ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

A Review on Enhanced Technique for Detection of Malicious Web Crawler

Priyanka Patankar, Prof. Smita Jangle

PG Student, Dept. of I.T., V.E.S. Institute of Technology, Mumbai, India

Associate Professor, Dept. of I.T., V.E.S. Institute of Technology, Mumbai, India

ABSTRACT: The use of internet has tremendously increased all over the world. The Internet offers a robust and flexible protected communication and computing environment to enable information to flow ideally with no down time. Web applications provide access to online services, gaining information from various sites and are also a valuable target for security attacks. The web contains huge data and it contains many websites which are monitored by a tool or a program known as a crawler. Collecting huge data by crossing the limitations of accessing that website seems to be a malicious attack and will be banned from connecting to the web server. Because of an explosive growth of the intrusion, necessity of anomaly based intrusion detection system (IDS) which is capable of detecting attacks on server, is necessary. Honeypot will be used for detected anomalies to keep server safe. Further the malicious crawler detected by the system will send alert to the server about malicious web crawler so that server can stay 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 a powerful system based on complete autonomy to the server for serving information available on the internet. Information over the internet is in distributed and non-linear text system known as Hypertext Document System Search engines used by internet browsers to explore the servers for required pages of information. Servers proceed this pages to the clients. The growth of the internet has transformed the way traditional necessary services of everyday life. 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 [9].

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 better organize the world's information and make it universally 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 reduce performance significantly. At the same time, they can bring 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,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

which is based on finding deviations from normal user behavior are considered intrusive. The second is misuse detection, it's characterized as a pattern or signature that IDS looks for [3].

II. 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 "World WideWebWanderer" was formed the first crawler. 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].

The index contained a list of URLs and a list of users wrote keywords and descriptions. The network overhead of crawlers initially caused much controversy, but this issue was resolved in 1994 with the introduction of the Robots Exclusion Standard which allowed web site administrators to block crawlers from retrieving part or all of their sites. Also, in the year 1994, "WebCrawler" was launched the first "full text" crawler and search engine. 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].

Several techniques which are meant for detection of web application related attacks and their advantages and disadvantages are presented. Various IDS tools available for network application protection; like SNORT, OSSEC, SQUIL, OSSIM, TRIPWIRE are discussed. In this analysis, it is inferred that the data complexity of application has been increased, the web application adapted to multi-tier design [7].

A new model and architecture of the WebCrawler using multiple HTTP connections 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 user gives the initial URL from the GUI provided. It begins with a URL to visit. As the crawler visits the URL, it identifies all the hyperlinks available in the web page and appends 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 reaches more than five levels from every home page of the websites visited and it is accomplished 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 retrieve information from the internet [10].

III. CRAWLER PATTERN ANALYSIS

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

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

(2) Not Accepting Cookies: Since HTTP is stateless, to keep the state of the user, cookies are used. However, 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 suspicious.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

(3) Bogus User Agents: Users cannot access the Internet directly. Instead, users use User Agents. Most commonly seen user agent is a web browser. All user agents use a user agent string to identify itself. All browsers will send out User Agent information. However, many crawlers are omitting user agents; others are simply identifying 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 capable for the current Internet, we can safely define a blacklist of user agent or even use machine learning algorithms to automatically generate a whitelist.

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

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

IV. CRAWLING POLICIES

Large volume and rate of change are two important characteristics of the web that generate a scenario in which web crawling is very important. Also, network speed has improved less than current processing speeds and storage capacities. The large volume implies that the crawler can only download a fraction 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 last pages from a site, it is very likely that new pages have been added to the site, or pages that have already been updated or even deleted. A crawler must carefully choose at each step which pages to visit next. Web crawler behavior is the result of a combination of policies. There are four policies:

1. **A Selection policy:** It decides which pages to download. Designing a good selection policy has an added difficulty: it must work with partial information, as the complete 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 simple re-visiting policies:

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

4. **Proportional policy:** The pages that change more frequently are re-visited. The visiting frequency is directly proportional to the (estimated) change frequency. In both cases, the repeated crawling order of pages can be done either at random or with a fixed order.

