Journal of Harbin Engineering University
ISSN: 1006-7043
Vol 45 No. 6
June 2024

Development of Text-based Big Data Collection Technology

Seung-Yeon Hwang[1], Jeong-Joon Kim[2,*]
[1]Dept. of Computer Engineering, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea
[2,*] Corresponding Author, Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea

Abstract
Big data collection technology is a very important part of the cyber strategy field due to the technological development of the Fourth Industrial Revolution. Big data collection technology has various technologies such as Flume, Crawling, Scoop, Scribe, and Kafka, and is widely commercialized. Among them, a solution was developed directly using crawling technology, and data was collected for security-related papers among cyber strategy fields. The collected data has PDF files of papers in the form of metadata and raw data. The collected metadata is stored in MongoDB, and the paper PDF file is distributedly stored in GridFS. A total of 20,775 paper PDF files were collected from the first publication date of each conference and journal to August 2020. Based on the collected data, technology for efficient collection and long-term preservation of public data in the cyber strategy field can be secured, and actual data collected from various collection sites can be quickly accessed through metadata.
Keywords: Big data collection, technology, Cyber strategy, Crawling technology, Metadata storage

1. Introduction

Big data collection technology in the cyber strategy field is a technology that searches for necessary data from various data sources related to cyber strategy and collects it manually or automatically, and secures refined data through search/collection/transformation. At the heart of related technologies, the Apache Software Foundation (ASF) supports and manages various open-source projects related to big data technologies as well as big data collection technologies [1]. Technologies related to big data collection include Flume, Crawling, SQOOP, Scribe, and Kafka [2]. Flume is a technology that effectively collects large amounts of log data in a distributed environment and transmits it to other places, enabling real-time log analysis [3]. Crawling is mainly used to collect data released on the Internet using web robots [4]. Scoop is a technology that supports data transfer between Hadoop and relational databases and is used to transfer data from databases such as MySQL to Hadoop Distributed File System (HDFS) [5]. Scribe is a log collection technology developed by Facebook that collects log data received in real time from many servers and stores it in a HDFS [6]. Kafka was first created in Linkedin, collecting log data as well as compressing transmission data and sending messages collectively through the messaging system [7].

Research on big data collection technology and system development at home and abroad is active, but research on learning big data collection technology in the field of strategy optimized for cyber strategic technology development and strategy analysis is insufficient [8]. In addition, since the cyber strategy field is a very important area in terms of future national and social security, research and development of related technologies are actively being conducted in most developed countries [9]. In order to secure leadership in the cyber strategy field and take preemptive action, it is very important to secure excellent source data with guaranteed reliability, which is key to developing infrastructure technologies in the cyber strategy field [10].

Considering the importance of the cyber strategy field, localization of related technologies is urgently needed as it is only a short-term solution and adversely affects the development of future-oriented national security fields. In addition, it is necessary to conduct research on related technologies that can identify public specialized data in the field of cyber strategy and automate and collect learning big data from various systems. Therefore, by developing data collection technologies specialized in cyber strategy fields, it

is expected to secure excellent source data with guaranteed reliability and be used as a key base technology for the development of AI-based cyber strategy systems.

Starting with Section 1 Introduction, Section 2 introduces related technologies for the collection solution developed in this paper, and selected conferences and journals. Section 3 introduces the main modules for the solution, UI for the developed solution, and how to use it. Section 4 describes the results of the collected conferences and journals, and this paper is concluded through the final conclusion.

2. Related Works

2.1 Sqoop

Scoop is a solution that can collect and store data between storage such as HDFS, Hive, and Hbase of Hadoop, a big data storage solution, from relational databases such as MySQL, Oracle, and MS-SQL, and store data from Storage back in relational databases [11].

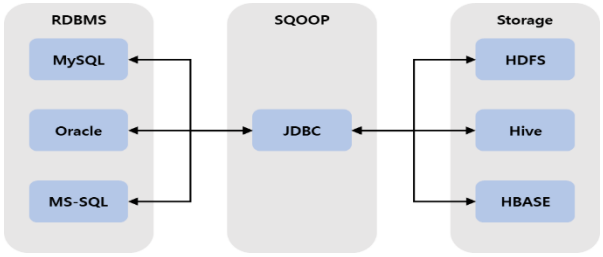


Fig. 1. The basic structure of Sqoop

Figure 1 shows the basic structure of the scoop solution, storing RDBMS' data in HDFS, Hive, and Hbase, which are storage in use using scoop, and JDBC drivers are required when using scoop. JDBC drivers are provided on each RDBMS official website. In addition, the process of storing RDBMS data in Storage is defined as Import, and the process of storing storage data in RDBMS is defined as Export.

2.2 Flume

Flume is a technology that collects log data developed as an open source project, and effectively collects large amounts of log data produced by multiple servers and provides the ability to transmit data to remote destinations such as HDFS. The simple and flexible structure allows you to configure various types of streaming data flow architectures, and uses Flume for log collection to ensure reliability, scale scalability, and functional scalability [12].

