Scientific Journal Impact Factor (SJIF): 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Malicious Web Crawler Detection using Intrusion Detection System

Ms. Priyanka Vijay Patankar[1] and Prof. Smita Jangale[2]
[1,2]Dept. of I.T., V.E.S. Institute of Technology, Mumbai India

Abstract— A robust and tensile protected communication and computing environment to enable information to flow ideally with no down time is offered by internet. Web applications gives access to online facilities, gaining information from multiple websites which are also a valuable target for security attacks. The web contains large data and it contains multiple websites which are monitored by a tool or a system known as a crawler. Gathering large data by crossing the limitations of accessing that websites seems to be a malicious attack and gets banned from connecting to the web server. Because of an explosive growth of the intrusion, there is essential requirement of anamoly based intrusion detection system (IDS) which is capable of detecting attacks on system server. The system proves higher performance, higher efficiency and lower maintenance cost, almost all malicious attacks are detected and the malicious codes encoded using C sharp platform. The security to the system is provided by using honeypot which will divert users from that work and the system will alert the server about malicious web crawler so that website can stay secure from them.

Keywords—Web Crawler, Malicious attack Security attacks, Intrusion Detection technique, Honeypot.

I. INTRODUCTION

The World Wide Web consists of loads of web pages with trillions of documents. Web pages increases dynamically with contents added to it. So it is quite difficult to obtain relevant information on the internet that the user has demanded from that particular search engine.

Crawlers behave significantly different from normal users since they are automated programs with pre-defined routines, thus allowing researchers to use fingerprint based techniques to classify them. Per analysis of the behaviors of several commonly seen crawlers and robots, we concluded several commonly seen patterns. By detecting those patterns, we can figure out malicious traffic effectively. By utilizing known HTTP and TCP features, active and passive network sensors can be put in the system to monitor this traffic and with HTTP features as well as TCP features, those traffic can be got rid of from the entire system with little computational resource consumption [3].

A crawler is a program that is used to download and store web pages, mostly for web search engine. A crawler traverses the World Wide Web in a systematic way intending to collect data or knowledge. Web crawlers are also known as web harvesters, robots, or a spider. A web crawler could be a system for the bulk of downloading of websites. A crawler begins placing an initial set of URLs, in a queue, where all URLs to be retrieved are kept and prioritized. The crawler gets a URL in some order from this queue, downloads the page, extracts any URLs within the downloaded page, and then in the queue it puts the new URLs. This whole process is continued. Finally the collected pages are used later for other applications, like for web search engine or a Web cache [1].

To organize the world's information in better way and make it universally accessible by any user, the crawlers are invented. Crawlers traverse against the Internet to fetch information. The purpose of Malicious Web Crawlers is designed for accessing data unlawfully; they lead heavy workload to the websites and reduce performance significantly. At the same time, they can lead to problems in privacy, intellectual property, and illegal economic profit, which have very badly slow down the healthy development of the Internet industry.

The security of web based applications should be addressed by means of careful design and thorough security testing. But unfortunately, this is often not the case. For this concern, security conscious development methodologies should be used by an intrusion detection infrastructure that is able to identify the attacks and provide early warning about suspicious activity occurs. An intrusion detection method has two main types: The one is anomaly detection which is based on finding deviations from normal user behaviour are considered intrusive. The second is misuse detection, it characterized as pattern or signature that IDS looks for. Pattern or signature might be a static string or a set sequence of actions [1].

Honeypot is technology with great potential for the field of network security. It can be understood as a resource used to divert attackers and hackers away from critical resources i.e. it is an observed trap. It can also be used to study an attacker's methods and tools. The value of a Honeypot lies in unauthorized and illicit use. Neither any authorized activity runs on these resources nor do they have any productive value i.e. no legitimate activity is carried out. It provides a large amount of valuable information for analysis and can detect variety of attacks, working even within encrypted environment. There are many advantages of using honeypot over other network security tools. As Honeypot collects data that is related only to attack or unauthorized activity, the data sets are very small but they carry large amount of information in them. The attacker's activities are captured in detail. It logs everything that comes there way including all the new tools and techniques that the attackers use. Honeypot are quite capable of working in the encryption enabled environment. Hence honeypot will also continue to work effectively in future. It is very simple to understand, deploy and maintain. It does not require costly resources and can be setup using cheap systems[4].

