A Review on Enhanced Technique for Detection of Malicious Web Crawler

D. Santhosh[1], R. Seetha[2], T. Sumithra[3], S. Yoga Archana[4], M. Saranya[5]

[1, 2, 3, 4] Student, Department of Computer Science and Engineering, KSRCE, TamilNadu, India.
Email: santhoshdj252@gmail.com[1], seetha71098@gmail.com[2], smilesumi11@gmail.com[3], auchurashi5612@gmail.com[4]

[5]Assistant Professor, Department of Computer Science and Engineering, KSRCE, TamilNadu, India.
Email: saranyam1295@gmail.com[5]

Abstract - The utilization of web has colossally expanded everywhere throughout the world. The Web offers a strong and adaptable secured correspondence and registering condition to empower data to stream preferably with no vacation. Web applications give access to online administrations, picking up data from different destinations and are additionally an important objective for security assaults. The web contains immense information and it contains numerous sites which are observed by an instrument or a program known as a crawler. Gathering gigantic information by intersection the impediments of getting to that ***site*** is by all accounts a malignant assault and will be restricted from interfacing with the web server. As a result of a dangerous development of the interruption, need of oddity based interruption identification framework (IDS) which is fit for distinguishing assaults on server, is essential. Honeypot will be utilized for identified abnormalities to protect server. Further the malignant crawler recognized by the framework will send caution to the server about malevolent web crawler with the goal that server can remain alert.

Keywords - Crawler, Malicious attack, Security attacks, Anomaly.

I. INTRODUCTION

Today's the world is critically dependent on the internet. The World Wide Web is internet client-server architecture and such an authoritative system based on complete independence to the server for serving information available on the internet. Information over the internet is in spread and non-linear text system known as Hypertext Document System Search engines used by internet browsers to explore the servers for necessary pages of information. Servers proceed this pages to the clients. The growth of the internet has altered the way traditional necessary services of daily life. Crawlers behave radically different from normal users since they are automated programs with pre-defined routines, thus allowing researchers to use fingerprint based technique to classify them. Per analysis of the behaviors of numerous commonly seen crawlers and robots, we accomplished several commonly seen patterns. By detecting those patterns, we can figure out malicious traffic efficiently. By utilize 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 whole system with little computational resource consumption[9]. A crawler is a program that is used to download and store up web pages, mostly for web search engine. A crawler traverses the World Wide Web in a systematic way intending to gather data or knowledge. Web crawlers also known as web harvesters, robots or a spider. A web crawler could be a system for the immensity of downloading of websites. A crawler begins placing an initial set of URLs, in a queue, where all URLs to be retrieve are kept and prioritized. The crawler gets a URL in several order from this queue, downloads the page, extract any URLs within the downloaded page, and then in the queue it put the new URLs. This whole process is continued. Finally the collected pages are used later for other application, like for web search engine or a Web cache [1]. To better organize the world's information and make it generally accessible, crawlers are invented to traverse against the Internet to fetch information. The purpose of Malicious web crawlers is designed for accessing data unlawfully; they bring heavy workload to the websites and decrease performance considerably. At the same time, they can bring troubles 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 watchful design and through security testing. But unfortunately, this is often not the case. For this concern, security conscious development methodologies be supposed to 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 major types: The one is anomaly detection. This is based on finding deviations from regular user behavior are considered intrusive. The next is misuse detection; it's characterize as a pattern or signature that IDS look for [3].

