GS-TMS: A Global Stream-based Threat Monitor System

Jiajia Miao
Ph.D. candidate
supervised by Quanyuan Wu
National University of Defense Technology
Changsha, China
+86-13787205177
jjmiao@nudt.edu.cn

ABSTRACT
Computer networks have become ubiquitous and integral part of the nation's critical infrastructure. How to grasp the real-time overall situation of the network security is very noteworthy to study. Current network security systems make great contributions in enhancing the network security. Nevertheless, these products are independent and autonomous, so they fail to share the results of the detected attacks. Consequently, such solutions cannot figure out an overview of the network security situation. In another perspective, building a new global monitoring system from scratch will suffer from redundant construction, more cost, and longer deploying time. To address the dilemma, we propose a novel solution called *GS-TMS* which reuses the log data generated by the existing widely-spread security systems. By introducing the data stream and data integration technologies, GS-TMS provides a desirable capability of quickly building a large-scale distributed network monitoring system. Furthermore, GS-TMS has additional notable advantages over current monitoring systems in scalability and flexibility.

1. INTRODUCTION
The Internet is now regarded as an economic platform and a vehicle for information dissemination at an unprecedented scale to the world's population. But this success has also enabled hostile agents to use the Internet in many malicious ways [1], and terms like spam, phishing, viruses, self propagating worms, DDoS attacks, etc. Hence, mitigating threats to networks have become one of the most important tasks of several governmental and private entities.

1.1 Limitations of the current systems
Intrusion detection systems, 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] for routers searches for signatures in flow payloads, and blocks flows on the fly whose payloads are found to contain known worm signatures.

PVLDB '08, August 23-28, 2008, Auckland, New Zealand
Copyright 2008 VLDB Endowment, ACM 978-1-60558-306-8/08/08

From the angle of the nation, they care about the security of not only the global network but also the specified area; for example, the government is concerned with how to protect the safety of the certain server group[1], including the tickets booking server, video server, web server and so forth. We will take this case as the background of our system.

Then, there are two notable limitations in the above systems:

- **The autonomy system is closed.** IDSes and other security products are autonomy systems, which cannot share the monitor results with each other. For example, even though the IDS of *enterprise A* detects a paragraph DDoS attack, the IDS of *enterprise B* might fail to get this situation in time. Therefore, *enterprise B* will be compromised by the same attack.
- **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 *'isolated island'*. Data integration is a widely adopted solution to solve this problem. This guides us to build an integrated network security platform to share the security information.

1.2 Requirements
The continuous growth of the network, coupled with the increasing number of the connected computers, poses more challenges to the network monitoring systems:

- **Data arriving at a high rate.** Such security systems monitor variety of continuous data that may be characterized as unpredictable and arriving at a high rate, including both packet traces and network performance measurements. For example, the ISP with high-speed switches, the data flow comes up to 40 Gbit/s [5].
- **Data generating and monitoring tasks are continuous.** In the network monitoring system of a large network, e.g., the backbone network of an ISP, the query results with the latest monitoring data changes constantly. So the query is not a one-shot query, but a continuous query.
- **Real-time response is pivotal.** With the development of computer technologies, hackers and virus technologies also improve rapidly. The attacks will do great harm due to the delayed

[1] Due to the sensitive nature, here we use *'www.xxx.com'* to denote the specified server group.

protecting responses. Therefore, the monitoring system requires real-time data processing.

Conventional DBMS's are deemed inadequate to provide the kind of online continuous query processing that would be most beneficial in this domain. A data stream system that could provide effective online processing of continuous queries over data streams would allow network operators to install, modify, or remove appropriate monitoring queries to support efficient management of the ISP's network resources [6].

Examining the data stream applications in network security, we come up with a new idea: the security log data can serve as the input of Data Stream Manager System (DSMS) to analyze the network security events in real time.

1.3 Our solution

As a summary, we propose a novel technology to materialize a global view of the network security status based on existing applications, as illustrated in Figure 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.

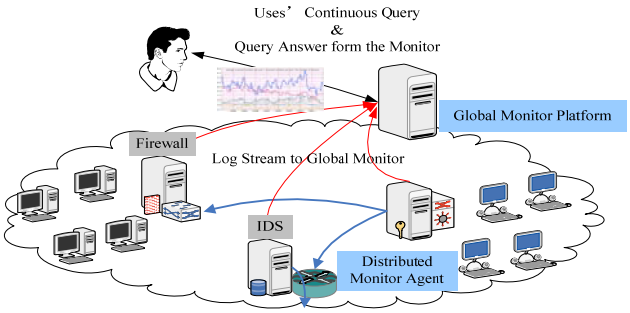


Figure 1. The deployment of GS-TMS.

