ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research

Vol. 16, Issue, 10, pp.*30256-30260*, October, 2024
DOI: https://doi.org/10.24941/ijcr.47900.09.2024

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

PROPOSAL OF A HYBRID DATA MODEL FOR VULNERABILITY DETECTION IN BIG DATA

KANGA Koffi[1], KAMAGATE BémanHamidja[2], BROU Aguie Pacome Bertrand[3] and OUMTANAGA Souleymane[4]

Doctor in Computer Science: Software Engineering and Database: INPHB Doctoral School Teacher – Researcher at ESATIC (African Higher School of ICT: Republic of Côte d'Ivoire); Laboratory of Information and Communication Sciences and Technologies, African Higher School of ICT, LASTIC-ESATIC, Abidjan, Ivory Coast, 18bp 1501 Abidjan 18; [2] Doctor in Computer Science: Networks and Security: INPHB Doctoral School Teacher – researcher at ESATIC (African Higher School of ICT: Republic of Côte d'Ivoire); Laboratory of Information, Communication Sciences and Technologies, African Higher School of ICT, LASTIC-ESATIC, Abidjan, Ivory Coast, 18bp 1501 Abidjan 18; [3]Doctor of Computer Science at the Laboratory of Information , Communication Sciences and Technologies, African Higher School of ICT, LASTIC-ESATIC, Abidjan, Ivory Coast, 18bp 1501 Abidjan 18; [4]Lecturer in Computer Science at the Houphouët Boigny National Polytechnic Institute in Yamoussoukro Computer Science and Telecommunications Research Laboratory

ARTICLE INFO

Article History:
Received 24th July, 2024
Received in revised form
17th August, 2024
Accepted 29th September, 2024
Published online 30th October, 2024

Key Words

Intrusion Detection, Vulnerability Detection, Honeypot, Data Mode.

Corresponding author:* **KANGA Koffi

ABSTRACT

Today, with the digital revolution and its corollary of exponential data growth, capturing large volumes of data from various sources for processing at a high and acceptable speed would be a wish; but securing this data seems even better. To do this, we propose in our article, a data model for detecting vulnerabilities in big data. This model derives from two models, which are: a vulnerability scanning model (allowing to keep track of data from the weaknesses of the various protection measures implemented in big data), - a model from honeypots (for capturing data from various intrusion attempts in big data). The implementation of this hybrid model makes it possible to efficiently perform CMDI, SQLI, XSS, brute force login, basic, OWASP, DEFAULT LOGIN type scans at the request of different users for securing their infrastructures. Also, in these honeypot management functions, this model allows to perform NMAP type analyses, to identify the sources of possible attacks and their different forms (injection – ddos – portscan, etc.).

Citation: KANGA Koffi, KAMAGATE BémanHamidja, BROU Aguiepacome Bertrand, OUMTANAGA Souleymane. 2024. "Proposal of a hybrid data model for vulnerability detection in big data.*". International Journal of Current Research*, 16, (10), 30256-30260.

INTRODUCTION

Talking about a data model for vulnerability detection in big data deserves some explanation. In fact, in the field of information science, a data model describes the way in which data is represented in an organization, an information system or a database. For some authors (De Mauro, 2016), a data model represents a structural base, represented in the form of a well-defined graphical characterization of a business information system. As for the vulnerability of a computer system, it falls under the domain of cyber-security. Thus, it can be defined as a weakness exploited by cybercriminals to gain unauthorized access to a computer system. After exploiting a vulnerability, a cyber-attack can execute malicious code, install malware, and even steal sensitive data.

The vulnerabilities can be exploited by various methods, notably SQL injection, buffer overflows, (XSS) cross-site scripting , open source exploitation kits that search for known vulnerabilities and safety weaknesses in web applications. Numerous vulnerabilities affect popular software, exposing many customers using this software to an increased risk of data breach or attack. These so-called zero-day[1] exploit are recorded by MITER as "Common Vulnerability Exposure » (CVE [2]). In 2021, according to OWASP, the most common vulnerabilities in computer systems can be classified into ten (10) categories (Figure 1). Detecting a vulnerability consists of using methods and tools to highlight a security flaw. For big

[1]Exploits zero-day : is about a computer science program allowing to exploit a vulnerability present in a software Or Again A computer science system but Which is unknown to the publisher.

