High Assurance Architecture of Streaming Data Integration System about Security Threat Monitor

Aiping Li, Jiajia Miao, Yan Jia

Computer Institute, National University of Defence Technology,
410073, ChangSha, China
apli1974@gmail.com

Abstract. How to grasp the real-time overall situation of the network security is worth to study. An increasing number of network security systems have been deployed in the backbone and the gateways of enterprises, including various Netflow systems, IDS, VDS, VS and firewalls. These products make great contributions in enhancing the network security. However, current network security systems are independent and autonomous. Consequently, such solutions cannot figure out an overview of the network security situation. In another perspective, building a new global monitoring system will suffer from redundant construction and longer deploying time. We propose a novel and high assurance solution called GS-TMS which reuses the log data generated by the existing systems. Based on the data stream and data integration technologies, GS-TMS provides a desirable capability in quickly building a large-scale distributed network monitoring system. Furthermore, GS-TMS has additional notable advantages over current monitoring systems in scalability and flexibility.

Keywords: Data Stream, Threat Monitor System, Security Log

1 Introduction

Mitigating threats to networks have become one of the most important tasks of several governmental and private entities[1]. Intrusion detection systems (IDS), such as Snort[2], monitor all incoming traffic at an edge network's DMZ, perform TCP flow reassembly, and search for known worm signatures. Cisco's NBAR[3] system for routers searches for signatures in flow payloads, and blocks flows on the fly whose payloads are found to contain known worm signatures. There are two notable limitations in the above systems: (1)The Autonomy System Is Closed. IDS and other security products are autonomy systems, which can't share the monitor results with others. (2)The detection is on the lower level. Most IDS are on the enterprise level[4] but not the ISP level. When the IDS of enterprise A finding a bots net and informing the ISP, we can thoroughly block the source IPs of the bots net within the backbone. In the database research field, the independent data, which cannot be shared, is termed as "island". Data integration is a widely adopted solution to solve this problem.

The continuous growth of the network, coupled with the increasing number of the connected computers, poses more challenges to the network monitoring systems. (1)Data flow arriving at a high rate. (2) Data generating and monitoring tasks are continuous. (3)Real-time response is pivotal.

Examining the data stream applications in network security, we propose a novel technology to materialize a global view of the network security status based on existing applications, as illustrated in Fig. 1. Our system, called *Global Stream-based Threat Monitor System (GS-TMS)*, utilizes existing enterprise gateway security systems, such as IDS, firewall, DDoS protection systems and so on. We retrieve the logs from these systems as the input stream of DSMS. Based on the technologies of data stream and data integration, GS-TMS can provide the users with a unified interface to perform real-time monitoring and analyzing.

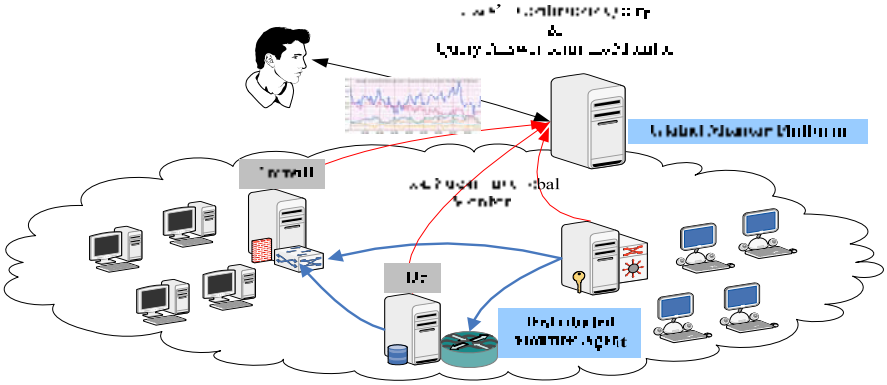


Fig. 1. The deployment of GS-TMS

From the perspectives of the implementation, constructing such a system faces the following challenges:

- How to automatically convert the existing security system logs to the input data stream?
- How to build the mapping between the heterogeneous data streams?
- How to design a suitable data stream query language for the network monitoring?
- How to solve the heterogeneity in query language level and data schema level?
- How to share and merge the results from the distributed heterogeneous systems?
- How to guarantee the validity of the input data streams from the distributed nodes?