My PhD thesis will focus on key technologies in building the GS-TMS system. In this paper, a comprehensive solution is presented and my current and future research works in this area is discussed.

In the remainder of this paper, we proceed as follows: in section 2, we catalog the goals of GS-TMS. In Section 3, we describe the architecture of GS-TMS. Next, in Section 4, we present the methods to verify our system in the artificial environment and the real-world environment. After describing related work in Section 5, we conclude our work in the last section.

2. DESIDERATA FOR GS-TMS SYSTEM

Continuous. From the user point of view, the query answers returned from GS-TMS is continuous instead of one-shot query. Accordingly, the feedbacks which users desire to retrieve from our system are continuously changing over time.

Automatic & Robust. To be adapted to the Internet's distributed and dynamic characteristics, our system should automatically deal with the joining and leaving requests of the distributed monitoring nodes. The automation characteristic of our system consists of two sub-requirements: automatically converting the log database into the data stream input and detecting the failure nodes.

Transparent. To eliminate the heterogeneity of the distributed nodes, all the heterogeneous systems which need to provide the unified upper interface to users should construct a map between the global view and the local view of the distributed nodes.

Efficient. Our system is set up in the distributed networks. Consequently, GMP should communicate with all of the distributed agents frequently. It is necessary to reduce the communication costs. Monitoring node is high-load, so it's also important to minimize the computing resource consumption of our agents.

Simple. We must predigest the user interface, including the query language and the results presentation styles, to make the system as simple as possible. This will improve the flexibility of our system to enable users customize their own enquiries according to different needs.

3. THE SYSTEM ARCHITECTURE

GS-TMS is a platform with new information integration technique which couples the log databases with the stream management technology. GS-TMS employs the log databases as stream inputs. The most prominent advantage of GS-TMS is that it is able to handle the security events timely and equip the underlying systems with a new capability of sharing the log data.

The architecture for GS-TMS is illustrated in Figure 2 (the Initializing phase) and Figure 3 (the processing phase). 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 issued from the *GMP*.

As shown in Figure 2, in the initializing phase, the outlined procedure is: 1) after the agents have been installed in the distributed nodes, DMA will convert the local log database to the stream input of *Light DSMS*. 2) Schema mapping module takes the global schema and local schema as input, then outputs the mapping results. Also, the mapping result will be submitted to GMP for query rewriting and result merging.

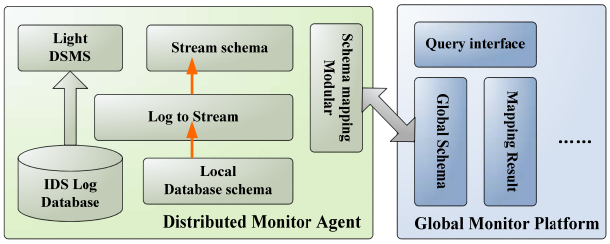


Figure 2. The initializing stage of GS-TMS.

As illustrated in Figure 3, in the processing phrase, the main tasks are: 1) Accepting the user's continuous queries, executing the query plan, performing the query rewriting operations according to the schema mapping results and decomposing the queries into the specific monitoring agents; 2) When DMA gets the specific query, it will process the log stream, and returns the query answer to GMP; 3) *Result Merge Module* merges the query answers obtained from different nodes following the schema mapping results and specific policies, and then presents the final query answers to users; 4) The attack events, which were detected by DMAs will be shared among all the other distributed nodes.

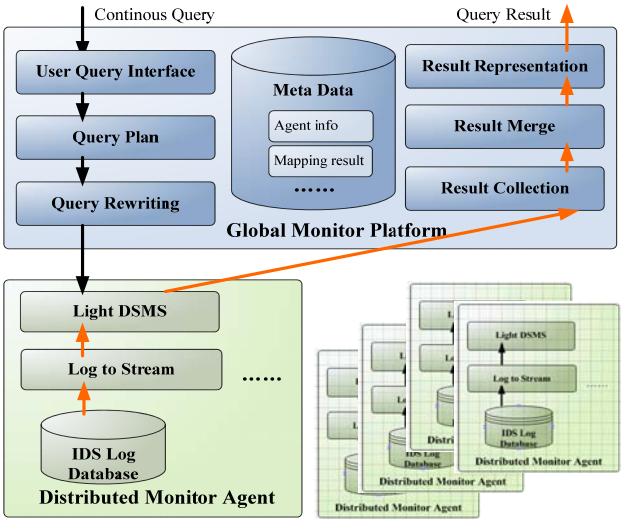


Figure 3. The processing stage of GS-TMS.