[2] CVE, For Common Vulnerabilities and Exposures, is a list of computer security weaknesses publicly disclosed.

data, several definitions have been proposed in research works (De Mauro, 2015; Ylijoki, 2016; Gärtner, 2017).

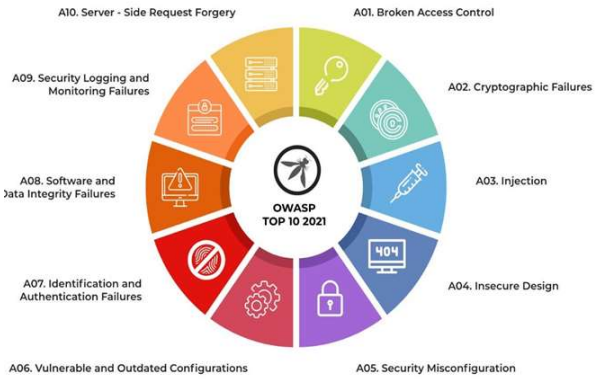

Figure 2. Top 10 OWASP vunerabilities

Here we have retained the one proposed by Gartner (Hemaizia, 2022). Thus, according to this company, big data is a concept that brings together a family of tools responding to a triple problem called the 3V rules (Volume – variety – velocity) (figure 2). Today, with the multiplication of data sources and the growing evolution of the quantities of data generated, the problem of the volume of data to be processed is essential. This problem would also be due to a very great diversity (Variety) of information (coming from various sources, unstructured, organized, Open…), and a certain level of speed (Velocity) to be achieved in their processing; in other words, the frequency of creation, collection and sharing of this data justifies this 3V problem. In big data, data storage is managed by four (4) types of DBMS (Database Management System) called nosql which are:

- Column-oriented DBMSs
- Key-value oriented DBMS
- Document-oriented DBMSs
- Graph-oriented DBMSs

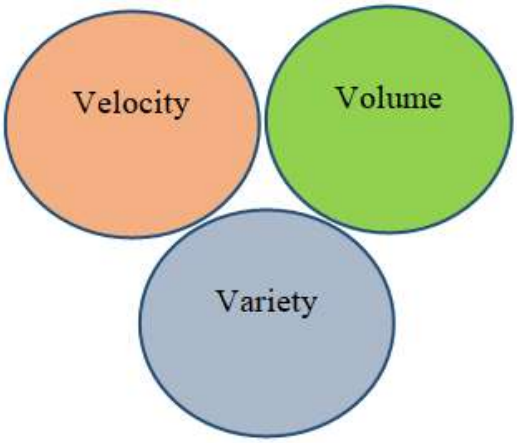

Figure 1 : the 3 v of big data

Today, with the digital revolution and its corollary of exponential data growth, capturing these large volumes of data from various sources for processing at a high and acceptable speed would be a wish; but securing this data seems even better. However, this security requires the implementation of a set of tools and processing methods (MAPREDUCE algorithm, machine learning, deep learning, etc.) based on data models.

Also, given that the storage of data in big data does not respect the same DBMS formats, it is necessary to propose a single and central model for these 4 DBMS families.

The remainder of our paper is organized as follows:

1. Section 2, we will present the state of the art
2. Section 3, we will identify our problem
3. In section 4, we will illustrate our contribution
4. Section 5 is devoted to a discussion and we will end with a conclusion in Section 6. In this section, we will outline some perspectives

State of the art

Quality control and performance study of NOSQL databases: In (6), the authors carried out a work focusing on the quality control of document-oriented data. They relied on a method for detecting and solving problems of schema overlap, data duplication and data incompleteness based on the frequency of appearance of data in document-oriented databases. It resulted in the establishment of a method called MFU (Most Frequently Used). This method (MFU), according to the authors, would include three phases:

- Detection of quality problems,
- Data repair,
- Quality check.