To keep information privileged and check automatic crawling programs not behaving normally, system proposed a method of detecting unwanted crawlers.

II. Background of related work

The first Internet "search engine", a tool called "Archie" shortened from "Archives", was developed in 1990 and downloaded the directory listings from specified public anonymous FTP (File Transfer Protocol) sites into local files, around once a month. In 1991, "Gopher" was created, that indexed plain text documents. "Jughead" and "Veronica" programs are helpful to explore the said Gopher indexes. In the year 1993, the first crawler "World WideWebWanderer" was formed. Although this crawler was initially used to measure the size of the Web, it was later used to retrieve URLs that were then stored in a database called "Wandex", the first web search engine. Another early search engine, "Aliweb" (Archie-Like Indexing for the Web) allowed users to submit the URL of a manually constructed index of their site [1].

In the year 1994, the first "full text" crawler and search engine "WebCrawler" was launched. The "WebCrawler" permitted the users to explore the web content of documents rather than the keywords and description written by the web administrators, reducing the possibility of confusing results and allowing better search capabilities. Around this time, commercial search engines being launched from 1994 to 1997. Also introduced in 1994 was Yahoo!, a directory of web sites that was manually maintained, though later incorporating a search engine. During these early years Yahoo! And Altavista maintained the largest market share. In 1998 Google launched, quickly capturing the market. Unlike many of the search engines at the time, Google had a simple, uncluttered interface, unbiased search results that were reasonably relevant, and a lower number of spam results. These last two qualities were due to Google's use of the Page Rank algorithm and the use of anchor term weighting. While early crawlers dealt with relatively small amounts of data, modern crawlers, such as the one used by Google, need to handle a substantially larger volume of data due to the dramatic enhance in the amount of the Web [1].

III. Architecture of web crawler

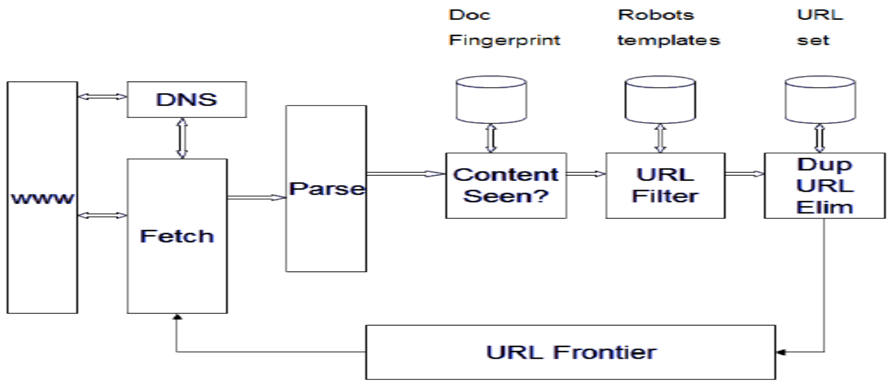

Figure:1 Architecture of Web Crawler

URL Frontier: It contains URLs to be fetched in the current crawl. At first, in URL Frontier a seed set is stored, and by taking a URL from the seed set a crawler begins.
DNS: DNS is domain name service resolution and it looks up the IP address for domain names.
Fetch: It is used to fetch the URL and for that it uses the HTTP protocol.
Parse: It is used to parse the page. In this text, images, videos, etc. and Links are extracted.
Dup URL Elim: Dup URL Elim is used to check the URL for duplicate elimination.
Content Seen?: It is used to test whether a web page with the same content has already been seen at another URL or not. It develops a way to measure the fingerprint of a web page.
URL Filter: It tells whether the extracted URL should be excluded from the frontier (robots.txt) or not. URL should be normalized (relative encoding).

IV. Web Crawler Identification

- **SESSION IDENTIFICATION:**

Session identification is the task of dividing a server access, log into individual web sessions. A web session is a group of activities performed by one individual user from the moment he enters a web site to the moment he leaves it. Session identification is typically performed first by grouping all HTTP requests that originate from the same IP address and the same user-agent, and second by applying a timeout approach to break this grouping into different sub-groups, so that the time-lapse between two consecutive sub-groups are longer than a pre-defined threshold. The key challenge of this method is to determine proper threshold-value, as different Web users exhibit different navigation behaviors. In the majority of web-related literature, 30-min period has been used as the most appropriate maximum session length. Hence, our log analyzer employs the same30-min threshold to distinguish between different sessions launched by the same user[2].