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

6. **A Politeness policy:** It decides how to avoid overloading websites.

7. **A Parallelization policy:** It decides how to coordinate distributed web crawlers. A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes.

V. CRAWLING TECHNIQUES

There are a few crawling techniques used by Web Crawlers, mainly used are:

A**. General Purpose Crawling:** A general purpose Web Crawler collects as many pages as it can from a particular set of URL's and their links. In this, the crawler is able to fetch a large number of pages from different locations. General purpose crawling can slow down the speed and network bandwidth because it is fetching all the pages.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

B. Focused Crawling: A focused crawler is designed to collect documents only on a specific topic which can reduce the amount of network traffic and downloads. The purpose 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 regions of the web and leads to significant savings in hardware and network resources.

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

VI. ARCHITECTURE OF WEB CRAWLER

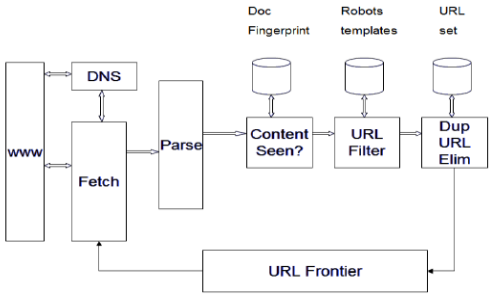

Fig. 1. Architecture of Web Crawler

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 look 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.

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).

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

a. PROCESS OF CRAWLING

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

- Select a starting seed URL or URLs
- Add it to the Processing queue
- Now pick the URL from the Processing queue
- Fetch the webpage corresponding to that URL
- Parse that webpage to find new URL links
- Add all the newly found URLs into the Processing queue

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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

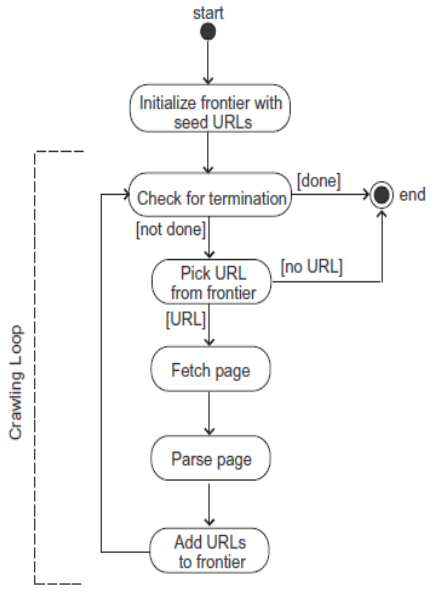

Fig. 2. Working of a web-crawler

b. WEB CRAWLER IDENTIFICATION

Web crawlers typically identify themselves to a Web server by using the User-agent field of an HTTP request. Web site administrators typically examine their Web servers' log and use the user agent field to determine which crawlers have visited the web server and how often.

Log Dataset preparation:

Supervised data-mining algorithms require pre-labelled training samples in order to learn (I.e. Build) a classification model for a particular dataset. In this section we give a brief 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 operation 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)

- 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 same 30-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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

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.

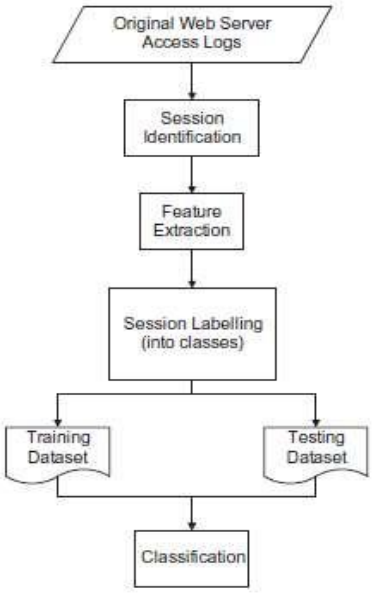

Fig 3. Web server access log-preprocessing

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.