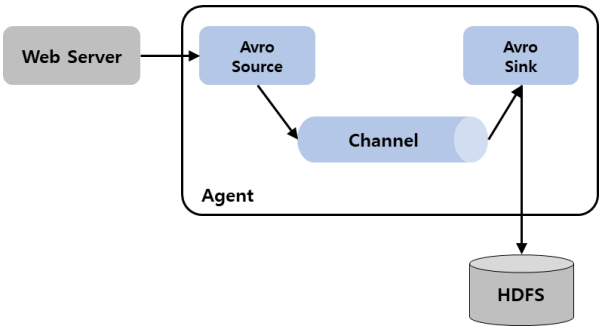


Fig. 2. The basic structure of Flume

2.3 Chukwa

Chukwa is an open source project created to collect log data of nodes distributed, store and analyze the collected data. The logs collected by Chukwa collect a variety of data, including monitoring logs, application logs, and Hadoop logs, and were developed to monitor log data over terabytes [13].

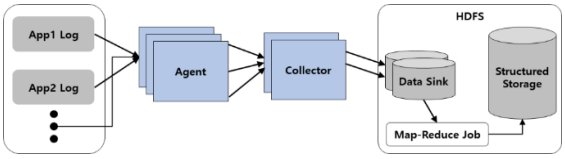


Fig. 3. The basic structure of Chukwa

Figure 3 shows the basic structure of Chuck, and the agent installs the log on the monitored source node to collect the log and sends the log file or server information to the collector. The collector receives log information from multiple agents and stores it in HDFS.

2.4 Scribe

Scribe is an application for real-time streaming log data collection from large servers developed on Facebook, and scribes are designed for network and system failures and aim for scalability and reliability. Facebook is installed and operated on a scale of thousands of units and collects 10 billion messages a day [14].

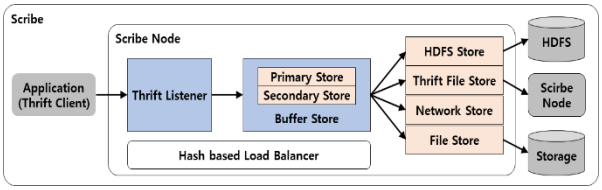


Fig. 4. The basic structure of Scribe

Figure 4 shows the basic structure of scribe. It consists of a single central scribe server and several local scribe servers, and the scribe server operates on all nodes of the system. If the central scribe server does not work, the local scribe server writes a message to a file on the local disk and sends the message again when the central scribe server

recovers to prevent loss of the message. The central scribe server writes a message to a file of a last destination, such as a distributed file system, or sends a message to a scribe server on another floor. In this case, the scribe server provides various types of stores using the concept of a store to store a message, which may also store a message in HDFS.

2.5 Crawling

Crawling is a computer software technology that allows users to extract desired information from websites and collects information and external data from web documents provided on the Internet, such as SNS, news, and web information. By programming information on the web that does not provide Open API services, you can collect a vast amount of data existing on the web to find desired information. A crawling algorithm suitable for work should be selected and used in consideration of network bandwidth, temporal problems, and hardware storage [15].

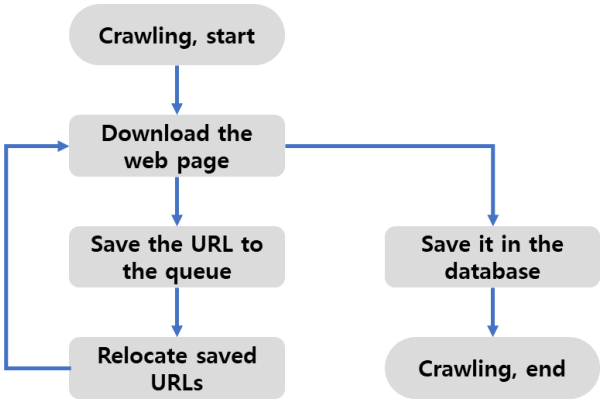


Fig. 5. Basic crawling flow chart

Figure 5 shows the basic flowchart of crawling. When the download is completed, starting with the download of the web page, the URLs on the web page are stored in the queue and rearranged according to importance among the stored URLs. In addition, the URL corresponding web page with high priority among the rearranged URLs is downloaded first, and then the URL is stored in the queue, and the URL is scheduled. The above process is repeated periodically until all URLs are visited, and at the end of the repetition, the necessary data is stored in the form of the database and data structure, and crawling is completed. Details of the crawling operation process may be added or changed to the flowchart described above according to the type of crawler and the crawling algorithm.

Among the types of crawling, Beautiful Soup is a Python-based library that extracts information from HTML and XML files, downloading HTML using requests or urlib, a built-in module of Python, and extracting data with Beautiful Soup. Because HTML is downloaded from the server, it is difficult to crawl to sites that do not use server-side rendering or sites that require JavaScript rendering. However, it is very easy to use, and when you use multi-process or multi-thread, the speed is fast [16].