- **FEATURE EXTRACTION:**

The System has adopted different features that are shown to be useful in distinguishing between malicious web crawlers and normal web crawlers. These features are enlisted below[2].

1. **Click number** – The click number metric appears to be useful in detecting the presence of the web crawlers because higher click numbers can only be achieved by an automated script (such as a web robot) and is usually very low for a human visitor.

2. **HTML-to-Image Ratio** – a numerical attribute calculated as the number of HTML page requests over the number of image files (JPEG and PNG) requests sent in a single session.
3. **Percentage of PDF/PS file requests** – a numerical attribute calculated as the percentage of PDF/PS file requests sent in a single session. In contrast to image requests, some crawlers, tend to have a higher percentage of the PDF/PS requests than human visitors.

4. **Percentage of 4xx error responses** – a numerical attribute calculated as the percentage of erroneous HTTP requests sent in a single session.
5. **Percentage of HTTP requests of type HEAD** – a numerical attribute calculated as percentage of requests of HTTP type HEAD sent in a single session. Most web crawlers, in order to reduce the amount of data requested from a site, employ the HEAD method when requesting a web page. On the other hand, requests coming from a human user browsing a web site via browsers are, by default, of type GET.
6. **Percentage of requests with unassigned referrers** – a numerical attribute calculated as the percentage of blank or unassigned referrer fields set by a user in a single session. Most web crawlers initiate HTTP requests with unassigned referrer field, while most browsers provide referrer information by default.

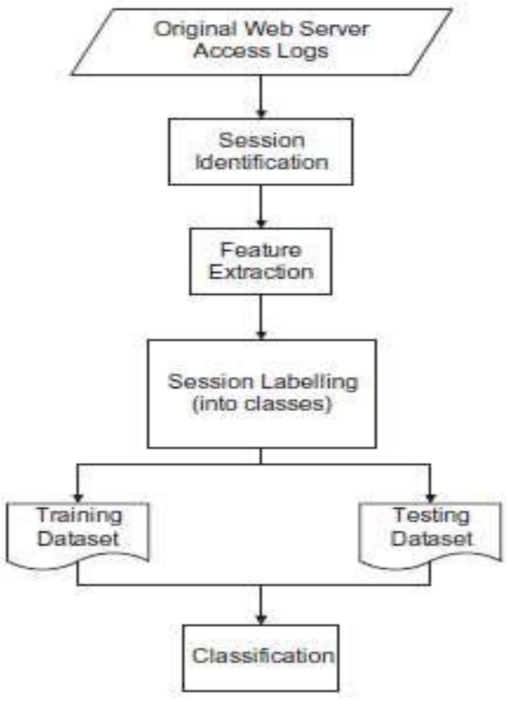

Fig 2. Web server access log-preprocessing

7. **'Robots.txt' file request** – a nominal attribute with values of either 1 or 0, indicating whether 'robots.txt' file was or was not requested by a user during a session, respectively. Web administrators, through the Robots Exclusion Protocol, use a special-format file called robots.txt to indicate to visiting robots which parts of their sites should not be visited by the robot.

Using few of the above features, malicious web crawler can be formed & designed into an Intrusion Detection System of the web server.

- **Log Dataset labelling:**

After the log analyzer parses the log file and extracts the individual visitor sessions, each session (i.e. the respective feature vector) is labelled as belonging to a particular class. Subsequently, 70% of the feature vectors are placed in the training, and 30% of the feature vector into the testing dataset.