3.1 The global schema

We construct the GS-TMS architecture following the GAV approach, so the global schema is defined in terms of the sources. Following the case mentioned in Section 1, we design several network monitoring query examples, as shown in Figure 4. Query (a) finds all <*source*, *destination*> pairs that transmit more than 1000 packets for two 10-second windows in a row, where destination is *'%xxx.com%'*, in order to catch the abnormal network traffics, while query (b) aggregates attack events by *'attack_type'*, updating per 5 seconds. Especially, we can find out the top100 IPs, which are considered as the SYN flood attacks.

Based on our understanding of the needs of network monitoring, we design the schema of data stream and relation tables. As shown in Figure 4(c), stream table 'Packets' is a 5-tuple, <*src_addr*, *dest_addr*, *length*, *attack_type*, *ts*>. There are also relational tables, like *Whois*, *Attack*, etc.

```
with
    Elephants as
        select P.src_addr, P.dest_addr, count(*)
        from Packets P [range '10 sec' slide '10 sec']
        group by P.src_addr, P.dest_addr
        having count(*) > 1000
(select P.src_addr, P.dest_addr, count(*)
        from Packets P [range '10 sec' slide '10 sec'],
        Elephants E [range '10 sec' slide '10 sec']
        Whois W
        where P.src_addr = E.src_addr
        and P.dest_addr = E.dest_addr
        and W.name LIKE '%xxx.com%'
        group by P.src_addr, P.dest_addr
        having count(*) > 1000);
```

(a) Find all <source, destination> pairs that transmit more than 1000 packets for two 10-second windows in a row, where destination is '%xxx.com%'

```
select P.src_addr, P.dest_addr, P.attack_desc
    from Packets P [range by '30 sec' slide by '5 sec']
    group by P.attack_type;
```

(b) Compute an attack matrix (aggregate attack by attack type) updating every 5 seconds.

```
-- Stream of IP header information
-- The 'inet' type encapsulate a 32-bit IP address
create stream Packets ( src_addr inet, dest_ddr inet,
        length integer, attack_type integer, ts timestamp)
        type unarchived;
-- Table of WHOIS information
create table Whois (min_addr inet, max_addr inet,
        name varchar);
-- Table of attack information
create table Attack (attack_id integer, attack_type integer,
        attack_name varchar);
```

(c) SQL's Data Definition Language for create streams and tables

Figure 4. Sample query and global schema

3.2 Log to stream

One advantage of GS-TMS is that it can coexist with legacy security systems, viz., reusing the existing security log databases. Existing security systems generated a large number of attack-warning logs [7][8], but these logs are only available to the local users, and just for the purpose of statistics. GS-TMS also gives support to convert the logs of these legacy systems to the local stream input of DMAs. This advantage enables the log information to share and data process timely.

We present our idea with two cases, such as *Cybervision*[2] IDS and *'863-917'*[3] platform's log conversion process. As shown in Figure 5, that is the database schemas of Cybervision and '863-917' platform. GS-TMS adopts the logs from '863-917' and IDS to analyze the security situation of the *'xxx.com'* server group timely.

As presented in Figure 5, we can draw the following conclusions: 1) Table *'event'* with *'timestamp'* is the core of these tables. 2) Table *'iphdr'* and *'netIDS_Eventlog'* include source IP address and destination IP address.

Therefore, we can draw the following laws:

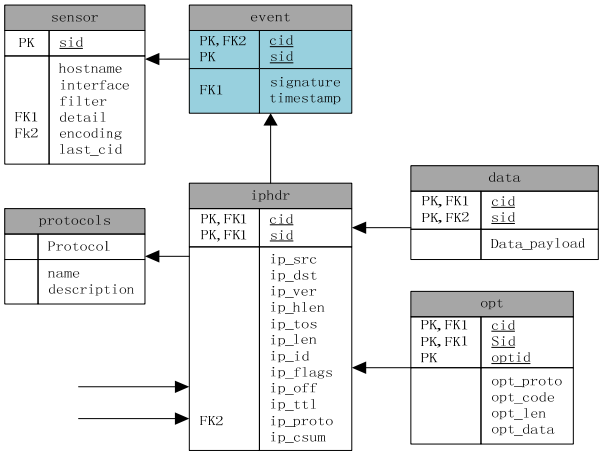
1) The core table must include timestamp field.