Another type of crawling selenium is a framework used for web automation tests (click button, scroll operation, text input, etc.), and crawlers using selenium can easily import data generated through JavaScript rendering from web pages. Since it is a concept of crawling through an Internet browser, you can import all of the web pages you actually see, and debugging methods are also intuitive. However, because it is a way to actually run a web browser, it is slow and takes up relatively much memory. Crawling to multiple browsers using multiprocessing can partially improve speed [17].

Scrapy, the last type of crawling, is a framework developed for crawling and can use various functions and plugins such as middleware, pipeline, JavaScript rendering, proxy, Xpath, and CLI. Features related to parallel processing, compliance with robots.txt, and download speed control can also be set, and cloud services are also provided to upload crawlers developed in scrapy. It can be linked to backend services of web frameworks such as Django, has a variety of plug-ins, is used as an all-around framework, has good guide documentation, and has a simple structure, so it is a framework that can help crawlers a lot. It is an advantage because there are various plug-ins, but there is an issue that plug-ins are not compatible with each other [18].

Tab. 1. Comparison of crawling libraries and frameworks

Library Name	Velocity	Resource	Guide	Community	Function	Difficulty
BeautifulSoup	Fast	Lightness	Normal	Normal	Not Much	Easy
Selenium	Slow	Heavy	Good	Good	Normal	Easy
Scrapy	Fast	Lightness	Good	Good	Much	Normal

2.6 Identification of public data in the field of cyber strategy

In this paper, a list of 16 security-related conferences and journals selected to identify public professional data in the field of cyber strategy is as follows.

The three conferences are the ACM Computer and Communications Security Conference (ACM CCS), The Network and Distributed System Security Symposium (NDSS), and USENIX Security (ACM Computer and Communications Security Conference), which are the three major conferences related to security. It mainly deals with various areas related to security, such as system security, network security, security policy, algorithms, and cryptography.

Seven of the remaining 13 journals have ACM Transactions on Autonomous and Adaptive Systems (ACM TAAS), ACM Transactions on Autonomous and Adaptive Systems (ACM TECS), ACM Transactions on Computer Systems (ACM TOCS), ACM Transactions on Database Systems (ACM TODS), ACM Transactions on Internet Technology (ACM TOIT), ACM Transactions on Privacy and Security (ACM TOPS), and ACM Transactions on Sensor Networks (ACM TOSN) organized by ACM. Four of the remaining six journals include IEEE Transactions on Information Forensics and Security (IEE TIFS), IET Information Security (IEEE IS), IEEE Security & Privacy (IEE SP), and IEEE Transactions on Security (IEEE TDSC). The remaining two journals include Science Direct Journal of Information Security and Applications (SDISA) and Science Direct Computers & Security (SDCS) organized by Science Direct. Thirteen journals are also security-related journals that cover various fields related to network, operating system, protocol, information forensics, big data, and cryptography.

3. Methods

Figure 6 shows the functions of the main modules of the architecture of the text-based big data collection solution implemented in this paper.

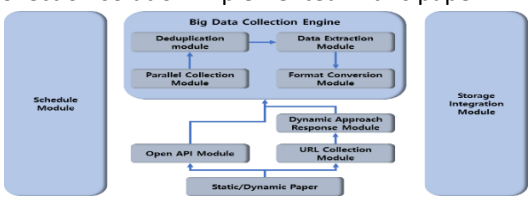


Fig. 6. Text-based big data collection technology architecture

3.1 Open API Module

The Open API module provides the ability to download paper files from statically open sites, rename downloaded paper files, and specify related preferences to bypass bot detection when accessing the site.

3.2 URL Collection Module

The URL collection module provides a function for collecting URL addresses to be searched from dynamically disclosed paper providing sites.

3.4. Parallel Collection Module

The Parallel Collection Module provides multi-thread-based parallel collection that can maintain high speed, minimal communication overhead, and error tolerance when collecting data from multiple paper-providing sites in a big data collection engine.

3.5. Deduplication Module

The Deduplication Module matches papers collected through a big data collection engine based on the Title field of metadata stored for similar papers to automatically move to the next paper collection URL without temporarily collecting duplicate papers.

3.6. Data Extraction Module

The Data Extraction Module provides the ability to identify and extract conference or journal names, paper titles, authors' names, summaries, date and time at the time of collection, date when the paper was published, keywords, journal types, citation indices, and paper PDF file names.

3.7. Format Conversion Module

The Format Conversion Module provides a function to convert papers (PDFs) collected through a big data collection engine into TXT format files and store the converted TXT files in the directory where the currently collected papers (PDF) files are located.

3.8. Schedule Module

The scheduler module provides a reservation function that allows you to specify a date and time for big data collection targets (static/dynamic paper provision sites) and automatically start collection of papers on the specified date and time.

3.9. Storage Integration Module

The storage interworking module provides a function for storing final papers (PDFs) collected through crawlers in a MongoDB-based GridFS file system.

