Crawl Analyzer: Analysis of Website and Security Certificates

KRISHNA SAINI[1], DR. ANURADHA KONIDENA[2]

[1] *Student, B. Tech (CSE),4th year, College of Engineering and Technology, IILM Academy of Higher Learning, Greater Noida, Uttar Pradesh, India*

[2] *Associate Professor, Department of CSE, College of Engineering and Technology, IILM Academy of Higher Learning, Greater Noida, Uttar Pradesh, India*

Abstract— Crawl Anlizer is the combination of two words Craw and Analyzer. There are millions of cyber-attacks happen all around the globe every day. This program will Craw the domain around the globe and analysis the security protocols which are needed to enhance the security of the website and protect the privacy of the users.

Indexed Terms-- Security, Privacy, WebCrawler, Tranko, Kibana

I. INTRODUCTION

In today's digital Era everyone is surfing on the World Wide Web (WWW). While using the Website, the user makes their digital identity over the website which helps the website to recognize the user. These ids are over the internet and could be accessed by anyone. To make sure these ids are been used by the authentic user there are many security protocols and certificates (metrics) that are been deployed on the websites to make a secure connection with a user. If those security metrics are not available on the website Cyber Criminals can get that ID and could use it to breach your private data and could cost you or your company damage in both the cyber world and physical world. To make sure that the top 1 million sites which are been accessed by people across the globe are secure enough to surf on them.

II. LITERATURE REVIEW

A. *Information Security*

Information Security is about the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording, or destruction of information. Information can be a physical or electronic one. Information can be anything like Your details or we can say your profile on social media, your data on mobile phone, your biometrics, etc. This Information Security covers many research areas like Cryptography, Mobile Computing, Cyber Forensics, etc.

B. *Types of Web Crawlers*

1. Breadth-First Crawler: it starts with a small set of pages and then explores other pages by following the link in the breadth first fashion [1]. Actually, web pages are not traversing strictly in the breadth first fashion they have a variety of policies that help them to decide which one to go first. For example, having a pile of the pages may go for the most important page first.

2. Incremental Web Crawler: It updates an existing set of downloaded pages instead of restarting from the beginning every time [2]. This involves some ways for determining if the page has changed since the last time it was crawled. A crawled will continuously crawl the web-based on its crawling cycles. An adaptive model is used, which will decide which page will be checked on the basis of previous data sets, thus highly updated results in low load are achieved.

3. From Focused Crawler: from focused crawlers handles spare distribution of forms on the Web. From crawler [3] avoids crawling through unproductive paths by limiting the search to a particular topic like learning links and the paths for the page contain searchable forms.

4. Focused Crawler: Focused crawler is predicated on the hypertext classifier which was developed by Chakrabarti et al [4, 5]. A focused crawler has three main components: a classifier that makes relevance judgments on pages crawled to decide on link expansion, a distiller which determines a measure of centrality of crawled pages to work out visit priorities, and a crawler with dynamically reconfigurable priority controls which is governed by the classifier and distiller. Its aim is to provide an easy alternative to overcome the difficulty which immediate pages that are ranked low associated with the topic at hand. The idea is to recursively execute an exhaustive search up to a given depth, starting from the relatives' of a highly ranked page

5. Hidden Web Crawler: A lot of data on the web actually resides within the database and it can only be retrieved by posting appropriate queries or by filling out forms online. Recently interest has been focused on access to this type of knowledge called "deep web" or "hidden web". Nowadays crawlers' crawl only publicly indexable web (PIW)

6. Parallel Crawler: As the size of the Internet grows, it becomes harder to retrieve the entire or a big portion of the internet by employing a single process. Therefore, many search engines often run multiple processes in parallel to perform the above task, in order that download rate is maximized. this sort of crawler is understood as a parallel crawler

7. Distributed Web Crawler: Distributed web crawler works on a network of workstations. Indexing the web may be a very challenging task thanks to the growing and dynamic nature of the online. Since the size of the web is expanding it becomes mandatory to parallelize the method of crawling to complete the crawling process in a very decent amount of time. a single crawling process even with multithreading is going to be insufficient for the case. in this case the method has to be distributed to multiple processes to create the method scalably. It scales up to several thousand pages per second. the speed at which the size of the web is growing it's imperative to parallelize the method of crawling. In distributed web crawler a URL server distributes individual URLs to multiple crawlers, which download web content in parallel. The crawlers then forward the downloaded pages to a central indexer on which links are extracted and shared via the URL server to the crawlers. This distributed nature of the crawling process reduces the hardware requirements and increases the overall download speed and reliability [6]. FAST Crawler [7] could be a distributed crawler, utilized by Fast Search & Transfer.