V. MALICIOUS WEB CRAWLER APPROACH

Web servers use a special file, robots.txt, to indicate what the acceptable behaviour for robots (spiders/crawlers) visiting the site is. The web crawler scenario checks if robots indexing the contents of the web site hosted by the server adhere to the instructions specified in the robots.txt file. An alarm is raised if any web crawler violates the specified instructions. For this scenario to function, system requires the User-Agent field to be logged; therefore, the server must be configured to log the requests in the Extended Log Format (ELF). The robots.txt file is formatted according to the Robots Exclusion Protocol. This file consists of a set of records in the form <Field> : <Value>. The record starts with one or more User-agent lines, specifying which robots the record applies to, followed by Disallow and Allow instructions to that robot. For example, consider the following records: User-agent: * Disallow: /cyberworld/map/ This example specifies that no robots should visit any URL starting with "/cyberworld/map/", except for the robot called cyber mapper. All robots must obey the first record in /robots.txt that contains a User-agent field whose value contains the name of the robot as a substring. If no such record exists, they should obey the first record with a User agent field with a "*" value, if present. If no record satisfies either condition, or no records are present, then the access is unlimited. The malicious web crawler attack scenario reads the robots.txt file and, for each request, uses the requests URL and User-Agent header field to check whether the crawler is allowed to access the requested URL. If not, a compromised state is reached and a response function is invoked. Because the malicious web crawler scenario is an instance of the generic counting scenario attack template it requires parameters to be specified for the number of invalid robot accesses that are to be considered an attack, the frequency alerts are to be generated during an attack, and the length of the inactivity timeout.

VI. Flow of Proposed System

The system has web server application. It will handle log of web server and log of network traffic. Malicious web crawler attackers sending DDoS attacks will be captured by Intrusion detection System and will give alert to the system about attack quickly.

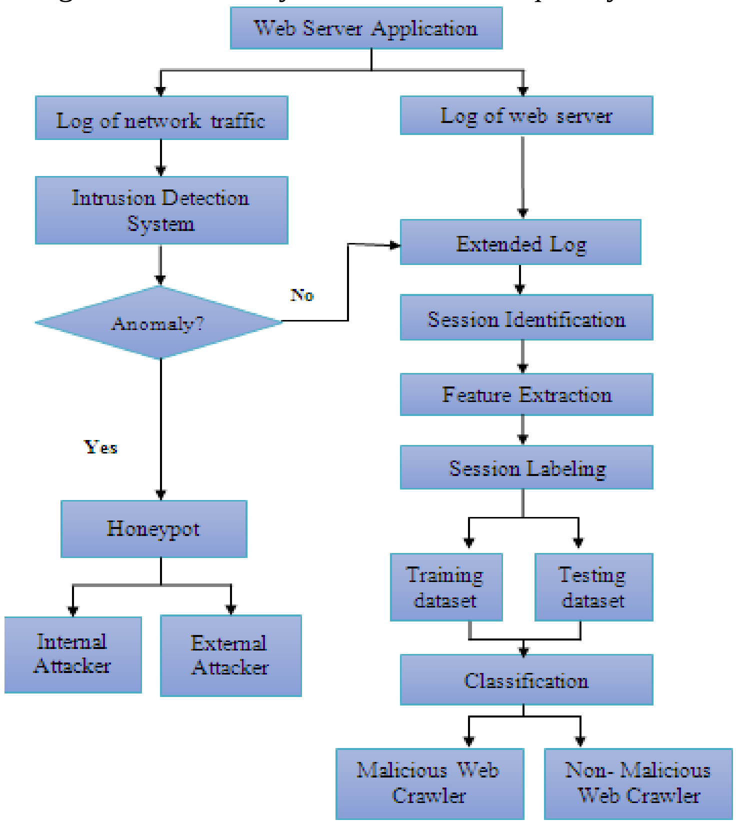

Fig 3. Data flow Diagram (DFD) of Proposed System

Whenever web crawler sends request to web server depending on various signature designed in intrusion detection system web crawler will be classified as normal or malicious web crawler. If web crawler is exceeding their limitations and performing unauthorized tasks then such crawlers are detected malicious then intrusion detection system will send alert to administrator about malicious web crawler and such malicious crawlers are caught by honeypot which will divert their path and will keep the server safe.