3.10. How to use the program

The UI of the text-based big data collection technology crawler developed through this study is shown in Figure 7.

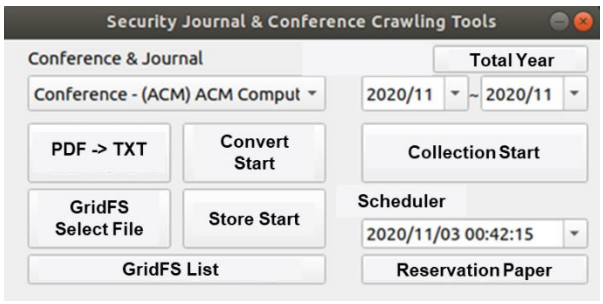


Fig. 7. Text-based big data collection technology crawler UI

Fig. 8. Order of paper collection process

Figure 8 shows the order of the paper collection operation process, and you can select a conference or journal to collect in the combo box below the Conference & Journal text, specify the period to collect through the Year/Month (2020/11) box below the year selection on the right, and click the Collection Start button to collect papers. Figure 9 shows the selection of conferences or journals to be collected in the combo box under the Conference & Journal text.

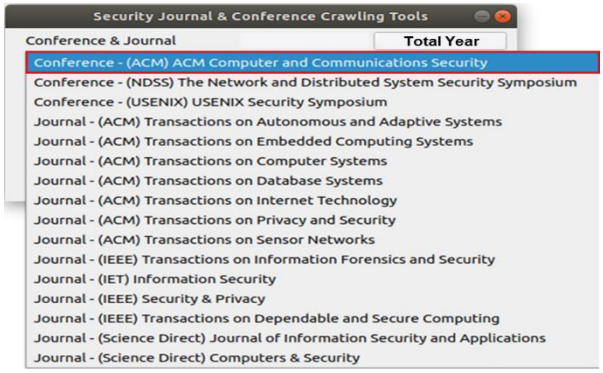


Fig. 9. The appearance of conferences and journals printed when selecting conference & journal

Figure 10 shows the year selection of the date to be collected as a calendar by clicking the combo box button in the Year/Month (2020/11) input box below the year selection.

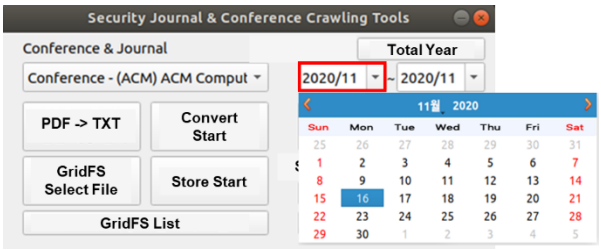


Fig. 10. Calendar output from the paper collection period date setting

In addition, when you click the Total Year button, the year selection is completed by automatically setting the start and last year in which the paper of the corresponding conference or journal was published, and when you click the Collection Start button, the paper collection begins. Figure 11 shows an example of the operation process in which papers are collected and the output of the message notification window that paper collection begins by selecting the ACMCCS conference, clicking Set Total Year, and clicking the Collection Start button.

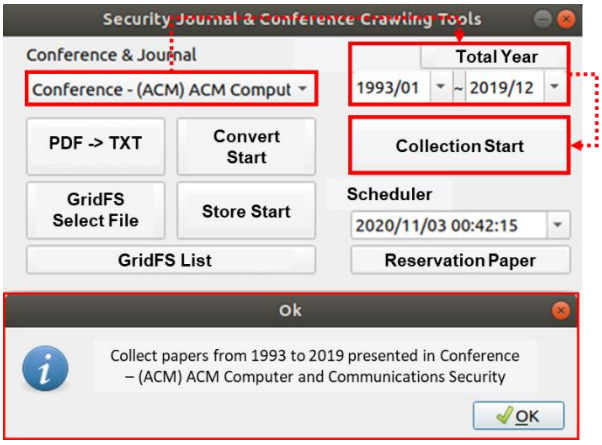


Fig. 11. The paper collection process and message notification window appearance

Figure 12 shows the sequence of the operation process of converting the collected paper PDF into TXT.

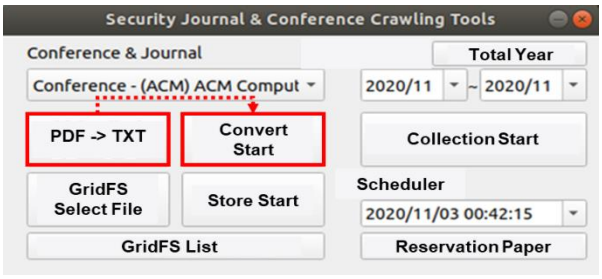


Fig. 12. The order of TXT conversion operation of the paper PDF

When PDF -> TXT button is clicked, a file explorer is executed, and when a PDF file to be converted is selected and the Convert Start button is clicked, the format of the PDF file may be converted to TXT. Figure 13 shows the selection of a PDF file in the file explorer executed by clicking the PDF-> TXT button.