2) The core table will continuously increase larger with the time going on.

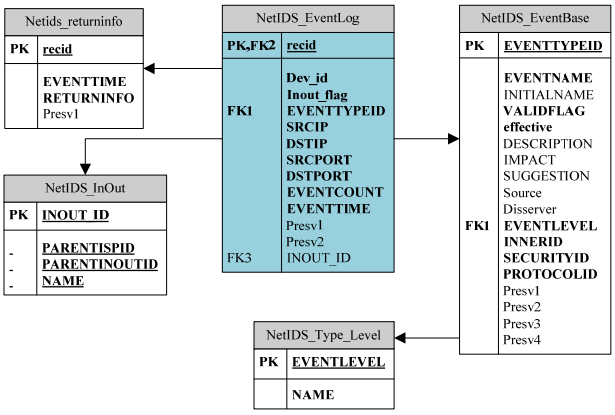
3) The main table, which contains the basic information (for example, IP info, etc.) should have a high degree and have the same keys as the core table.

[2] The leading IDS product in China. http://www.venusense.com/

[3] The '863-917' platform is a distributed networking monitor system, which deployed by CNCERT/CC. This system catches the Internet packets in backbone networking and logs these data for further analysis job.



(a) schema of Cybervision



(b) schema of '863-917'

Figure 5. The local source schema.

Examining the above analysis, we should firstly extract an E-R graph from a relational database through the reverse engineering technology [9], and utilize the matrix computing to work out the highest degree of the tables. Finally, we will figure out the optional core tables and main tables.

3.3 Schema matching

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 [10].

We will take the global schema and local schema as the inputs. DMA then gives the mapping between two schemas as the outputs. These results are stored in GMP for future query writing and result merging. As illustrated in Figure 6, the log stream schema of the '863-917' and Cybervision can provide the data for global view.

Following the case we discussed above, we have 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. As an illustration, the source address and destination address are supposed to be formatted in form of 'XXX.XXX.XXX.XXX'.

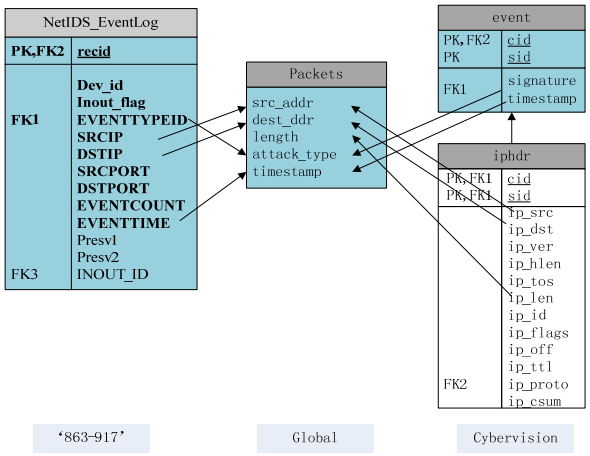


Figure 6. The schema mapping between global and local.

In GS-TMS, matching is relatively simple in schema scale. We consider of the specific application, using a hybrid algorithm [10] to improve the performance, with high recall rate and high precise rate.

3.4 Query Rewriting

In our system, 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.

Previous research works have proposed many techniques to solve the heterogeneous schema issues [11]. 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.

Reviewing the case we mentioned in Figure 4(b), we will illustrate the query rewriting process in Figure 7 as follows:

```
select H.ip_src, H.ip_dst, E.signature
        from event E [range by '30 sec' slide by '5 sec']
        iphdr H
        group by h.signature;
```

(a) Query execute in Cybervision

```
select N.SRCIP, N.DSTIP, N.EVENTYPE
        from NetIDS_EventLog N
                [range by '30 sec' slide by '5 sec']
        group by N.EVENTYPE;
```

(b) Query execute in '863-917'

Figure 7. The query in local DSMS.

In future work, we will extend query rewriting module to deal with two-level heterogeneous: 1) Heterogeneity between the global schema and the local source schema. 2) Heterogeneity between the different light DSMSs.