II. RELATEDWORK

The index enclosed a list of URLs and a list of users wrote keywords and descriptions. The network slide of crawlers initially caused much disagreement, but this trouble was determined in 1994 with the introduction of the Robots Exclusion Standard which allowed web site administrators to block crawlers from retrieve part or all of their sites. In 1994, "WebCrawler" was launched the initial "full text" crawler and search engine. The "WebCrawler" acceptable the users to discover the web content of documents fairly than the keywords and description written by the web administrators, reducing the chance of confusing results and allow better search capabilities. About this time, commercial search engines creature launched from 1994 to 1997. Also introduced in 1994 was Yahoo!, a directory of web sites that was physically maintained, though later incorporating a search engine. During these early years Yahoo! And AltaVista maintained the biggest market share. In 1998 Google launched, rapidly capturing the market. Dissimilar many of the search engines at the time, Google had a simple, uncluttered interface, unbiased search results that were reasonably correlated, and a minor number of spam outcome. These last two characters were due to Google's use of the Page Rank algorithm and the use of anchor term weighting. While early crawlers dealt with comparatively small amounts of data, modern crawlers, such as the one used by Google, need to handle a considerably larger volume of data due to the dramatic increase in the amount of the Web [1]. Some techniques which are meant for detection of web application related attacks and their advantages and disadvantages are presented. Various IDS tools obtainable for network application protection; like SNORT, OSSEC, SQUIL, OSSIM, TRIPWIRE are discussed. In this analysis, it is inferred that the data difficulty of application has been improved, the web application adapted to multi-tier design [7]. A new model and architecture of the WebCrawler via multiple HTTP relations to WWW is presented. The multiple HTTP connection is applied using multiple threads and asynchronous Downloader part so that the overall downloading process is optimized. The client gives the initial URL from the GUI provided. It begins with a URL to visit. As the crawler visit the URL, it identifies all the hyperlinks obtainable in the web page and append them to the list of URLs to visit, known as the crawl frontier. URLs from the frontier is iteratively visited and it ends when it reach more than five levels from every home page of the websites visited and it is proficient that it is not required to go deeper than five levels from the home page to capture most of the pages visited by the people while trying to recover information from the internet [10].

III.CRAWLER PATTERN ANALYSIS

The majority crawlers are not scripted knowledge and are simply traversing against all links found in a page with a fixed interval. For those crawlers, the following patterns and are amazingly high performing in detection [9].

(1) Continuous Requests: Many crawlers are programmed to parse an entry page, remove links in the entry page and visit each link instantly or after a fixed or arbitrary interval. For robots, in order to get the whole site as fast as possible, the interval is likely to be short. Despite of the interval, in the access log, we can observe resultant and continuous requests. By defining an adequate threshold of visiting the site, we can figure out probable crawlers.

(2) Not Accepting Cookies: While HTTP is stateless, to keep the state of the user, cookies are used. conversely, due to the nature of crawlers which is stateless, it does not keep cookies sent from the server. Thus, requests from the same or similar (in the same C class) IP address which never send cookie information can be very doubtful.

(3) Bogus User Agents: User cannot access the Internet frankly. As an alternative, users use User Agents. The majority commonly seen user agent is a web browser. All user agents use a user agent string to recognize itself. All browsers will send out User Agent information. However, a lot of crawlers are omitting user agents; others are simply identify themselves as crawlers or very old browsers including Internet Explorer 3.0 running on Windows 95 or Netscape4.78 on Solaris. Since those old browsers are not proficient for the present Internet, we can safely define a blacklist of user negotiator or even use machine learning algorithms to robotically generate a white list.

(4) Not Loading/Executing Scripts: Opposed to web browsers which has incorporated scripting engine (mostly ECMA Script interpretation engine, whether fully purposeful and complying with standards or not), spiders are not outfitted with scripting engines in most cases for simpler implementation and faster carrying out. Thus, by putting pitfall and triggers in the source code, we may be able to implement traps for web spiders and preset bots. However, considering the insecurity nature of the Internet, thresholds should be set and timeouts should be available.

(5) High Fetch Rates: Another ordinary approach in implementing web spiders and crawlers is to fetch pages as fast as probable. However, normal users lean to load several pages at a time, read the pages and load another batch of pages after a comparatively long period[9].

IV. CRAWLING POLICIES

Huge volume and rate of modify are two important characteristics of the web that produce a scenario in which web crawling is very important. Also, network speed has enhanced less than current processing speeds and storage capacity. The large volume imply that the crawler can only download a portion of the Web pages within a given time, so it needs to prioritize its downloads. The high rate of change implies that by the time the crawler is downloading the final pages from a site, it is very probable that new page have been added to the site, or pages that have previously been reorganized or even deleted. A crawler must suspiciously choose at every step which pages to visit next. Web crawler actions is the result of a combination of policies. There are four policies:

1. A Selection policy: It decide which page to download. Designing a fine selection policy has an added complexity: it must work with biased information, as the entire set of Web pages is not known during crawling.