In each phase, their proposal would address the three types of data quality issues that they group into the triplet

(Schema – incompleteness and data duplication): As for (7), a comparative study of nosql databases and therefore big data is carried out in a general manner. The authors start from the general presentation of the concepts related to nosql, the ACID properties (Atomicity, Consistency, Isolation and Durability) of relational databases and their inadequacies; the common properties of nosql databases (BASE **(**Basically Available - SoftState–E ventually Consistent**))** (Figure 3) and the CAP theorem (figure 4) (Consistency (Coherence**)**-Availability (Availability)- Partition Tolerance (Distribution)). Specifically, the goal they set themselves is to study the different NoSQL databases available on the market in terms of architecture and data model and then to analyze the performance of two of them (Cassandra and MongoDB) using a test tool called YCSB. The expected result is to know the performance of each of the two databases and to make a comparison based on the test results generated by YCSB.

Figure 2. ACID vs BASE property of DBMS

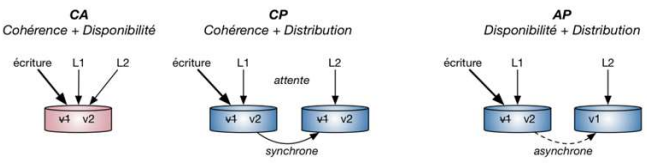

Figure 4. CAP Theorem

Data model and intrusion detection: In (Spitzner, 2002), the authors started from the observation that the volume of computer attacks increases exponentially in this new bipolar world with their corollaries of considerable damage. These attacks appear specific, powerful, intelligent and very complex. To stem the concerns created by these attacks, tools such as IDS and HONEYPOT are proposed by different research works (Wicherski, 2006; Goebel, 2006; Ben Younes, 2007), allowing to have prior knowledge on the attacks. According to the authors of the same paper (Spitzner, 2002), the search for prior knowledge constitutes one of the weak points of IDS in generating false alerts. As for the honeypot, these are tools used to study and acquire knowledge about attack techniques and behaviors. Thus, any movement towards a honeypot would be considered suspicious and therefore dangerous. In their analysis, the authors of this paper concluded that IDS and honeypots (HONEYPOT) have individual limitations and it would be beneficial to associate them to increase the intrusion detection capabilities of a computer system. To do this, they proposed in their paper as a contribution a quantitative and probabilistic model using data mining techniques to enrich the knowledge from IDS. In analyzing this work, the authors did not address the aspects related to data models allowing the storage of knowledge from IDS, honeypots and even the probabilistic and quantitative model that they proposed in (Spitzner, 2002).

Network intrusion detection: In (Lee, 1998), the authors start from the observation that the wireless equipment existing around us is often configured with low security levels, therefore vulnerable. In these conditions, several attacks or intrusions and intrusion attempts are possible. To do this, the authors of (Lee, 1998) propose intrusion detection tools as elements of solutions to this problem. In their solution proposals, they proceed to the modification and extension of a tool (Orchids**)** which is based on the verification of models and the detection of intrusions in wireless networks. Also, using signatures, attack scripts (chopchop – ARP replay**)** were written, implemented and detected in a real environment. In the analysis of this work, the static aspects (data) were not taken into account, much less a data model was proposed. In (Boudaoud, 2000), an intrusion detection system using data mining techniques is carried out. The main idea of this work lies in the use of classification techniques to recognize consistent and inconsistent behaviors of users and other computing resources. Also association rule algorithms and pattern frequency search are used in this work to compute the recording models. As a result of their work, an agent-based architecture for intrusion detection systems, where learning agents compute and continuously provide updated (detection) models to the detection systems is proposed. In the analysis, the authors did not consider any underlying data model of their work, much less the type of database (nosql for big data).

Intrusion detection based on multi-agent systems (Sajjaa, 2022**):** In this paper, the authors start from the observation that securing a computer system could be done in a preventive or reactive manner. Indeed, preventive security would consist in protecting a computer resource against unauthorized or abusive access. However, according to them, completely ensuring the security of a computer system is impossible. As for the reactive approach, it could find solutions to the preventive approach because it could allow detecting possible attacks early in order to avoid damage. This reactive technique could be similar to an intrusion detection technique.