3.5 Other modules

Broken Mappings Detect. In a dynamic environment, the sources frequently change their query interfaces and data formats [12]. Such changes often invalidate the semantic mappings and cause system failures. Hence, once the system is deployed, the administrator has to monitor it over time to detect and repair the broken mappings. Today, such continuous monitoring is well-known to be extremely labor intensive. Thus, developing new techniques to reduce the maintenance cost is critical for the widespread deployment of data integration systems in practice. We propose the Detecting Broken Mappings Based on Fuzzy Reasoning module, or DBMFR for short), which definitely improves the correct ratio of the checking invalidation mapping [13].

Attack Events Feedback. The lack of network data sharing among organizations impedes the cooperation in network defense; as the attacks typically cross organizational boundaries, an effective prevention requires the defenders to look beyond their own perimeter in cooperation with other organizations. In our system, the lower agents can not only provide the data for build the global view, but also access to the upper results feedback. The agents can improve local security by sharing the attack events feedback from other nodes. This module shares the attack events between different organizations.

4. Initial Experiments

In this section, we will present the experiments on real network traffic data to evaluate the function of our system. GS-TMS is part of the *'xxx network security situation analysis and display system'*, which is designed to maximize the availability and reliability of *'xxx.com'*. Firstly, we present our experimental setup, then experimentally show that our system always satisfy the requirements discussed in Section 1.2.

4.1 Experimental Setup

As illustrated in Figure 8, we deployed DMAs in two legacy systems, Cybervision and '863-917' platform. These two security systems monitor the network traffics and security events about three servers: file server, web server and application server. DMAs translate these log databases to the data stream, i.e., the input of light DSMS, and execute the query order from GMP. Then DMAs product the query answers, and return them back to GMP. At last, GMP merges the distributed query results and shows them to final users timely by broken line graph.

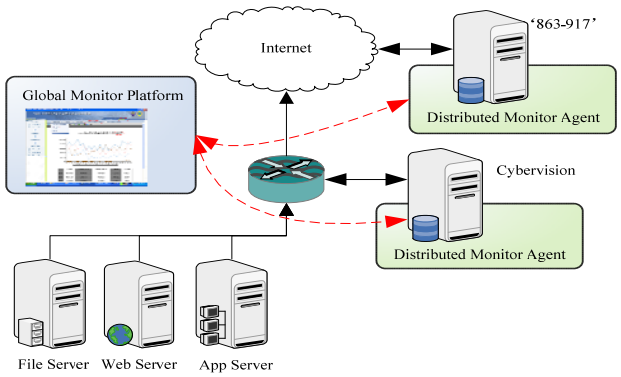


Figure 8. The diagram of the deployment.

4.2 Function evaluation

The system design objectives and main functions have been realized, as shown in the screenshots. Next, we will illustrate each figure with the design goals of each query.

Firstly, we give the query as shown in Figure 9(a). We want to find out the top 5 attacks behaviors, and try to perform corresponding responses. Figure 9(b) shows the attack counts till current time, wherein x-axis is the real time while y-axis reflects the numbers of security events.

```
select count(*)
        from Packets P [range by '5 seconds' slide by '5 seconds'],
        Whois W
where P.dest_addr >= W.min_addr
        and P. dest_addr < W.max_addr
        and W.name LIKE '%xxx.com%';
group by P.attack_type;
```

(a) Query for top 5 attacks

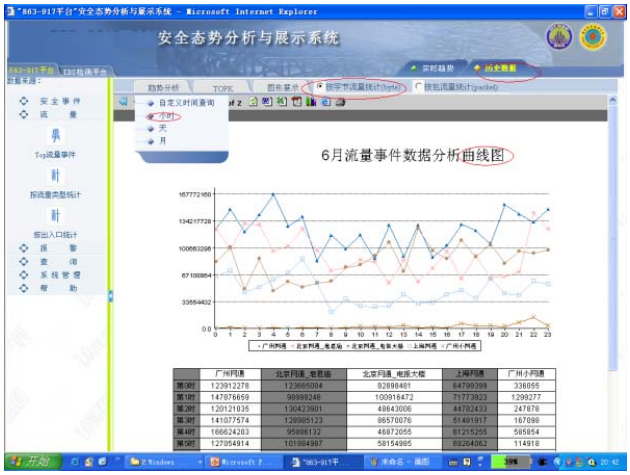
(b) The interface of query(a) answer

Figure 9. Query and the interface of query answer 1.