2. A Re-visit policy: It decides when to check for changes to the pages. There are two plain re-visiting policies:

3. Uniform policy: All pages in the group with the same frequency are re-visited.

4. Proportional policy: The pages that alter more often are re-visited. The visiting frequency is directly proportional to the (estimated) vary frequency. In mutual cases, the repeated crawling order of pages can be done either at casual or with a unchanging order.

5. Optimal re-visiting policy: It is neither the uniform policy, nor the relative policy. The best method for keeping average freshness high includes ignoring the pages that change too frequently, and the optimal for keeping middling age low is to use access frequencies that monotonically (and sub-linearly) enlarge with the rate of change of each page.

6. A Politeness policy: It decides how to stay away from overloading websites.

7. A Parallelization policy: It decides how to coordinate dispersed web crawlers. A parallel crawler is a crawler that runs numerous processes in parallel. The objective is to exploit the download rate while minimize the transparency from parallelization and to avoid repeated downloads of the same page. To stay away from downloading the same page more than once, the crawling system require a policy for handing over the new URLs bare during the crawling process, as the same URL can be found by two dissimilar crawling processes.

V.CRAWLING TECHNIQUES

There are a small number of crawling techniques used by Web Crawlers, mainly used are:

A. General Purpose Crawling: A common purpose Web Crawler collect as many pages as it can from a exacting set of URL's and their links. In this, the crawler is able to obtain a large number of pages from dissimilar locations. General purpose crawling can deliberate down the speed and network bandwidth because it is fetching all the pages.

B. Focused Crawling: A focused crawler is designed to collect documents only on a exact topic which can decrease the amount of network traffic and downloads. The intention of the focused crawler is to selectively look for pages that are appropriate to a pre-defined set of matters. It crawls only the relevant.

C. regions of the web and leads to significant

D. savings in hardware and network resources.

E. Distributed Crawling: In distributed crawling, multiple processes is used to crawl and download pages from the Web.

VI.ARCHITECTURE OF WEBCRAWLER

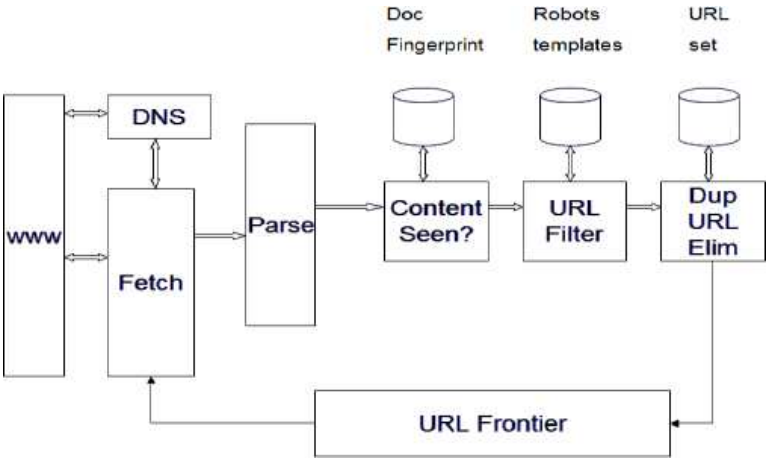

Fig. 1.Architecture of Web Crawler

URL Frontier: It contains URLs to be fetched in the present crawl. At first, 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 look up the IP address for domain names.

Fetch: It is used to obtain 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.

Content Seen : It is used to test whether a web page with the same substance has previously been seen at another URL or not. It develops a way to compute the fingerprint of a web page.

URL Filter : It tells whether the extracted URL should be expelled from the frontier (robots.txt) or not. URL should be normalized (relative encoding).

Dup URL Elim: Dup URL Elim is used to verify the URL for duplicate elimination.

VII.PROCESS OF CRAWLING

The vital working of a web-crawler can be summarized as follows [4]:

- Select a initial seed URL or URLs
- Add it to the dealing out queue
- Now choose the URL from the Processing queue
- Obtain the webpage related to that URL
- Parse that webpage to find new URL links
- Add all the recently found URLs into the Processing queue

Go to step (2) and repeat while the Processing queue is not vacant [5].