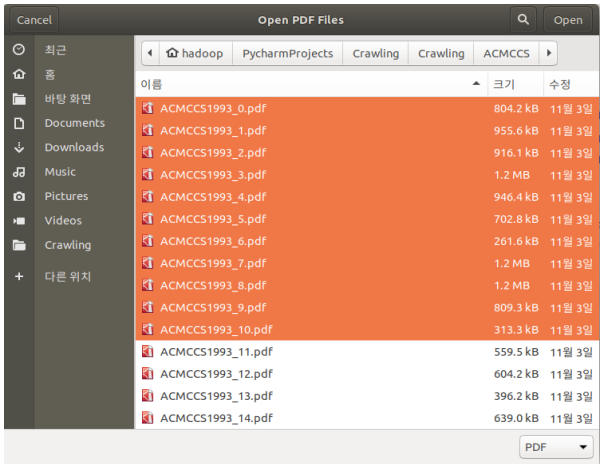


Fig. 13. Select a list of PDF files to convert in the file explorer

Figure 14 shows how the previously selected PDF file is converted into a TXT format and stored in a directory location where it exists.

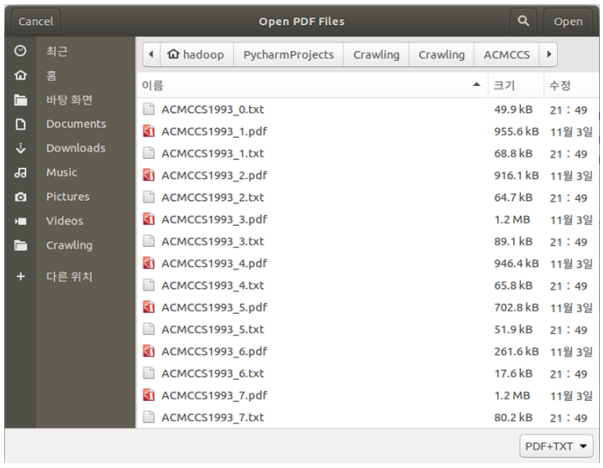


Fig. 14. Check the converted TXT file list in the file explorer

Figure 15 shows an example of selecting a PDF file to convert to TXT and clicking the Convert Start button to convert it, and outputting a message notification window that the conversion has been completed.

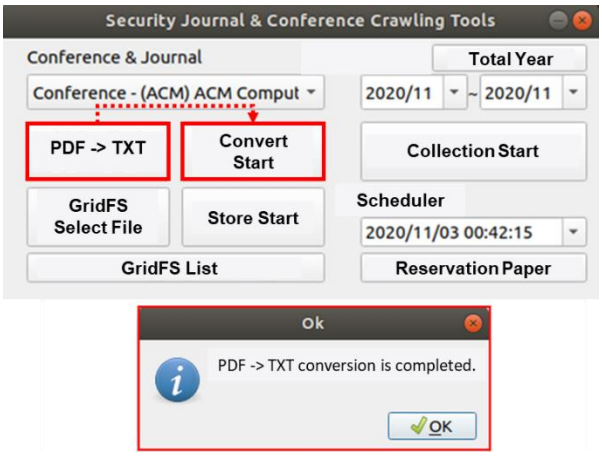


Fig. 15. Paper PDF -> TXT conversion operation process and message notification window appearance

Figure 16 shows the sequence of the GridFS file storage operation process.

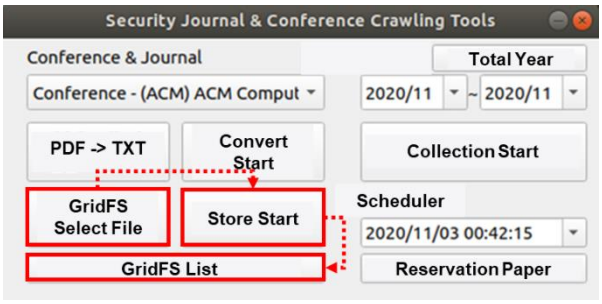


Fig. 16. GridFS file storage operation sequence

When you click Select GridFS File, the file explorer is executed. Then, select a PDF file to be saved in GridFS, and press the Store Start button to save the previously selected PDF file in GridFs. Figure 17 shows the selection of PDF files to be stored in the file explorer running through the GridFS save file selection click.

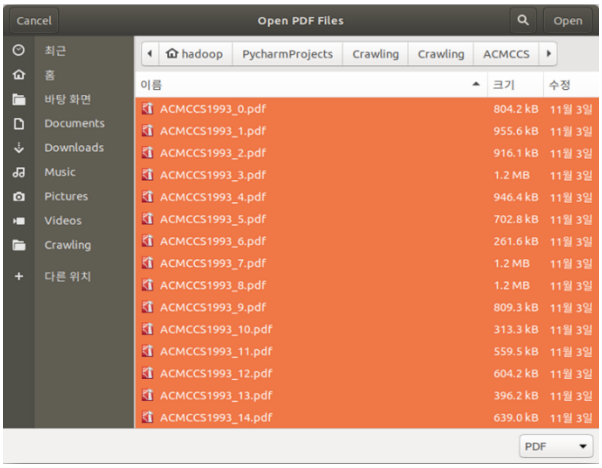


Fig. 17. Select a list of PDF files to save in the file explorer

Figure 18 shows a new window window opening by clicking GridFS List and outputting a list of files stored in GridFS in the form of Table View.