Administrators also care about the abnormal network traffics, so we give a query in Figure 10(a). The query is invoked to get the top *n* network traffics, which are connected with xxx.com. Figure 10(b) shows the cumulated throughout.

```
select count(P.length)
        from Packets P [range by '5 seconds' slide by '5 seconds'],
        Whois W
where P.dest_addr >= W.min_addr
        and P. dest_addr < W.max_addr
        and W.name LIKE '%xxx.com%';
group by P.src_addr;
```

(a) Query for top n network traffics



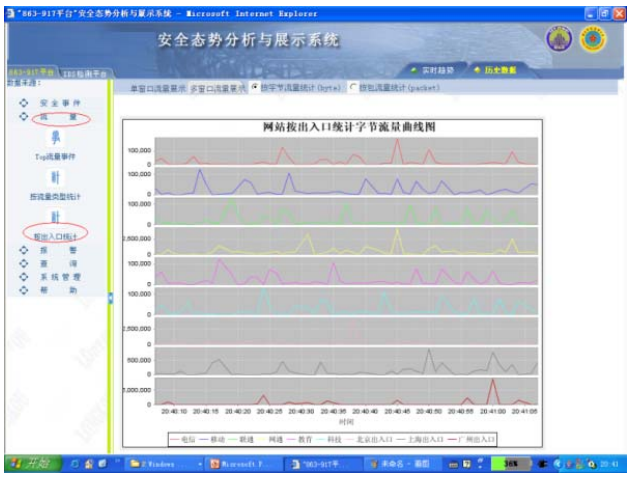
(b) The interface to show Query(a) answer

Figure 10. Query and the interface of query answer 2.

At last, we give a query to catch the network traffic from the IPs, which is also the user concern. If there are some abnormal events, administrators can block the IPs immediately. Figure 11(b) illustrates the throughout change trends for every specified IPs.

```
select count(P.length), P.src_addr
        from Packets P [range by '5 seconds' slide by '5 seconds'],
        Whois W
where P.dest_addr >= W.min_addr
        and P. dest_addr < W.max_addr
        and W.name LIKE '%xxx.com%'
        and (P.src_addr LIKE 'XXX.XXX.XXX%' or
          P.src_addr LIKE 'XXX.XXX.XXX%' );
```

(a) Query for network traffic of specific IPs



(b) The interface to show Query(a) answer

Figure 11. Query and the interface of query answer 3.

These figures prove that we can reuse the log data to avoid the repeated developments. With the data stream technology, user can define the query directly suites for the monitor demands. In addition, the query answers are continuous and change with the time, so the administrator can tackle the security events in time.

4.3 Performance evaluation

IDSes monitor all incoming traffic, perform TCP flow reassembly, and search for the known worm signatures. So IDS system is high CPU and memory overhead expenses. It's important to compare the performance of IDS without DMA deployed with the performance of IDS with DMA deployed. The test can evaluate the performance overhead which is exacted to the IDS systems by deployment of DMA. We adopt the standard samples of the different attacks to simulate the network traffic in different size of packet.

Cybervision server's configuration is as follows: CPUs are double XEON 2.4, memory is 2G, NIC is 1000M, and storage is DOM with 128 M. Firstly, we care about the CPU costs with different packets scale.

Also as shown in Figure 12, x axis denotes the different bandwidth occupancy rate, while the y axis represents the IDS' CPU occupancy rate. The s-resize means with DMA, and n-resize means without DMA. Then the dotted line between them represents the CPU rate of DMA. As shown in Figure 12, with the increasing of the bandwidth occupancy rate, the CPU utilization rate as high as 3% to 15% accordingly. But the increase is still within an acceptable range.

Also as shown in Figure 13, x axis represents the different bandwidth occupancy rate, while the y axis denotes the IDS' attack detection precision. From left to right side, each column means the different package size of 61/128/512/1518 bytes. And the white areas denote the attack detection precision with DMA, while the solid areas represent the precision without DMA. Obviously, we can see that the precision of detecting decreased, while we deployed DMA in IDS server.

Also as shown in Figure 14, x axis represents the different number of connect per second, while the y axis denotes the IDS' attack detection precision. From left to right side, the columns in a group mean HTTP connections and TCP connections. And the blank areas represent the precision of detecting with DMA, while the solid areas represent the detecting without DMA. Clearly, with the increasing of the connections, the accuracy has a different degree of decline.

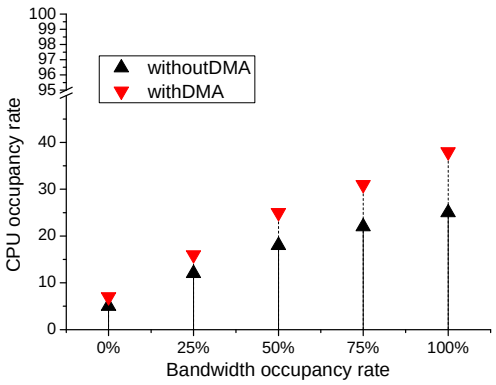


Figure 12. The CPU occupancy rate with different bandwidth.