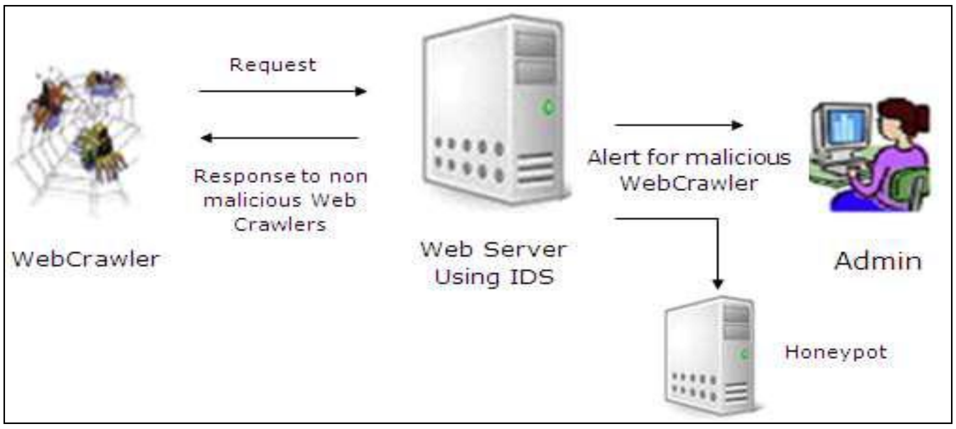
Request
Response to non malicious Web Crawlers
WebCrawler
Web Server Using IDS
Alert for malicious WebCrawler
Admin
Honeypot

Fig 4. Malicious Web Crawler Detection using IDS

It consists of 4 phases:

Phase I- Determining request of Malicious Web crawlers

1. Web crawler sends Request to Web server. Behavior of Web Crawler is registered in Host log file.

2. Web server initially responds to all Web crawlers just to know its behavior.

3. System Administrator will compare host log files & features that distinguish malicious web crawler & non malicious web crawler.

Phase II- Designing Signature based IDS in web server

1. System Administrator will install signature based intrusion detection System in Web server. Signatures in IDS are of malicious web crawlers which can be used to detect malicious web crawlers attacking web server hence forth & it will also alert system administrator.

Phase III – Detection of malicious Web crawler

1. Web crawler sends Request to Web server.
2. Request is checked against signatures of malicious web crawler designed in IDS.
3. If request matches signature then web server sends alert to system administrator.
4. If request does not matches signature then web server respond back to web crawler.

Phase IV – Diversion of Malicious web crawler to honeypot

1. Web crawler sends Request to Web server.
2. Request is checked against signatures and behavior of all web crawlers.
3. If web crawler is performing operations under their limitations then system will give response to them.
4. If web crawler is performing operations beyond their limitations then web server will divert them to honeypot.

VII. Implementation

The detection of web crawlers was evaluated with the following two classifiers: C4.5, SVM. The implementation of each algorithm is performed using c sharp. Each classifier is trained on training dataset and then tested on another supplementary dataset. In the testing phase, the classification results generated by the trained classifiers are compared against the test-data's respective session labels as they have originally been derived by the log analyse.

- **C4.5 Classifier Algorithm:**

The main goal of the C4.5 algorithm is to bring in the split information function which punishes the attributes with large domains. In the C4.5 algorithm split-measure function is projected as a ratio of the information gain and the split information.

- **SVM Classifier Algorithm:**

SVM are often considered as the classifier that makes the greatest accuracy outcomes in text classification issues as well as observed as cutting-edge models for binary classification of very high dimensional data. First, index the term in ascending order. Then, all the terms are weighted according to its features. If the score of weighting is greater than zero (weight>O), the term is classified as benign web page. Otherwise, the term is classified as malicious web page.

VIII. RESULTS

- **The Experimental results and Comparison between classification algorithm:**

Number of Log Files loaded	Number of Normal users searched (a)	Number of Malicious Crawlers (b)	Total output (a + b)
582	182	85	267
1050	198	134	332
2055	363	239	602
3060	528	344	872

Table 1 : Experimental results using C4.5 Classification Algorithm

Number of Log Files loaded	Number of Normal users searched (a)	Number of Malicious Crawlers (b)	Total output (a + b)
582	437	145	582
1050	850	200	1050
2055	1735	320	2055
3060	2620	440	3060

Table 2: Experimental results using SVM Classification Algorithm

Log Files No.	Difference between Existing And Improved Algorithm	Existing C4.5 Algorithm (%)	Improved SVM Algorithm (%)	Percentage Difference in Accuracy (%)
582	315	45.87	100	8.25
1050	718	31.61	100	30.92
2055	1453	29.38	100	70.71
3060	2188	28.49	100	71.50

Table 3: Comparison between C4.5 Classification Algorithm SVM Classification Algorithms

Using Standard Formula:

Accuracy improved = ((No. of count with threshold)-(No of count without threshold))/(No. of count without threshold) * 100

