012-03-24)

Wikipedia ref 3. 2012. Web crawler. http://en.wikipedia.org/wiki/Web_crawler (Hämtad 2012-03-24)

Figurer

Försättsbladet, Google bilder, *webscraping*,
http://www.3idatascraping.com/images/data_scraping.jpg,
Kontrollerad online 2012-03-24.

Figur 1, illustration av Södertörns infirmationsinsamlingsprocess
http://webappl.web.sh.se/p3/ext/content.nsf/aget?openagent&key=guide_till_informationssokning_och_sokteknik_1311062250300

Figur 2, Google bilder, *Semantic annotation recognizing*,
http://www.deg.byu.edu/ding/research/OntologyAssembler.jpg,
Kontrollerad online 2012-03-24.

Figur 3, Google bilder, *robots.txt*,
http://www.technyat.com/wp-content/uploads/2011/06/robots_txt_visual.gif
Kontrollerad online 2012-03-24.

Figur 4-5, Ilustration av Lars Andersson.