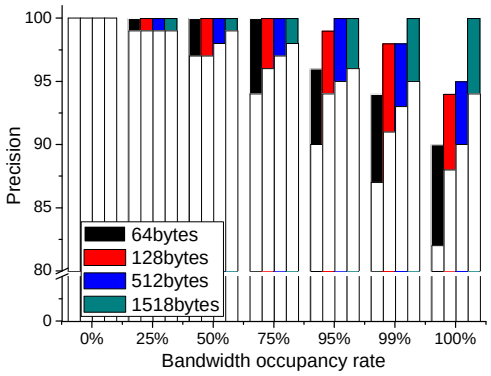


Figure 13. The precision rate with different bandwidth rate.

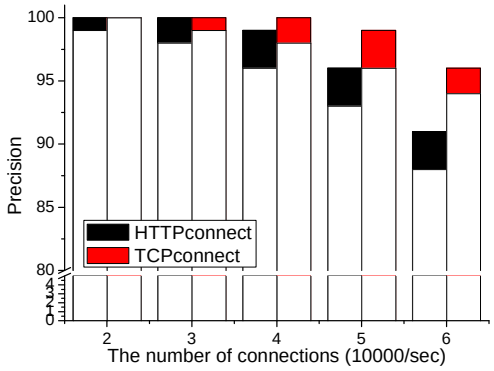


Figure 14. The precision rate with different data arrival rate.

In conclusion, the DMAs will inevitably increase the load of network monitoring server, like IDS, firewall, etc. Also the detecting precision of IDS systems is decreased. So we will do more research work about how to reduce the overload efficiently, including building the effective synopsis of data stream to enhance the query process, and so on.

5. RELATED WORK

5.1 Existing network monitor system

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

More recently, distributed network monitoring has gained more attention from the research community. The projects, exploring Neti@Home [14], ForNet [15], DIMES [16] and Domino [17] all use agents on the end systems to monitor network traffic, whether both for intrusion detection and response or for straightforward network mapping and performance.

There are also many similar network monitors deployed around the world. For example, the NCS plans to build GEWIS [18] (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 provides government and industry users with the warning of the threats that could degrade service, such as denial-of-service attacks against the DNS that control Internet traffic.

JPCERT/CC has started deploying the ISDAS [19] (Internet Scan Data Acquisition System). ISDAS has a wide distributed arrangement of sensors, and observes various scan activities; worm infections, probing vulnerable systems, etc. It provides the summarized scan trends observed on the web page. Moreover, the observed data are used as a basis of JPCERT/CC activities on publishing alerts and advisories, security awareness programs, etc.

PlanetLab [20] has also announced a plan to build their own monitor based on distributed wide aperture sensors. Building and deploying a threat monitor is not a cumbersome task for anyone with some unoccupied address space in hand.

With the analysis of the related work, 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, they are totally the national behavior, but not a federative behavior. The open architecture of GS-TMS makes itself more scalable and suitable for the dynamic network environment.

5.2 Data stream management system

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 [21] and Gigascope [22] 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 [23] DSMS as a network monitoring tool by modeling and running queries and making a simple performance analysis.

Recently, many researchers are concerned with solving network security problems in characteristics of the data stream. For example, Frederick et al. [24] focused on burst behavior in passive network monitoring job. Ram Keralapura [25] thought such monitoring applications are inherently continuous and distributed while must be designed to minimize the communication overhead introduced by them. Graham Cormode [26] concerned that Emerging large-scale monitoring applications require continuous tracking of complex data analysis queries over collections of physically distributed streams.

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.

6. CONCLUSION AND FUTURE WORK

In summary, we propose a comprehensive solution for Global Threat Monitor System. Our main contribution in this paper is to present a set of new methods for the integrated 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.

Although we have described our vision and goals of our system in this paper, a lot of effort still remains to bring our ideas to fruition. There are many important problems that need to be solved, we list them as follows:

1) We should do some experiments to test the performance of our system, include the communication costs between DMAs with GMP, the rate of packets loss, the rate of false detecting, and the CPU costs of agent, etc.

2) Next, we should analyze the results. The analysis should help us to improve the DMAs query processing and to reduce communication costs.