- **The Graphical Comparison between Classification Algorithm**

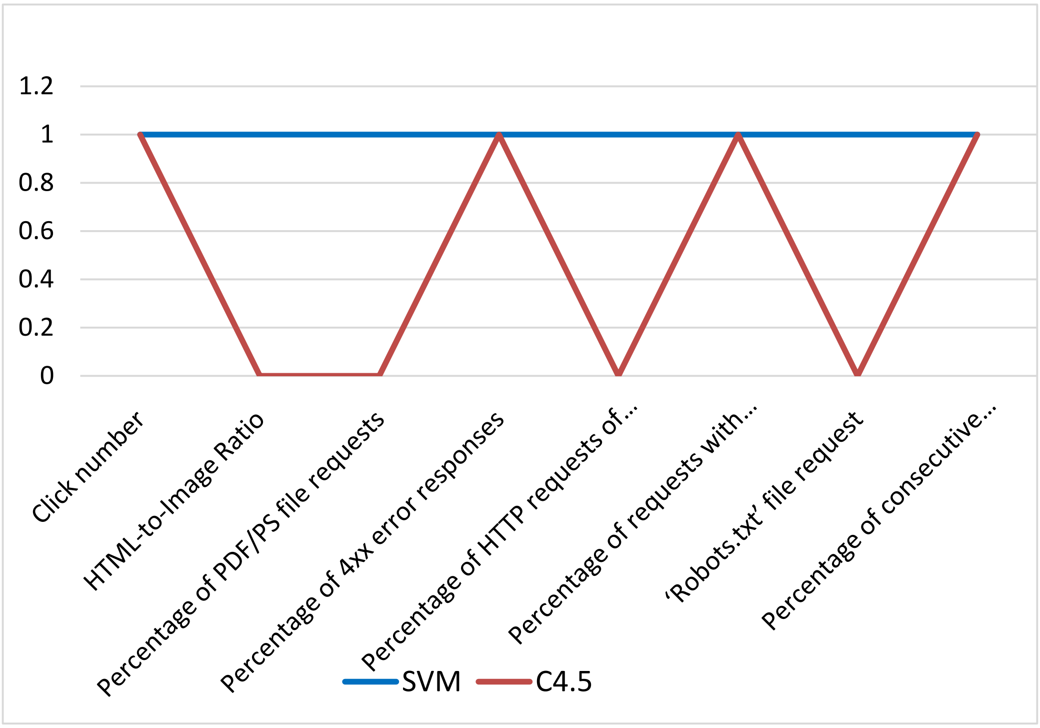

Fig 5. The feature evaluation between two classification algorithms

IX. Conclusion

Web Crawler is information retrieval which traverses the Web and downloads web documents that suit the user's need, eventhough there are various methods approached to detect malicious behavior of web crawler. There is scope to improve the performance. An Intrusion Detection System is a new approach to detect a malicious web crawler and identify them easily before going to the training set so that we can analyze attacker's behavior and a security to the system is given by using honeypot for it. The results are calculated on the basis of C4.5 classification algorithm and SVM classification algorithm. Using threshold over the existing algorithm of SVM we found good results compare to basic algorithms. The experimentation results are quite outstanding in predicting the attacks. The identification of normal users and malicious users with respective parameters of classification algorithm is evaluated and analyzed.

X. Future Work

Our system is efficient in predicting and visualizing potential multistage attacks. Our stem is basic but effective; there is a room to improve the performance of the system by modifying the algorithms we applied. Now that we have developed a system that can predict multistage attacks, our aim is to improve the performance of the system, to add more features to find novel attacks.

References

[1] V. S. Dhaka, Sanjeev Kumar Singh " Web Crawler: A Review", International Journal of Computer Applications (0975 – 8887) Volume 63– No.2, February 2013.
[2] Dusan Stevanovic, AijunAn, NatalijaVlajic "Feature evaluation for web crawler detection with data mining techniques", @ 2012 Elsevier Ltd. All rights reserved.
[3] Gen Hattori, Kazunori Matsumoto, Chihiro Ono and Yasuhiro Takishima "Identification of Malicious Web Pages for Crawling Based on Network-Related Attributes of Web Server"; 978-1-4244-7820-0/10/$26.00 ©2010 IEEE
[4] Ioannis Avraam, Ioannis Anagnostopoulos "A Comparison over Focused Web Crawling Strategies", 2010.