VIII.WEB CRAWLER IDENTIFICATION

Web crawlers naturally discover themselves to a Web server by using the User-agent field of an HTTP request. Web site administrators classically observe their Web servers' log and use the user agent field to conclude which crawlers have visited the web server and how often

A) Log Dataset preparation

Supervised data-mining algorithms need pre-labeled training sample in order to learn (I.e. Build) a classification model for a exacting dataset. In this section we give a concise overview of our log analyzer that has been used to generate a workable dataset– comprising both training and testing data samples – from any given web-log file. The process of the log analyzer is carried out in three stages: (1) session identification, (2) features extraction for each identified session, and (3) session labeling (See Fig.3)

B) Session identification:

Session identification is the assignment of dividing a server access, log into individual web sessions. A web session is a collection 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 combination all HTTP requests that originate from the same IP address and the same user-agent, and second by applying a timeout approach to crack this grouping into unlike 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 conclude 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 suitable maximum session length. Hence, our log analyzer employs the same 30-min threshold to differentiate between different sessions launched by the same user[2].

C) Feature extraction:

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

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

2. HTML-to-Image Ratio – A arithmetical attribute calculated as the number of HTML page requests in excess of the number of image files (JPEG and PNG) wishes sent in a single session.

3. Percentage of PDF/PS file requests – a arithmetic attribute calculated as the percentage of PDF/PS file requests sent in a single session. In difference to image requests, some crawlers, lean 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 arithmetical attribute calculated as percentage of requests of HTTP type HEAD sent in a single session. Most web crawlers, in order to decrease the amount of data requested from a site, employ the HEAD method when requesting a web page. On the further 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 arithmetical attribute calculated as the percentage of blank or unassigned referrer fields set by a user in a single session. Most web crawlers start HTTP requests with unassigned referrer field, while most browsers provide referrer information by default.

7. 'Robots.txt' file request – A insignificant attribute with values of either 1 or 0, representative 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 specify 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 created.

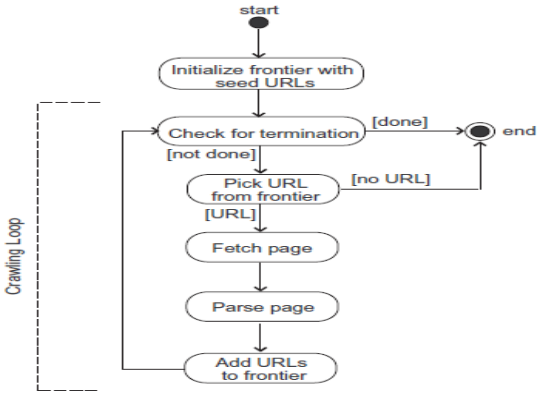

Fig. 2.Working of a web-crawler

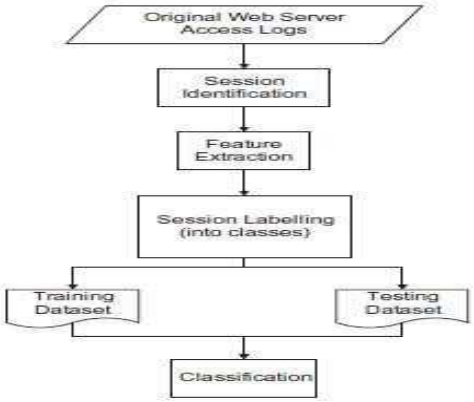

Fig 3. Web server access log-preprocessing

Log Dataset labeling:

After the log analyzers parses the log file and extract the entity visitor sessions, each session (i.e. the respective feature vector) is labeled as belonging to a exacting class. Consequently, 70% of the feature vectors are placed in the training, and 30% of the characteristic vector into the testing dataset.

IX.THE PROPOSED SYSTEM

n the proposed system, there will be a web server application. The web server application will have an intrusion detection system which is deliberate to discover malicious web crawler. A Data flow diagram of the system is shown in figure 4. Malicious Web Crawler Detection using IDS diagram is revealed in figure 5.

