ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

An Efficient Intrusion detection system for network behaviors using Fuzzy logic based Rules

Sumathi M[1], Umarani R[2]

Department Of Computer Science, Mahendra Arts & Science College, Salem, Tamilnadu, India[1]

Department Of Computer Science,Sri Saradha College For Women, Salem-16, Tamilnadu, India[2]

Abstract: Internet services and web applications have become an inextricable part of daily life, enabling communication and the management of personal information from anywhere. Artificial Intelligence plays a driving role in security services. This paper proposes a dynamic model Intelligent Intrusion Detection System, based on Fuzzy rules based AI approach for intrusion detection. The techniques are modeled using fuzzy logic with network profiling that uses simple data mining techniques to process the network data. The proposed system that combines anomaly, misuse and host based detection. Simple Fuzzy rules allow us to construct if-then rules that reflect common ways of describing security attacks. For host and Network based intrusion detection use fuzzy rules and machine learning along with self organizing hash maps. Suspicious intrusions can be traced back to its original source path and any traffic from that particular source will be redirected back to them. Both network traffic and system audit data are used as inputs for both. Experimental results proves that system out performs other techniques.

Keywords: Anomaly detection, network behavior, intrusion detection, Data Centre Security

I. INTRODUCTION

Recently, have seen an interest in products that continuously monitor a database system and report any relevant suspicious activity [1]. Database activity monitoring has been identified by Gartner research as one of the top five strategies that are crucial for reducing data leaks in organizations [2], [3]. Organizations have also come to realize that current attack techniques are more sophisticated, organized, and targeted than the broad-based hacking days of past. Often, it is the sensitive and proprietary data that is the real target of attackers. Also, with greater data integration, aggregation and disclosure, preventing data theft, from both inside and outside organizations, has become a major challenge. Standard database security mechanisms, such as access control, authentication, and encryption, are not of much help when it comes to preventing data theft from insiders [4]. Such threats have thus forced organizations to reevaluate security strategies for their internal databases [5]. Monitoring a database to detect potential intrusions, intrusion detection (ID), is a crucial technique that has to be part of any comprehensive security solution for high-assurance database security. Policy matching is the problem of searching for policies applicable to an anomalous request. The main contributions of this paper can be summarized as follows: 1. Present model architecture for specifying intrusion response policies in the context of a DBMS. 2.Present fuzzy rules for administration of response policies.3.Present casual mapping to efficiently search the policy database for policies that match an anomalous request. 4. Use light weight virtualization technique with our response policy mechanism, and conduct an experimental evaluation of our techniques. The rest of the paper is organized as follows: Section 2 presents the related works of our intrusion detection and response policy methods. Section 3 presents the design and implementation of Fuzzy rules based intrusion detection system.Discuss the policy matching algorithms in Section 4. Section 5 discusses the implementation details of our response mechanism, and reports the experimental results concerning the overhead incurred by our techniques. Conclude in Section 6 with directions for future work.

II. RELATED WORKS

The concept of database response policies was first introduced by us [7]. The current paper is a major extension of our previous work. The policy matching algorithms in the current paper take into account arbitrary predicates while the scheme in [7] only considers equality predicates. Also, the JTAM policy administration model presented in this paper is a novel contribution. The concept of fine-grained response actions such as suspends, and taint has been introduced by us [8]. However, the work in [8] presents the design and implementation of an access control model that is capable of supporting such fine-grained response actions. First, since the approach is preventive, it requires fundamental changes to the existing access control mechanism of a DBMS. A subscription in a pub-sub system is similar to a response

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

policy, and an event is the anomaly detection event in our system.

III. PROPOSED SYSTEM

A. Construction of Multitier Anomaly detection system for class of Attacks:

Employ a lightweight virtualization technique to assign each user's web session to a dedicated container, an isolated virtual computing environment. In our system, an attacker can only stay within the web server containers that he/she is connected to, with no knowledge of the existence of other session communications. Thus ensure that legitimate sessions will not be compromised directly by an attacker.