The research on GS-TMS aims to help government make use of the existing security logs to monitor the network status efficiently. In the remainder of this paper, we proceed as follows: the related work is described in section 2. We describe the architecture of GS-TMS in Section 3. Next, we present the methods to verify our system. The conclusion is given in the last section.

2 Related Work

As the variety and the sophistication of attacks grow, early detection of potential attacks will become crucial in mitigation the subsequent impact of these attacks.

More recently, distributed network monitoring has received much attention from the research community. Projects such as Neti@Home[18], ForNet[19], DIMES [20] and Domino[21] all use agents on end systems to monitor network traffic, whether for intrusion detection and response or for straightforward network mapping and performance.

There are also many similar monitors deployed around the World. For example, the NCS plans to build GEWIS[22] (Global Early Warning Information System) around existing Internet performance tools integrated into a cohesive suite that can provide a top-level view of system performance. GEWIS monitor the performance of the Internet and provide warnings to government and industry users of threats that could degrade service, such as denial-of-service attacks against the DNS that control Internet traffic.

According to the related work mentioned above, we can draw a conclusion that there have been many studies focusing on the establishment of the overall threat monitoring system, but to our knowledge, most of them acquire network security events just by their own sensors which have been deployed by themselves. Such system cannot have high expansibility. Moreover, it is totally a national behavior, but not a federative behavior.

DSMS has been proposed to integrate data collection and processing of network streams in order to support on-line processing for various network management applications. The STREAM[23] and Gigascope[24] projects have made performance evaluations on their DSMSs for network monitoring. In both projects, several useful tasks have been suggested for network monitoring. Plagemann et al. have evaluated an early version of the TelegraphCQ[11] DSMS as a network monitoring tool by modeling and running queries and making a simple performance analysis.

Examining existing work, the data stream management technologies are become increasingly perfect, more and more network monitoring applications have adopted the data stream technologies. Therefore, there are two key challenging issues: how to integrate these distributed data management systems and how to provide a top-level security view? How to share the early warning information deployed in the different internal monitoring systems? The key to figure out these two questions is no other than the data stream integration technology.

3 The System Architecture

As illustrated in Fig. 2, GS-TMS comprises two primary components: Global Monitor Platform (GMP) and Distributed Monitor Agent (DMA). GMP is mainly responsible for interacting with users, as well as merging the query rewriting, results and other tasks; DMA is in charge of the log conversion, i.e., executing the specific query tasks from the GMP. We describe the GS-TMS architecture with two stages: initializing phase and processing phase. In the initializing stage, the main task is: 1) after the agents have been installed in the distributed nodes, DMA will convert local log database to the stream input of Light DSMS. Firstly, the log to stream module automatically extracts an entity relationship schema from the local log database through reverse engineering. Secondly, it analyses the main log table, and convert it

to stream data as the input of Light DSMS. 2) Schema mapping module takes the global schema and local schema as input, then outputs the mapping results. Then, this mapping result will be submitted to Global Monitor Platform.

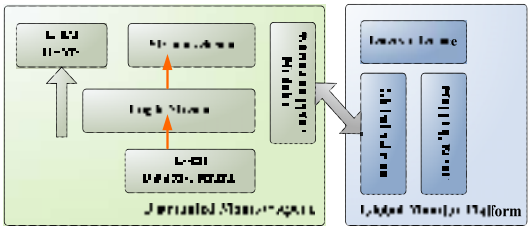


Fig.2. The initializing stage of GS-TMS

As shown in Fig. 3, in the processing stage, the main task is: 1) Accepting the user's submission of query, executing the query plan (for minimizing communication costs), query rewriting according to the results of schema mapping and decomposing query into the specific monitoring agents; 2) When DMA gets the specific query sentence, it will analyze the log stream, and then returns the query answer to GMP; 3) Result Merge Module merges the query answers from different nodes according to the schema mapping results and specific policies, and then presents the final query answers to users; 4) The attack events, which were detected by DMA will be shared among all the other distributed nodes.