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.

Log Dataset labelling:

After the log analyser 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

VII. THE PROPOSED SYSTEM

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

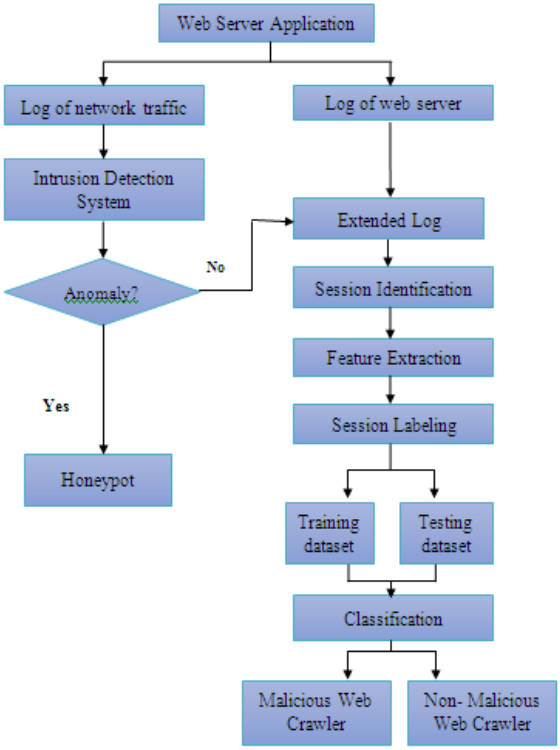

Fig 4: Data Flow Diagram of Malicious Web Crawler Detection using IDS

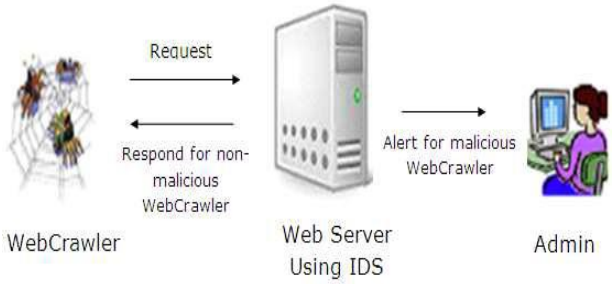

Fig 5. Malicious Web Crawler Detection using IDS

Whenever web crawler sends request to web server, Intrusion Detection System will detect the suspicious anomaly and will send it to honeypot to keep server secure. If IDS couldn't detect some of the internal anamoly, those crawlers will be send to extended log and that web crawler will be examined and classified as normal or malicious web crawler. If web crawler is detected malicious then system will send alerts to the administrator about the malicious web crawler.

VIII. CONCLUSION

Even though there are various methods approached to detect malicious web crawler; they are quite complex to handle. An Intrusion Detection System is a new approach to detect a malicious web crawler and identify them easily. And usage of Honeypot is new approach to Malicious web crawler detection.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

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

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, Aijun An, Natalija Vlajic "Feature evaluation for web crawler detection with data mining techniques", 2012 Elsevier Ltd. All rights reserved.
[3] N. Sakthipriya, K. Palanivel "Intrusion Detection for Web Application: An Analysis"; International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013.
[4] Gautam Pant, Padmini Srinivasan, and Filippo Menczer, " Crawling the web", Springer 2004.
[5] Namrata H.S Bamrah, B.S. Satpute, Pramod Patil " Web Forum Crawling Techniques ", International Journal of Computer Applications (0975 – 8887) Volume 85 – No 17, January 2014.
[6] 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 Technologies 2011.
[7] N. Sakthipriya, K. Palanivel "Intrusion Detection for Web Application: An Analysis"; International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013.
[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] 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, January 2013.
[10] 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.

BIOGRAPHY

Priyanka Vijay Patankar is a Junior Research Assistant in the Information Technology Department, from V.E.S. Institute of Technology, Mumbai University. She received Bachelor of Technology (B.Tech I.T.) degree in 2013 from S.N.D.T. University, Mumbai, India. Her research interests are Computer Networks (wireless Networks).