C. Desired Features of Crawlers

1. Speed: Since HTTP request takes one second to complete-some will take much longer to fail to respond at all. A simple crawler cannot fletch more than 86,400 pages per day, with this rate it will take 634 years to crawls 20 billons pages. Therefore, a large number of machines are required to fulfil the purpose. Hence, an effective mechanism should be followed to increase the crawling rate.
2. Politeness: crawling algorithm needs to be designed in a way that it only sends one request to a server at a time to avoid bombarding of requests on a page which leads to an overloading of the page. To avoid this politeness delay is introduced between the request. This would help to minimize the risk.
3. Excluded content: A file *robot.txt* of a web page is needed to be fletch first, before fetching a page from the site to find out whether the webmaster has specified how many files can be crawled [3].
4. Duplicate Content: Crawler should eliminate and recognize the duplicate data found on different URLs. Methods like checksum, visitor counter, fingerprinting, etc. are needed for this purpose.
5. Continuous Crawling: Carrying out full crawling after regular intervals isn't a beneficial approach to follow. This leads to low-value and static pages.
6. Spam Rejection: A crawler should be ready to reject links to URLs on the present blacklist and may lower the priority of pages that are linked to or from blacklisted sites.

III. METHODOLOGY

A. Raw Data Collection

Over the World Wide Web (WWW), there is a traffic checker which is checking the Traffic on different websites. Mainly there are 4-5 traffic rankers like

Alexa, CISCO Umbrella, Quantcast, and Majestic. They monitor the traffic over the WWW and made a list of 1 million top traffic domains over the past 3 months. Here we are getting our data from Tranco. Tranco's rank is based on a combination of Alexa, Umbrella, and Majestic.

The metadata which is provided by them is mainly Publicly Indexable web (PIW) and which helps not to risk our speed of delivering the results. While it also helps the crawler not to search deeper on the web or in the hidden web as it could rely on the surface web which could be accessed much easier.

In today's scenario, an average person surfs many on the surface web like Facebook, Instagram, Amazon, etc. the surface net has some series of protocols which are been standardized by the search engines (like google, duck-duck go, etc). These search engines use filters to present their search to a user. Therefore, every site make sure to have those series of protocols to get filtered and display as a result of search engines

B. *Multi-Threading*

As the number of metadata is large, therefore, we have used distributed crawler which helps to work in parallel searching using the multithreading technique that runs multiple threads all at once by swiftly switching between the threads with CPU help or we can say context switching. Although it also allows sharing of data space with the main thread inside a process that shares information and communication with other threads simpler than individual processing. The aim is to perform multiple tasks at once, which increases performance and improve rendering.

C. *Software and Hardware Requirment*

The minimum hardware requirement to run the program requires 4GB of ROM; 2GB of RAM; a 4-core processor; and an Internet Connection (5 Mbps). While the preferred requirement is 8GB of RAM; 4GB of ROM; a 6-core processor and an Internet connection (50 Mbps).

Software Requirement is Python 3.0 and operating system (Windows); Elastic search and Kibana as localhost.