In their contributory approach to setting up an intrusion detection system, they propose a multi-agent intrusion detection system. Specifically, they set up an organizational model of this MAS (multi-agent system) (figure 5) and also a functional model of a security agent. This organizational model presents two (2) levels which are:

- The management level composed of a security policy manager and intranet management agents
- The local level which manages a set of local agents (extranet local agent, intranet local agent and internal local agent.)

Communication between these two levels consists of sending messages, delegating monitoring functions and intrusion detection functions.

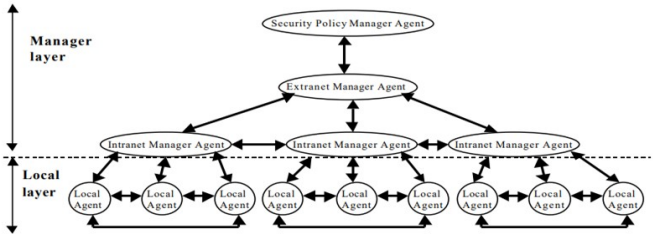

Figure 3. SMA Organizational Model (Sajjaa, 2022)

This functional model allowed the authors to set up a security event diagram (figure 6). This diagram shows the generic security management functions (*GenericSecurityEvent)*. These functions can be audit management functions *(AuditEvent)*, network management functions (*NetworkEvent)*and other types of functions.

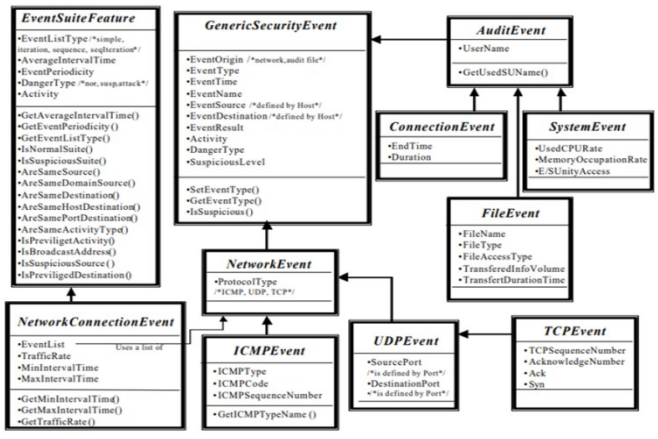

Figure 4. Security Event Uml Class (15)

Based on the analysis of this work, it appears that efforts to attempt a solution have been proposed. However, the aspects relating to the data model (figure 6) proposed are incomplete. Indeed, the data resulting from the different communications between multi-agent systems are not stored or even the underlying model of the proposed storage structure is incomplete.

Web intrusion detections: In (Bau, 2010), the authors start from the observation that, to find solutions to the different attacks on web applications, it would be necessary to implement protection and testing techniques including firewalls, IDS, and web page scanners.

To do this, they propose a web page clustering technique that allows the identification of vulnerabilities from a black box analysis. Under these conditions, each exploited vulnerability makes it possible to ensure that it does not correspond to a false positive. In particular, their work focused on SQL injections. The result of this work was the implementation of a SQL injection vulnerability scanner.

Problematic

From this literature review, it emerges that excellent research work has been conducted. This work has led to the implementation of IDS (intrusion detection system), honeypot and vulnerability scanning tools in web environments, computer networks and also big data, of which we have presented the most representative ones with regard to their functional models and interesting architectures. However, these works and tools produced still suffer from problems that make it impossible to effectively detect vulnerabilities and intrusions with a view to taking effective countermeasures. The question that arises in these conditions is the following:
Is there a reference data model around which these different works and tools could find their realizations so that the countermeasures to be taken and possible future intrusion detection works can be improved? Everything necessary to guarantee the quality of data exchanged and stored in a BIG DATA context.

Contribution

The state of the art that we have presented, reveals the high level of quality of the research work carried out with a view to securing computer data. However, certain aspects relating to the data resulting from the various vulnerability scan detection processes, intrusion detection from IDS and even honeypots set up are not taken into account. To do this, we are presenting a model that could serve as a basis for storing and identifying possible data from different vulnerability scans, data from IDS manipulation and also discoveries from honeypots.