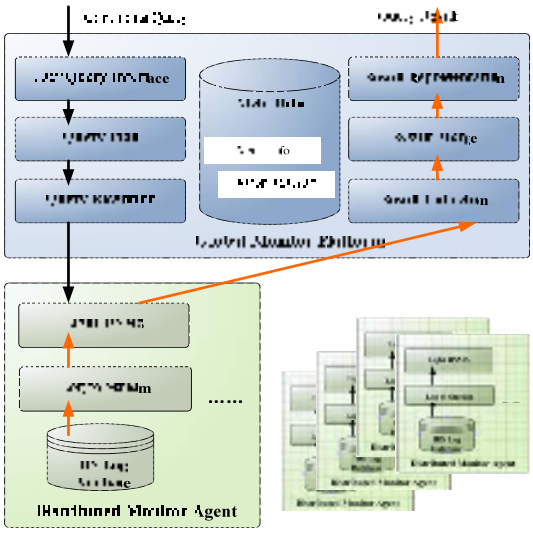


Fig.3. The processing stage of GS-TMS

GS-TMS can coexist with legacy security systems, reusing the existing security log database. Existing security systems, such as IDS, firewall and DDoS detecting systems have a large number of attack-warning logs[5][6], but these logs are only available to local users, as a time log is also the basis of the analysis. GS-TMS also

gives support to convert the logs of these legacy systems to the local stream input of DMA. This enables the log information sharing and data stream processing.

As shown below, that is the database schemas of Snort V1.06[2]. As shown in Fig. 4, we can observe that: 1) Table event with timestamp is the core of these tables. 2) Table iphdr includes source IP address and destination IP address. So, we can draw the following laws:

1. The core table must include timestamp field.
2. The core table will continuously increase larger with the time going on.
3. The main table, which contains the basic information should has a high degree and has the same keys as the core table.

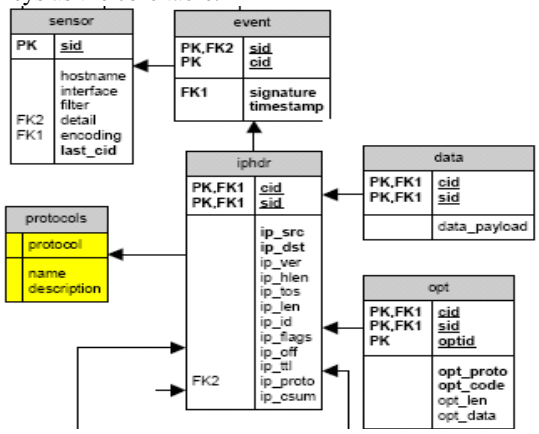


Fig 4. The snort schema v1.06

Based on the above analysis, we must firstly extract an E-R graph from a relational database through reverse engineering [7], and then utilize the matrix compute to find the highest degree of the tables.

A fundamental operation in the manipulation of schema information is matching, namely, taking two schemas as the inputs and producing a mapping between the elements of the two schemas that correspond semantically to each other[8].

Following the case we motioned above, we have the some observations:1)The scale of the input schema should be as less as possible to make the match in our system simple.2)The data instances of that case should be organized in fixed format. In GS-TMS matching is relatively simple in schema scale. We consider of the specific application, using a hybrid algorithm to improve the performance, with high recall rate and high precise rate.

These data management systems adopt a variety of query language, such as CQL[9], TQL[10], TelegraphCQ[11] language and so on. The CQL is an expressive SQL-based declarative language for registering continuous queries against streams and stored relations. Taking into account we build our query language prototype based on CQL.