D. *Flow Chart*

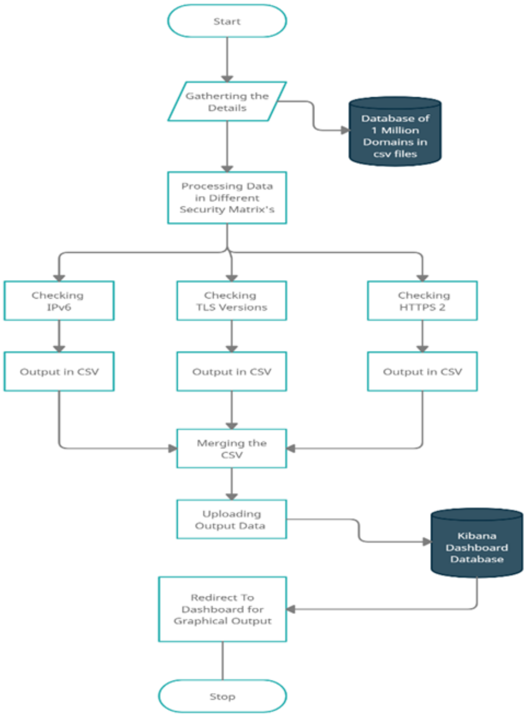
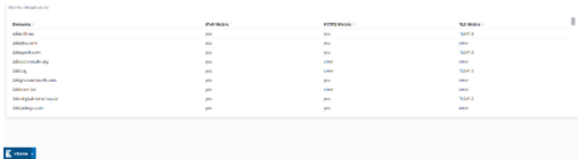


Fig. 1: Working of CrawlAnliser

The Crawlanliser gathers the raw or metadata which is been crawled and uploaded on Tranko. Tranko is a site that collects the crawled data from Alexa and umbrella (cisco) and combines it with its algorithm and avails for public use as an OPEN-SOURCE data. The metadata contains about 1 million of crawled data with the current ranking and their last ranking within the last 30 days. The data contains Global ranking, Domain, TLD. Subnet reference, IDN domain Previous Global ranking, and many more. The data has been stored and then proceeds for the security checks.

The Security checks are IPv6 Protocols, TLS version, and HTTPs 2 check. These security checks will be done through the scripts which are coded in python. This script will take the domain name as input and passes through the code and store the output in a CSV file. Each security check script will run separately and store the data. The stored data will be merged into one file. The file will be uploaded to a Dashboard that is running on Elastic Search and Kibana, the dashboard will give us a better presentation and make it easy to understand the data.

IV. RESULT



Domains	IPv6 Metric	HTTP2 Metric	TLS Metric
[illegible]	yes	yes	TLSv1.3
[illegible]	yes	yes	error
[illegible]	yes	yes	TLSv1.3
[illegible]	yes	error	error
[illegible]	yes	error	TLSv1.3
[illegible]	yes	yes	error
[illegible]	yes	error	error
[illegible]	yes	yes	TLSv1.3
[illegible]	yes	yes	error

Fig. 2: Dashboard

The dashboard displays the list of domains with the column heading of security metrices like HTTP2, IPv6, and TLS version, with the results along with the domain.

V. CONCLUSION

Information Security is the major key that helps the user to keep his digital identity safe and secure. Since there is number of cyber-crimes are increasing at a rapid pace in today’s world. Therefore, Information security is playing a key role.

This project works in the field of Information Security where its main lookup is for privacy and security domain which is today's need for everyone. This checking of the top 1 million helps the researchers to secure the website and make sure the secure connection and no mishappening happens.

Where, a massive number of 1 million domain checking the checking security metrics on each domain can cost more time, therefore more the number of threads lesser the time for the output could act as a limitation

While in the future we can add more security metrics parameters to perform security checks and could also implement a Sub-domain finder where we can check the subdomains of the site.

REFERENCES

[1] Shkapenyuk V. and Suel T. (2002), “Design and Implementation of a high-performance distributed web crawler”, In Proc. 18th International Conference on Data Engineering

[2] Nemeslaki, András; Pocsarovszky, Károly (2011), “Web crawler research methodology”, 22nd European Regional Conference of the International Telecommunications Society

[3] Sandeep Sharma (2008), “Web-Crawling Approaches in Search Engines”, Thapar University, Patiala.

[4] S. Chakrabarti. Mining the Web. Morgan Kaufmann, 2003..

[5] Priyanka-Saxena (2012), “Mercator as a web crawler”, International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, ISSN: 1694- 0814.

[6] DhirajKhurana, Satish Kumar (2012), “Web Crawler: A Review”, International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, ISSN: 2231 –5268.

[7] K. M. and Michelsen, R. (2002). Search Engines and Web Dynamics. Computer Networks, vol. 39, pp. 289–302, June 2002.