7. ACKNOWLEDGMENTS

This research is supported by National Basic Research Program of China under Grant No.2007AA01Z474, No.2006AA01Z451, National High-Tech Research and Development Plan of China under Grant No. 2007AA010301, and National Science Fund for Outstanding Youths under Grant No. 60625203.

8. REFERENCES

[1] US-CERT, "Technical Cyber Security Alerts"; http://www.us-cert.gov/cas/techalerts/.

[2] THE SNORT PROJECT, "Snort - the de facto standard for intrusion detection/prevention"; http://www.snort.org/.

[3] Cisco Systems, "Network Based Application Recognition "; http://www.cisco.com/en/US/products/ps6616/products_ios_protocol_group_home.html.

[4] C.A. Siegel, T.R. Sagalow, and P. Serritella, "Cyber-Risk Management: Technical and Insurance Controls for Enterprise-Level Security," Information Systems Security, vol. 11, 2002, pp. 33-49.

[5] C. Estan and G. Varghese, "Data streaming in computer networking," Workshop on Management and Processing of Data Streams, 2003.

[6] B. Babcock et al., "Models and issues in data stream systems," Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2002, pp. 1-16.

[7] S. Kandula et al., "Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds," 2nd Symposium on Networked Systems Design and Implementation (NSDI), 2005.

[8] H.A. Kim and B. Karp, "Autograph: Toward Automated, Distributed Worm Signature Detection," USENIX Security Symposium, vol. 286, 2004.

[9] M. Andersson, "Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering," 13th International Conference on the Entity-Relationship Approach, Manchester, United Kingdom, December 13-16, 1994: Proceedings, 1994.

[10] P.A. Bernstein, S. Melnik, and P. Mork, "Interactive schema translation with instance-level mappings," Proceedings of the 31st international conference on Very large data bases, 2005, pp. 1283-1286.

[11] A. Cali, D. Lembo, and R. Rosati, "Query rewriting and answering under constraints in data integration systems," Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003), 2003, pp. 16–21.

[12] R. McCann et al., "Mapping maintenance for data integration systems," Proceedings of the 31st international conference on Very large data bases, Trondheim, Norway: VLDB Endowment, 2005, pp. 1018-1029.

[13] J. Miao et al., "Detecting Broken Mappings for Deep Web Integration," Semantics, Knowledge and Grid, Third International Conference on, 2007, pp. 56-61.

[14] C.R. Simpson Jr, "NETI@ home," Software on-line: http://neti. gatech. edu, 2003.

[15] K. Shanmugasundaram et al., "ForNet: A Distributed Forensics Network," Computer Network Security: Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2003, St. Petersburg, Russia, September 21-23, 2003: Proceedings, 2003.

[16] DIMES, "The DIMES project"; http://www.netdimes.org/new/.

[17] V. Yegneswaran, P. Barford, and S. Jha, "Global Intrusion Detection in the DOMINO Overlay System," Proceedings of Network and Distributed System Security Symposium, 2004.

[18] Bob Brewin, "Feds planning early-warning system for Internet"; http://www.computerworld.com/securitytopics/security/hacking/story/0,10801,75248,00.html.

[19] JPCERT/CC, "Internet Scan Data Acquisition System (ISDAS)"; http://www.jpcert.or.jp/isdas/index-en.html.

[20] PlanetLab, "An open platform for developing, deploying, and accessing planetary-scale services"; http://www.planet-lab.org/.

[21] A. Arasu et al., "STREAM: The Stanford Data Stream Management System," a book on data stream management edited by Garofalakis, Gehrke, and Rastogi, 2004.

[22] C. Cranor et al., "Gigascope: a stream database for network applications," Proceedings of the 2003 ACM SIGMOD, 2003, pp. 647-651.

[23] S. Chandrasekaran et al., "TelegraphCQ: continuous dataflow processing," Proceedings of the 2003 ACM SIGMOD, San Diego, California: ACM, 2003, pp. 668-668.

[24] F. Reiss and J.M. Hellerstein, "Declarative Network Monitoring with an Underprovisioned Query Processor," Procs. of ICDE, April, 2006.

[25] R. Keralapura, G. Cormode, and J. Ramamirtham, "Communication-efficient distributed monitoring of thresholded counts," Proceedings of the 2006 ACM SIGMOD international conference on Management of data, 2006, pp. 289-300.

[26] G. Cormode and M. Garofalakis, "Sketching streams through the net: distributed approximate query tracking," Proceedings of the 31st international conference on Very large data bases, 2005, pp. 13-24.