We address the problem of query rewriting in global-as-view data integration systems. User queries are formulated over the global schema, and the system suitably

queries the sources, providing an answer to the user, who is not obliged to have any information about the sources.

Many techniques haven been provided to solve the heterogeneous schema issues[12]. In other word, system interpreted the user-input query, which according to the global schema, to some specific query, which according to the local schema. In GS-TMS, we need to query rewriting module work in two-level heterogeneous: 1)Heterogeneity between the global schema and the local source schema. 2)Heterogeneity between the different DSMS.

In a dynamic environment, sources frequently change their query interfaces, data formats[13]. Hence, once the system is deployed, the administrator has to monitor it over time to detect and repair the broken mappings. Now such continuous monitoring is well-known to be extremely labor intensive. Hence, developing techniques to reduce the maintenance cost is critical in practice. We proposed the Detecting Broken Mappings Based on Fuzzy Reasoning module, which improve the correct ratio of checking invalidation mapping[14].

4 Evaluation Design

The purpose of simulation experiment is to verify the functionality of the system, and evaluate the system performance both in the initializing phase and the processing phase. In the initializing phase, we select the most popular IDS system in China, like TopSec[15], ICEYE[16], Snort[2] to construct the testbed. These produces occupy over 80% of the market quotient. We focus the performance of converting log database to data stream and the precise of schema mapping algorithm here. We use *Precision* to describe the performance of converting log data to stream, *Recall* to express the precise of schema map algorithm, which is as :

$$Precision = \frac{total\ convert\ data}{total\ data} \quad Recall = \frac{correct\ convert\ data}{total\ convert\ data}.$$

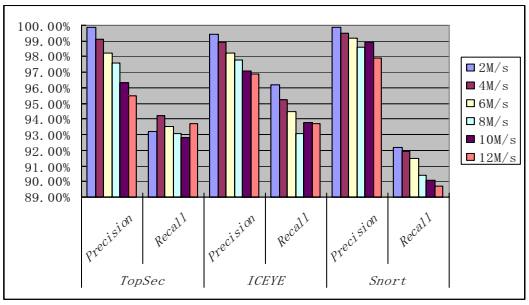


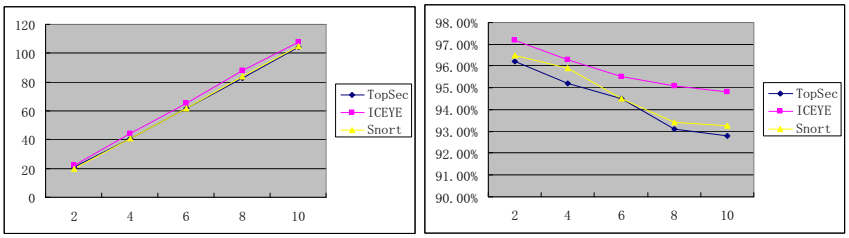
Fig. 5. The initializing phase experiment result

We simulated the log data produced speed from 2M/s to 12MB/s. As shown in Fig. 5, all the *Precision* of legacy system log data is above 95%, we think the result is good for the legacy systems. What's more, the *Recall* of each system is fairly good,

almost above 90%, the reason we think maybe the meta map data has some problem, we will try again to improve the recall ratio of each system.

In the processing phase, we use the log data from Dshield, due to its data is more comprehensive than the log that we can provide. Here we focus how to reduce the communicate costs between GMP and DMAs and how to improve the precise of query rewriting algorithm. Follow the above description, we use *Cost* express the communicate costs and *QPrecision* to express the query precision. We get the experiment result in Figure 6. The Cost is described in milliseconds, where the experiment environment is 100M/s. The *QPrecision* is described as:

$$QPrecision = \frac{correct\ query\ result}{total\ data}$$



(a) *Cost* result (b) *QPrecision* Result

Fig. 6. The processing phase experiment result

As shown in Fig. 6, the communication cost is almost line increase along the data stream increase, this is fairly good. What's more, the extra commutation cost is very small as shown in Fig 6(a). The *QPrecision* of each system is also above 92%, and the decrease of *QPrecision* is not very fast, and when the data scale is increase, the *QPrecision* tends to the constant. The result has shown the architecture are good.