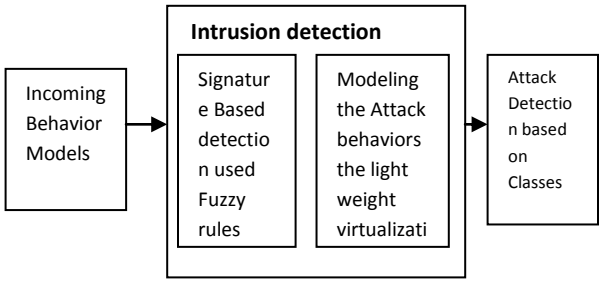

Fig 1: Function Flow Diagram of an Efficient intrusion detection System based On Fuzzy rules

IV. CONSTRUCTION OF NETWORK BEHAVIOR ANALYSIS MODEL

At the database side,Unable to tell which transaction corresponds to which client request. The communication between the web server and the database server is not separated, and can hardly understand the relationships among them. In typical three-tiered web server architecture, the web server receives HTTP requests from user clients and then issues SQL queries to the database server to retrieve and update data. Even if knew the application logic of the web server and were to build a correct model, it would be impossible to use such a model to detect attacks within huge amounts of concurrent real traffic unless had a mechanism to identify the pair of the HTTP request and SQL queries that are causally generated by the HTTP request .

Machine Learning Approach to Detect the Unknown Traffic using fuzzy rules.

Our architecture would have been similar to the conventional one where a single web server runs many different processes. Moreover, if the database authenticates the sessions from the web server, then each container connects to the database using either admin user account or non admin user account and the connection is authenticated by the database. By default, Double Guard normalizes all the parameters. Of course, the choice of the normalization parameters needs to be performed carefully.

Creation of normality models of isolated user sessions by a light- weight virtualization technique:

Due to their diverse functionality, different web applications exhibit different characteristics. Many websites serve only static content, which is updated and often managed by a Content Management System (CMS). For a static website.Build an accurate model of the mapping relationships between web requests and database queries since the links are static and clicking on the same link always returns the same information. However, some websites (e.g., blogs, forums) allow regular users with Non administrative privileges to update the contents of the server data. These queries cannot match up with any web requests, and keep these unmatched queries in a set NMR.

V. BUILDING A CAUSAL MAPPING PROFILE (MODELS) FOR TRAFFICS TO THE SERVERS FOR ESTIMATING THE INTRUDER

The same web request may result in different SQL query sets based on input parameters or the status of the webpage at the time the web request is received. For each unique HTTP request and database query, the algorithm assigns a hash table entry, the key of the entry is the request or query itself, and the value of the hash entry is AR for the request or AQ for the query, respectively. The algorithm generates the mapping model by considering all three mapping patterns that would happen in static websites.

VI. EXPERIMENTAL ANALYSIS

In our prototype,Chose to assign each user session into a different container; however, this was a design decision. For instance, Assign a new container per each new IP address of the client. In our implementation, containers were recycled based on events or when sessions time out.. Initially, Thus deployed a static testing website using the Joomla [10] Content Management System. In this static website, updates can only be made via the back-end management interface. This was deployed as part of our center website in production environment and served 52 unique webpages. For our analysis, Collected real traffic to this website for more than two weeks and obtained 1,172 user sessions.

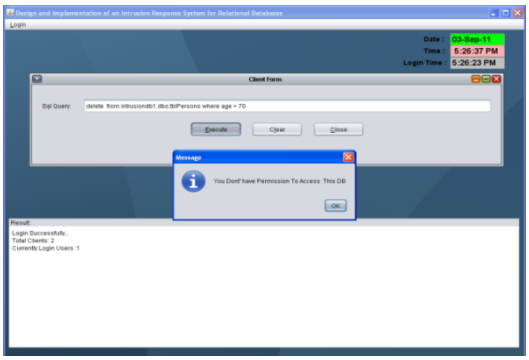

Fig 2: Experimental Result: denying action to the access the database through the rules matching analysis.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

In our deployment, site visitors were allowed to read, post, and comment on articles. Fuzzy rule models for the received front-end and back-end traffic were generated using these data. Thus discuss performance overhead, which is common for both static and dynamic models, in the following section.

VII. PRIVILEGE ESCALATION ATTACK

For Privilege Escalation Attacks, according to our previous discussion, the attacker visits the website as a normal user aiming to compromise the webserver process or exploit vulnerabilities to bypass authentication.