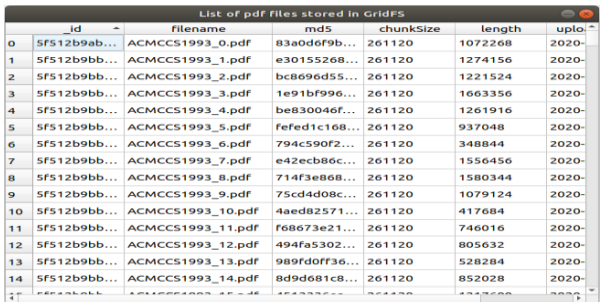


Fig. 18. Output a list of PDF files stored in GridFS

Figure 19 shows an example of saving by selecting a PDF file to be saved in GridFS and clicking the Store Start button, and outputting a message notification window that the save has been completed.

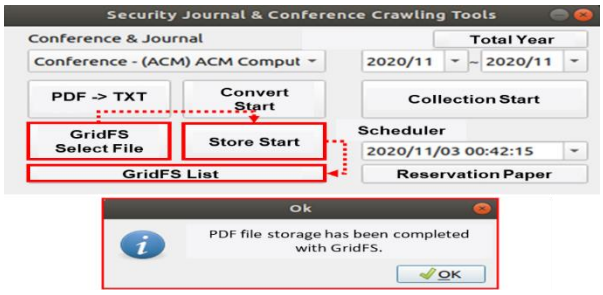


Fig. 19. GridFS file saving operation process and message notification window appearance

Figure 20 shows the sequence of the scheduler operation process.

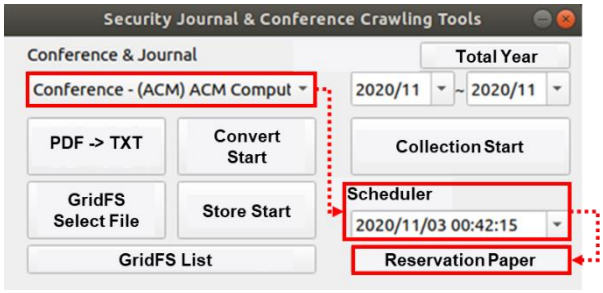


Fig. 20. The order of the scheduler

You can select a date and time from the calendar by selecting a conference or journal to be collected from the combo box under the Conference & Journal text, and a combo box expressing the date and time under the scheduler text. And after selection, you can collect papers on the previously set date and time by pressing the Reservation Paper button. Figure 21 shows the appearance of a calendar output by selecting a combo box in the date and time input box under the scheduler text.

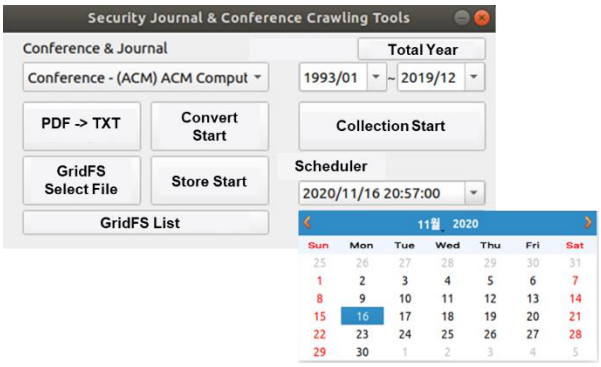


Fig. 21. The calendar that is printed on the schedule's date and time settings

Figure 22 shows an example of the order of the scheduler operation process. Select the ACMCCS conference and click Total Year to set the entire period of papers to collect. In addition, in the Date and Time input box below the scheduler text, specify the date and time to start collecting papers, and click the Reservation Paper button to show an example of the operation process in which papers are scheduled and collected. Finally, when you click the Reservation Paper button, you can see the output of a message notification window and a reminder message notification window that can check the date and time of the paper collection reservation.

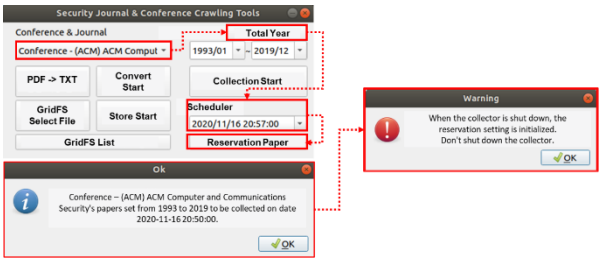


Fig. 22. Schedule movement process and message notification window

4. Results

4.1 Metadata

Metadata on big data collection targets using crawlers developed through this study differs by society, but generally shows similar relationships.