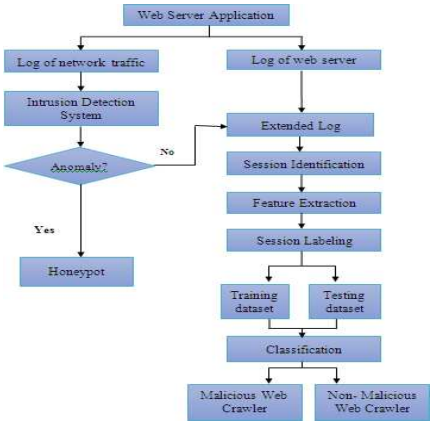

Fig 4. Data Flow Diagram of Malicious Web Crawler Detection using IDS

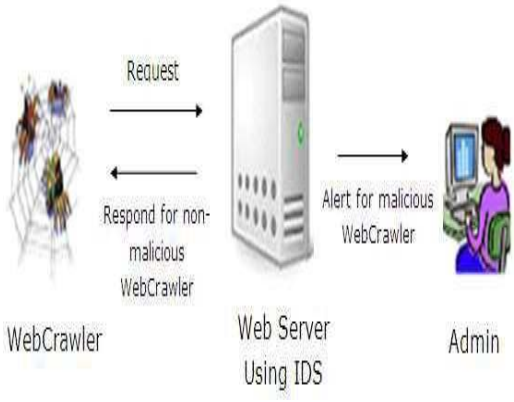

Fig 5. Malicious Web Crawler Detection using IDS

Whenever web crawler sends request to web server, Intrusion Detection System will identify the suspicious anomaly and will send it to honey pot to keep server secure. If IDS couldn't detect some of the internal anomaly, those crawlers

will be send to extended log and that web crawler will be examined and classified as usual or malicious web crawler. If web crawler is detected malicious then system will send alerts to the administrator concerning the malicious web crawler. Even though there are various methods approached to identify malicious web crawler; they are quite difficult to handle. An Intrusion Detection System is a new approach to detect a malicious web crawler and identify them simply. And convention of Honey pot is new approach to Malicious Web Crawler Detection.

X.CONCLUSION

Web Crawler is information rescue which traverse the Web and downloads web documents that suit the user's need. Crawlers are fundamentally used to create a replica of all the visited pages, which are afterward processed by a search engine that will index the downloaded pages that help in quick searches.

REFERENCES

[1] Namrata H.S Bamrah, B.S. Satpute, PramodPatil " Web Forum Crawling Techniques ", International Journal of Computer Applications(0975–8887) Volume 85 – No 17, January 2014.

[2] N. Sakthipriya, K. Palanivel "Intrusion Detection for Web Application: An Analysis"; International Journal of Scientific & Engineering Research, Volume 4, Issue 5,May-2013.

[3] V. S. Dhaka, Sanjeev Kumar Singh "Web Crawler: A Review", International Journal of Computer Applications (0975 – 8887) Volume 63–No.2, February2013.

[4] DeXiang Zhang, DiFan Zhang and Xun Liu, "A Novel Malicious Web Crawler Detector: Performance and Evaluation"; IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January2013.

[5] N. Sakthipriya, K. Palanivel "Intrusion Detection for Web Application: An Analysis"; International Journal of Scientific & Engineering Research, Volume 4, Issue 5,May-2013.

[6] DusanStevanovic, AijunAn, NatalijaVlajic "Feature evaluation for web crawler detection with data mining techniques", 2012 Elsevier Ltd. All rights reserved.

[7] DusanStevanovic, NatalijaVlajic, AijunAn"Unsupervised Clustering of Web Sessions to Detect Malicious and Non-malicious Website Users"; The 2nd International Conference on Ambient Systems, Networks and Technologies2011.

[8] RajashreeShettar, Dr. Shobha G, "Web Crawler On Client Machine", Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol II IMECS 2008, 19-21 March,2008.

[9] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, "Deriving Marketing Intelligence from Online Discussion," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 419-428, 2005.

[10] Gautam Pant, PadminiSrinivasan, and FilippoMenczer, " Crawling the web", Springer2004.