The purpose of the experiments is only to show the algorithms and architecture we given is logical. The optimization of the algorithms will be done in the future work.

5 Conclusion

With the rapid increasing of the internet attack events, it is impending to have a Global Early Threat Monitor System. In this paper, we propose a comprehensive solution for Global Threat Monitor System. There are a number of components in the solution, and each of them is also a research issue in this area. Our main contribution in this paper is to present a set of new methods for integration distributed security system and provide a global view of the security status. Moreover, we adapt the legacy systems to our stream-based agents, so we can deal with the attack events more promptly.

There are still many problems need to be solved, including reducing communication costs, providing more supports for different operators in language, etc. Finally, real-world environment experiment to test the system is needed.

Acknowledgements. This work is supported by the National High-Tech Research and Development Plan of China ("863" plan) under Grant No. 2006AA01Z451, No. 2007AA01Z474 and No. 2007AA010502.

References

1. Technical Cyber Security Alerts, http://www.us-cert.gov/cas/techalerts
2. The de facto standard for intrusion detection/prevention, http://www.snort.org/.
3. NBAR System, http://www.cisco.com/en/US/products/ps6616/
4. Siegel C.A., Sagalow T.R., Cyber-Risk Management: Technical and Insurance Controls for Enterprise Security, Information Systems Security, vol.11(2002)
5. Kandula S., Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds, 2nd Symposium on Networked Systems Design and Implementation (2005)
6. Kim H.A., Karp B., Autograph: Toward Automated, Distributed Worm Signature Detection, USENIX (2005)
7. Anderson M., Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering, pp.13--16, IEEE Press, Manchester(1994)
8. Bernstein P.A., Melnik S., Mork P., Interactive schema translation with instance-level mappings, Proceedings of the 31st VLDB, pp. 12--83 (2005)
9. Arasu A., Babu S., Widom J., The CQL continuous query language: semantic foundations and query execution, The VLDB Journal, vol. 15, pp. 121--142(2006)
10. Barbara D., The Characterization of Continuous Queries, International Journal of Cooperative Information Systems, vol. 8, pp. 295--323(1999))
11. Chandrasekaran S., TelegraphCQ: continuous dataflow processing, Proceedings of the 2003 ACM SIGMOD, pp. 668--668, ACM, California (2003)
12. Cali A., Lembo D., Rosati R., Query rewriting and answering under constraints in data integration systems, Proc. of the 18th IJCAI, pp. 16-–21 (2003)
13. McCann R., Mapping maintenance for data integration systems, Proceedings of the 31st VLDB, pp. 1018--1029(2005)
14. Miao J. et al., Detecting Broken Mappings for Deep Web Integration, Semantics, Knowledge and Grid, Third International Conference, pp. 56--61(2007)
15. TopSentry product introduction, http://www.topsec.com.cn/products/ids.asp
16. ICEYE product introduction, http://www.nsfocus.com/1_solution/1_2_2.html.
17. The annual report 2007, http://www.cert.org.cn/.
18. Simpson Jr C.R., NETI@ home, Software on-line: http://neti. gatech. edu, (2003)
19. Shanmugasundaram K. et al., ForNet: A Distributed Forensics Network, Computer Network Security, 2nd International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS, Petersburg (2003)
20. The DIMES project, http://www.netdimes.org/new/.
21. Yegneswaran V., Barford P., Jha S., Global Intrusion Detection in the DOMINO Overlay System, Proceedings of NDSS(2004)
22. Feds planning early-warning system for Internet, http://www.computerworld.com/
23. Arasu A. et al., STREAM: The Stanford Data Stream Management System, a book on data stream management edited by Garofalakis, Gehrke, and Rastogi(2004.)
24. Cranor C. et al., Gigascope: a stream database for network applications, Proceedings of the ACM SIGMOD, pp. 647--651(2003.)