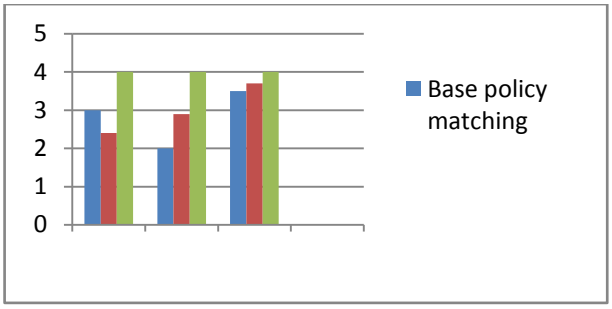

Fig 3: Experiment based on matching samples based rule types.

However, Double Guard separates the traffic by sessions. If it is a user session, then the requests and queries should all belong to normal users and match structurally.

TABLE I:

DETECTION RESULTS FOR ATTACKS

Intrusion operation	Snort	Green SQL	Fuzzy Based Model
Privilege Escalation	No	No	Yes
Web Server Aimed Attack	No	No	Yes
Sql Injection Attack	No	Yes	Yes
Direct DB	No	Yes	Yes

Most of these attacks manipulate the HTTP requests to take over the web server is shown in Table 1. As a second tool, Used Metasploit loaded with various HTTP based exploits. However, Fuzzy based model was able to detect these attack sessions. Here,Point out that most of these attacks are unsuccessful, and Fuzzy based model captured these attacks mainly because of the abnormal HTTP requests.

VIII. CONCLUSION

Firewall security, like any other technology, requires proper management to provide the proper security service. Thus, just having a firewall on the boundary of a network may not necessarily make the network any secure. The Firewall Policy Advisor presented in this paper provides a number of user-friendly tools for purifying and protecting the firewall policy from anomalies. The administrator can use the firewall policy advisor to manage a general firewall security policy without prior analysis of filtering rules. In this work,Formally defined all possible firewall rule relations and Used this to classify firewall policy anomalies.Then model the firewall rule information and relations in a tree-based representation. Based on this model and formalization, the firewall policy advisor implements two management tools:

REFERENCES

[1] SANS, "The Top Cyber Security Risks," top-cyber-security-risks/, 2011.
[2] B.I.A. Barry and H.A. Chan, "Syntax, and Semantics-Based Signature Database for Hybrid Intrusion Detection Systems," Security and Comm. Networks, vol. 2, no. 6, pp. 457-475, 2009.

[3] D. Bates, A. Barth, and C. Jackson, "Regular Expressions Considered Harmful in Client-Side XSS Filters," Proc. 19th Int'l Conf. World Wide Web, 2010.
[3] D. Bates, A. Barth, and C. Jackson, "Regular Expressions Considered Harmful in Client-Side XSS Filters," Proc. 19th Int'l Conf. World Wide Web, 2010.
[4] M. Christodorescu and S. Jha, "Static Analysis of Executables to Detect Malicious Patterns," Proc. Conf. USENIX Security Symp., 2003.
[5] M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna, "Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications," Proc. Int'l Symp. Recent Advances in Intrusion Detection (RAID '07), 2007.
[6] H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion-Detection Systems," Computer Networks, vol. 31, no. 9, pp. 805-822, 1999.
[7] V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna, "Toward Automated Detection of Logic Vulnerabilities in Web Applications," Proc. USENIX Security Symp., 2010.
[8] Y. Hu and B. Panda, "A Data Mining Approach for Database Intrusion Detection," Proc. ACM Symp. Applied Computing (SAC), H. Haddad, A. Omicini, R.L. Wainwright, and L.M. Liebrock, eds., 2004.
[9] Y. Huang, A. Stavrou, A.K. Ghosh, and S. Jajodia, "Efficiently Tracking Application Interactions Using Lightweight Virtualization," Proc. First ACM Workshop Virtual Machine Security, 2008.
[10] H.-A. Kim and B. Karp, "Autograph: Toward Automated Distributed Worm Signature Detection," Proc. USENIX Security Symp., 2004.
[11] C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), Oct. 2003.
[12] S.Y. Lee, W.L. Low, and P.Y. Wong, "Learning Fingerprints for a Database Intrusion Detection System," ESORICS: Proc. European Symp. Research in Computer Security, 2002.