Data models from vulnerability scans: Our first data model that we propose focuses on exposing the different vulnerability scans with their different associated methods. This data model that we are proposing allows you to manage scanning operations (scan). These scan operations concern owasp scans, basic scans, sql scans, xss, bruteforce_login, default_login, basic scans and scans_cmdi. In our approach, these scan operations are executed by users (user**)** at their request. The latter make scan requests addressed to an administrator. In our data model, these requests are materialized by the class (request_scan_particular). Also, the scan operation can be broken down into several types of operations which are:

- **OWASP scans** (Figure 1) represented by the data class “scan_owasp”
- **Standard scans** are called basic scans represented by the data class “scan_basic”
- **Brute force login scans**. These scans are of the bet type and include (Hybrid brute force attacks: allow to try or submit thousands of expected words and dictionary words, or even random words. Reverse brute force attacks: allow attempts to obtain password derivation key by performing exhaustive searches). They are represented in our data model by the class called "scan_brute_force_login"
- **Xss scans** ; these are scans relating to the injection of xss code into web applications; they are represented by the “ scan_xss ” class
- **SQL code scans**: These types of scans are performed in order to detect SQL vulnerabilities in applications. This type of scan is represented by the “ scan_sqli ” class
- **Default login scans**: these types of scans allow us to know if vulnerabilities due to default passwords exist in the applications. They are also represented in our data model by the class " scans_default_login "

All these vulnerability testing operations are supervised by an administrator represented in our data model by “administrator”

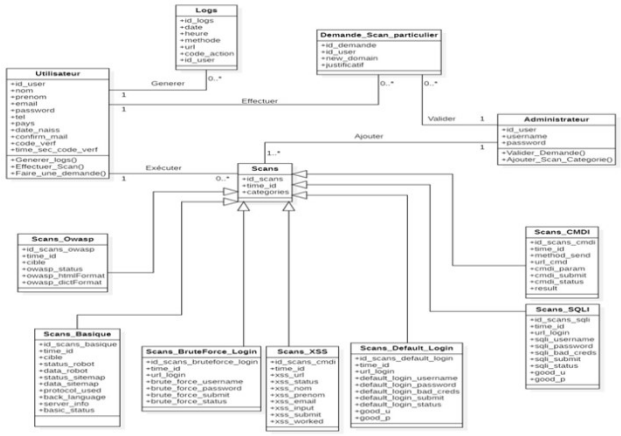

Figure 5. Data model from vulnerability scans

Data model from honeypots (Figure 8): Our second model for managing data from vulnerability scans relates to honeypots. Indeed, on the data model (figure 8), honeypots are represented by the class "honey_pot). It describes a honeypot by the IP address of the workstation on which it is installed, the listening port, the client node seeking to connect and the connected state of the server. These honeypots identified in their class, perform analyses through communication sockets through the class "analyse". This class has the attribute "socket". These analyses concern attacks **(**represented by the class **attack)** or of the nmap type (represented by the class NMAP_ADPTER). Concerning attacks, they can fall under the set consisting of sql injection (class Sql_Injection) , Ddos attack (class Ddos) or port scan (PortScan) . The client node seeking to connect to a server is identified using the class " client_connectort ") .