The ACMCCS conference, ACMtass, ACMtecs, ACMtocs, ACMtods, ACMtoit, ACMtoit, and ACMtosn journals are partially expressed as "null" because they sometimes do not provide data of the Author column and the Astract column among metadata, and the entire "null" column data.

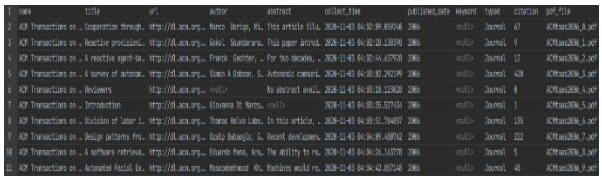


Fig. 23. Metadata samples collected from the Journal of ACMCCS, ACMtass, ACMtecs, ACMtocs, ACMtods, ACMtoit, ACMtops, and ACMtosn

NDSS and Usenix Security conference are partially expressed as "null" because they sometimes do not provide data for Author columns and Abstract columns among metadata, and data for Keyword columns and Cation columns are not provided by NDSS conference, so the whole is expressed as "null".

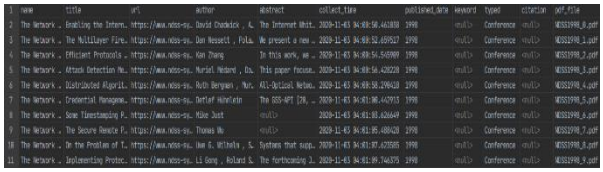


Fig. 24. Metadata samples collected by NDSS, Usenix Security conferences

IEEEis IEEEsp, IEEEtdsc, and IEEEtifs journals can extract and store all data because they provide all the metadata column data that can be collected, but are partially expressed as "null" because they sometimes do not provide data from the Abstract column and Keyword column.

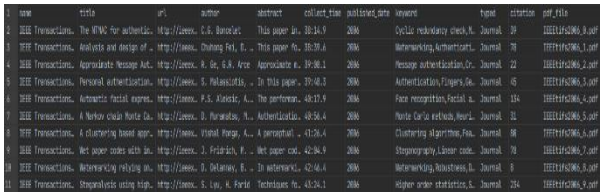


Fig. 25. Metadata samples collected from IEEEis IEEEsp, IEEEtdsc, IEEEtifs journals

SDjisa, SDcs, and journals can extract and store all data because they provide all the metadata column data that can be collected, but they are partially expressed as "null" because they sometimes do not provide data from Author columns, Abstract columns, and Keyword columns.

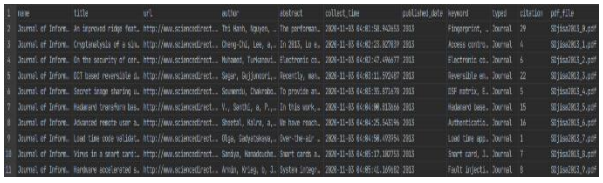


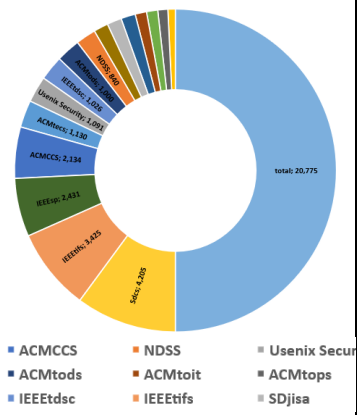
Fig. 26. Metadata samples collected from SDjisa, SDcs journals

4.2 Number of Papers Collected

Tab 2 shows the total number and number of papers collected when papers were collected based on each year of the conference and journal that can be collected using crawlers developed through this study.

Tab. 2. Collected counts and totals

Conference/ Journal name	Number of counts	Graph
ACMCCS	2,134	
NDSS	840	
Usenix Security	1,091	
ACMtaas	313	
ACMtecs	1,130	
ACMtocs	468	
ACMtods	1,000	
ACMtoit	476	
ACMtops	419	
ACMtosn	607	
IEEEis	604	
IEEEsp	2,431	
IEEEtdsc	1,026	
IEEEtifs	3,425	
SDjisa	606	
Sdcs	4,205	
Total	20,775	



A total of 20,775 papers were collected, and SDcs journals were the largest among the 4,205 papers collected, followed by IEEEtifs journals with 3,425 IEEEsp journals with 2,431 and ACMCCS conferences with 2,134. The fewest conferences and journals collected were the ACMtaas journals, with 313.

5. Conclusions

In this paper, we develop a solution that can collect text-based big data, crawling and metadata data existing on the web in the world's top three security conferences and various related journals, and store

paper files in the form of raw data as PDF. Technology for efficient collection and long-term preservation of public data in the field of cyber strategy can be secured, and actual data collected from various sites can be quickly accessed through the collected metadata. In addition, since the web pages of conference and journal collection sites are different, ordinary users who have difficulty finding papers to collect can easily collect papers through the UI by using this text-based big data collection crawler.