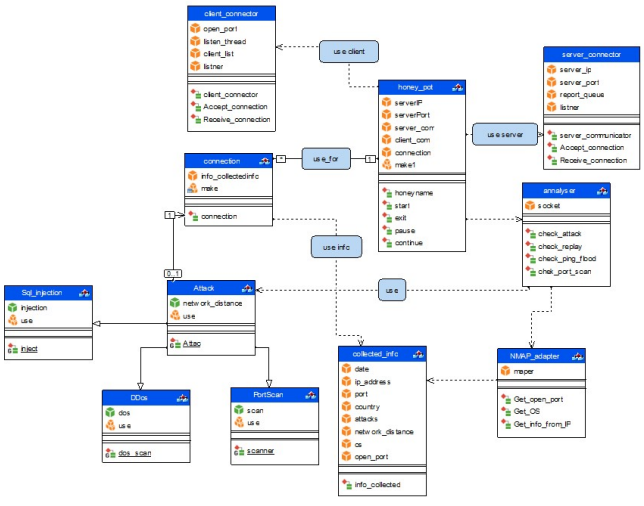

Figure 6. Data model from honeyPods

DISCUSSION

In this data proposing model work, for vulnerability detection, we have proposed two data models. The first relates vulnerability scanning operations in order to detect possible flaws in different applications. The second relates to the honeypot that can be deployed in big data environments in order to analyze and store the different behaviors of potential attackers. The hybrid nature of our proposal comes from the pooling of these first two models in order to obtain an element containing the functionalities and classes of the previous models. The advantage associated with this hybridization is obtaining a complete data storage model resulting from security operations. The implementation of this hybrid model in security tools could give them a high level of completeness compared to tools using one of the first two models, because it is incomplete.

Conclusion and perspectives

The objective of this work was to propose a hybrid data model to detect potential vulnerabilities in big data environments. To do this, we reviewed the various representative works to our knowledge. From these works we identified the various limitations and made our contribution which consists in associating a data model for storing data from vulnerability scans and a second model for storing data from the results of using honeypots.

Future work on our proposal could consist of:

- The implementation of an expert system whose knowledge base and premises would come from our proposed model.
- The implementation of a vulnerability detection architecture with components using data from our current model.

REFERENCES

1. De Mauro, A., Greco, M., & Grimaldi, M. (2016). A formal definition of Big Data based on its essential features. *Library review* .*
2. De Mauro, A., Greco, M., & Grimaldi, M. (2015, February). What is big data? A consensual definition and a review of key research topics. In *AIP conference proceedings* (Vol. 1644, No. 1, pp. 97-104). American Institute of Physics.
3. Ylijoki, O., & Porras, J. (2016). Perspectives to definition of big data: a mapping study and discussion. *Journal of innovation management* , *4* (1), 69-91.
4. Gärtner, B., & Hiebl, MR (2017). Issues with big data. In *The Routledge companion to accounting information systems* (pp. 161-172). Routledge.
5. HEMAIZIA, B. (2022). NoSQL data quality control.
6. BEDDA, I. (2018). Comparative study of the performance of NoSQL DBMSs Case study: MongoDB Vs Cassandra.
7. Bouzayani, H. (2012). *Quantitative model for intrusion detection: a collaborative architecture IDS-HONEYPOT* (Doctoral dissertation, University of Quebec in Outaouais).
8. Spitzner, L. (2002). Honeypots: Tracking hackers addison wesley professional.
9. Wicherski, G. (2006). Medium interaction honeypots. *German Honeynet Project* .
10. Goebel, J. (2006). *Advanced Honeynet based Intrusion Detection* (Doctoral dissertation, Master's thesis, RWTH Aachen University).
11. Ben Younes, R. (2007). Study and implementation of a method for detecting intrusions in 802.11 wireless networks based on formal model verification.
12. Saidi, S. (2016). *Contributions of Bayesian networks in intrusion detection systems* (Doctoral dissertation, Ibn Khaldoun-Tiaret University).
13. Lee, W., & Stolfo, S. (1998). Data mining approaches for intrusion detection.
14. Boudaoud, K. (2000). A multi-agent system for intrusion detection. *Proceedings of the Journées Doctorales Informatique et Réseaux (JDIR)* , 6-8.
15. Sajjaa, G.S., Pallathadka, H., Naved, M., & Phasinam, K. (2022). Various Soft Computing Based Techniques for Developing Intrusion Detection Management System. *ECS Transactions* , *107* (1), 3335.
16. Umer, MA, Junejo, KN, Jilani, MT, & Mathur, AP (2022). Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection* , 100516.
17. Akrout, R., Alata, E., Kaâniche, M., & Nicomette, V. (2014). Identification of Web vulnerabilities and generation of attack scenarios. *Journal of Information Sciences and Technologies-TSI Series: Technique and Computer Science* , *33* (9-10), 809-840.
18. Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010, May). State of the art: Automated black-box web application vulnerability testing. In *2010 IEEE symposium on security and privacy* (pp. 332-345). IEEE.
19. Halfond, W.G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the IEEE international symposium on secure software engineering* (Vol. 1, pp. 13-15). IEEE.

****