In addition, securing source data specialized in the field of cyber strategy and using it for analysis and prediction is expected to contribute to strengthening the competitiveness of global cyber warfare. In addition, it is expected that big data collection technology will strengthen competitiveness in a data-based economic society by automating and collecting source data in various fields as well as cyber strategy fields. Therefore, by researching and developing strategic technology learning big data collection technology optimized for the cyber strategy field based on existing big data collection technology at home and abroad, we can expect a foothold to take a leap forward in the cyber strategy field.

Acknowlegement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1F1A1062953).

References

[1] Kabinna, S., Bezemer, C. P., Shang, W., & Hassan, A. E. (2016). Logging library migrations: A case study for the apache software foundation projects. In 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR) (pp. 154-164). IEEE.

[2] Anuradha, J. (2015). A brief introduction on Big Data 5Vs characteristics and Hadoop technology. Procedia computer science, 48, 319-324. https://doi.org/10.1016/j.procs.2015.04.188

[3] Birjali, M., Beni-Hssane, A., & Erritali, M. (2017). Analyzing social media through big data using infosphere biginsights and apache flume. Procedia computer science, 113, 280-285. https://doi.org/10.1016/j.procs.2017.08.299

[4] Pant, G., Srinivasan, P., & Menczer, F. (2004). Crawling the web. In Web dynamics (pp. 153-177). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-10874-1_7

[5] Chen, L., Ko, J., & Yeo, J. (2015). Analysis of the influence factors of data loading performance using Apache Sqoop. KIPS Transactions on Software and Data Engineering, 4(2), 77-82. https://doi.org/10.3745/KTSDE.2015.4.2.77

[6] Bumbalek, Z., Zelenka, J., & Kencl, L. (2010, July). E-Scribe: ubiquitous real-time speech transcription for the hearing-impaired. In International Conference on Computers for Handicapped Persons (pp. 160-168). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14100-3_25

[7] Wang, G., Koshy, J., Subramanian, S., Paramasivam, K., Zadeh, M., Narkhede, N., Rao, J., Kreps, J & Stein, J. (2015). Building a replicated logging system with Apache Kafka. Proceedings of the VLDB Endowment, 8(12), 1654-1655. https://doi.org/10.14778/2824032.2824063

[8] Siboni, G., & Assaf, O. (2016). Guidelines for a national cyber strategy (p. 12). Tel Aviv: Institute for National Security Studies.

[9] Lilly, B., & Cheravitch, J. (2020). The past, present, and future of Russia's Cyber strategy and forces. In 2020 12th International Conference on Cyber Conflict (CyCon), 1300, 129-155. IEEE. https://doi.org/10.23919/CyCon49761.2020.9131723

[10] Eom, J. H., Kim, N. U., Kim, S. H., & Chung, T. M. (2012). Cyber military strategy for cyberspace superiority in cyber warfare. In Proceedings title: 2012 international conference on cyber security, cyber warfare and digital forensic (cybersec) (pp. 295-299). IEEE. https://doi.org/10.1109/CyberSec.2012.6246114

[11] Vohra, D. (2016). Using apache sqoop. In Pro Docker (pp. 151-183). Apress, Berkeley, CA.

https://doi.org/10.1007/978-1-4842-1830-3_11

[12] Jung, S., & Shin, Y. (2018). Study of the big data collection scheme based Apache Flume for log collection. International Journal of Computer Theory and Engineering, 10(3), 97-100. https://doi.org/10.7763/IJCTE.2018.V10.1206

[13] Rabkin, A., & Katz, R. (2010, November). Chukwa: A system for reliable large-scale log collection. In Proceedings of LISA'10: 24th Large Installation System Administration Conference (p. 163).

[14] Borthakur, D., Gray, J., Sarma, J. S., Muthukkaruppan, K., Spiegelberg, N., Kuang, H., ... & Aiyer, A. (2011). Apache hadoop goes realtime at facebook. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (pp. 1071-1080). https://doi.org/10.1145/1989323.1989438

[15] Cothey, V. (2004). Web-crawling reliability. Journal of the American Society for Information Science and Technology, 55(14), 1228-1238. https://doi.org/10.1002/asi.20078

[16] Uzun, E., Yerlikaya, T., & KIRAT, O. (2018). Comparison of Python libraries used for Web data extraction. FUNDAMENTAL SCIENCES AND APPLICATIONS, 87.

[17] Bruns, A., Kornstadt, A., & Wichmann, D. (2009). Web application tests with selenium. IEEE software, 26(5), 88-91. https://doi.org/10.1109/MS.2009.144

[18] Fan, Y. (2018). Design and implementation of distributed crawler system based on scrapy. In IOP Conference Series: Earth and Environmental Science, 108(4), 042086. IOP Publishing. https://doi.org/10.1088/1755-1315/108/4/042086