Sharing Information for Event Analysis over the Wide Internet

Masahiro Nagao, Kazuhide Koide, Akihiro Satoh, Glenn Mansfield Keeni, and Norio Shiratori

Abstract: **Cross-domain event information sharing is a topic of great interest in the area of event based network management. In this work we use data sets which represent actual attacks in the operational Internet. We analyze the data sets to understand the dynamics of the attacks and then go onto show the effectiveness of sharing incident related information to contain these attacks. We describe universal data acquisition system for event based management (UniDAS), a novel system for secure and automated cross-domain event information sharing. The system uses a generic, structured data format based on a standardized incident object description and exchange format (IODEF). IODEF is an XML-based extensible data format for security incident information exchange. We propose a simple and effective security model for IODEF and apply it to the secure and automated generic event information sharing system UniDAS. We present the system we have developed and evaluate its effectiveness.**

Index Terms: **Backscatter, darknet, event based network management, event information sharing, incident object description and exchange format (IODEF), network management system, worm propagation.**

I. INTRODUCTION

The Internet is now an important infrastructure of modern society. The need for effective network management is growing in importance. In this work we address some issues related to cross-domain event information sharing [1] for effective network management. Information sharing is very important for domain managers managing their respective local domains. It enables them to understand the network behavior in the context of the global Internet which is a distributed autonomous system. Often effects of events in the Internet, e.g., the outbreak of a computer virus, distributed denial of service (DDoS) attacks, etc. are observed at the boundaries of the local networks. To understand and effectively react to these types of events, information sharing is important. We use the term *event information* to indicate data related to the event. This data is analyzed to understand the events.

We discuss some scenarios in which cross-domain event information sharing is needed by domain managers. One scenario relates to worm propagation. Computer worms spread quickly and widely by transmitting copies of themselves to vulnerable hosts. This activity is generally manifested as unusual network activity which may or may not be noticed by a network manager. A manager who notices the activity may diagnose it as worm activity and then proceed to block the activity; others may not notice or may simply overlook it due to ignorance about the nature of the activity. In such situations, sharing the information of worm detection and filtering might remedy the ignorance and thereby enable other managers to detect the worm in their own networks quickly and to initiate action against worm propagation. Cross-domain event information sharing is important for correlating apparently disparate events occurring over a wide area. Such correlation may reveal a pattern. In one scenario, say, several sites in disparate locations are attacked at roughly the same time. The individual domain managers may not be aware of the complete picture and may not notice any pattern. However, if several of the targeted domains are related to national infrastructure systems like public utilities, the emergent pattern will raise the possibility that the attack is an act of cyber terrorism. If the information about the time, target organization and network traffic features of an attack could be shared with other organizations, the purpose of the attack might become clear. In that sense event information sharing is vital for national defense against cyber terrorism.

One of the major issues in cross-domain event information sharing is *automation*. In most event information systems, an essentially manual mode of information scanning and event information sharing is employed. Incident reports are published on web pages, and an interested manager will browse, correlate and analyze the contents manually. By automation, we envisage that relevant data will be collected automatically in the wake of a pre-defined event, without any human intervention. This data comprises the event information. We also envisage that the event information should be in a format that is amenable to processing by event query and analysis applications without any human intervention. A major barrier to automation in event information sharing is the absence of a common machine parseable data format for describing events or incidents. There are several management information models, such as structure of management information version 2 (SMIv2) [2] and guidelines for the definition of managed objects (GDMO) [3], but there is no standard format that can be used for selecting data elements based on corresponding attributes from a large data store and exchanging the selected information between two parties. To overcome this barrier, we propose a generic event data format for event in-

Manuscript received October 20 , 2008; approved for publication by James Won-Ki Hong, Division III Editor, January 20, 2010.

This work was partially supported by the Ministry of Internal Affairs and Communications in Japan, SCOPE project (071502003), and by the Ministry of Education, Culture, Sports, Science and Technology in Japan, Grants-in-Aid for JSPS Fellows, 20002129 and for Scientific Research, 19200005.

M. Nagao and A. Satoh are with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan, email: nagao@shiratori.riec.tohoku.ac.jp, satoh@ka.riec.tohoku.ac.jp.

K. Koide is with the KDDI Corporation, Tokyo, Japan, email: ka-koide@kddi.com.

G. M. Keeni is with the Cyber Solutions Inc., Sendai, Japan, email: glenn@cysols. com.

N. Shiratori is with the Research Institute of Electrical Communication, Tohoku University, Sendai, Japan, email: norio@shiratori.riec.tohoku.ac.jp.

formation based on the standard incident object description and exchange format (IODEF) [4].

Automatic cross-domain event information sharing is difficult because managers have to take into account security and privacy issues. Event information that can be shared with users in the same domain may be out of bounds to users of another domain. Even in the same domain, sensitive information will be accessible only to users with appropriate access privileges. To avoid problems, most systems simplify matters by removing all potentially security-sensitive parts and sharing the rest of the event information with the public by publishing the information in a centralized repository. In other systems, the information is not filtered, but access to the information is granted to members of a small and restricted group, if at all. There is a trade-off between security and openness for event information sharing. We address the related issues and attempt to introduce the flexibility that allows more openness for event information sharing without raising the security risks.

In Section II, first, we explain the results of our investigation on the dynamics of attacks in the Internet. Next, we consider applications of information sharing and problems to be solved. In Section III, we explain our proposed scheme, and in Section IV, we explain its implementation. In Section V, we evaluate our proposed scheme and discuss its availability and effectiveness. In Section VI, we introduce some related works. Finally, we conclude in Section VII.

II. INFORMATION SHARING AND ITS APPLICATION

A. Dynamics of Attacks in the Internet

Network managers are primarily concerned about events in their own networks. The Internet is a very large distributed autonomous system. An event that occurs in one network may affect several other networks, so event information sharing, that is, event information access from other domains and to other domains, is essential for event management.

First, we investigated the dynamics of attacks in the Internet with two datasets. One is the witty worm dataset [5] and the other is the backscatter 2007 dataset [6]. Both are provided by cooperative association for Internet data analysis (CAIDA) [7] and contain packet traces monitored by the UCSD network telescope, which is one of the darknets [8].

A darknet is a portion of allocated IP address space in which there are no active servers, services or hosts. Packets observed in the darknet are unsolicited. These packets have probably arrived due to mis-configuration but more likely due to a mistake or intentional scanning by malware for vulnerable devices or due to attacks from spoofed addresses. Traffic in the darknet is useful for trend analysis of attacks, worms, viruses, and so on.

These datasets monitored in the darknet are mainly related to worm propagation and DDoS spreading. These activities have been studied in several papers [9]–[11]. Unlike these works, we focus on domain-wise monitoring and cross-domain information sharing.

A.1 Witty Worm Propagation

Witty worm is a computer worm discovered in March 2004. The actual propagation of the witty worm was investigated in [12] and [13] and its propagation model has been analyzed in [10], [14]–[16]. While these works mainly focused on the activity of each infected host, we delve into the actual domain-wise propagation of the witty worm.

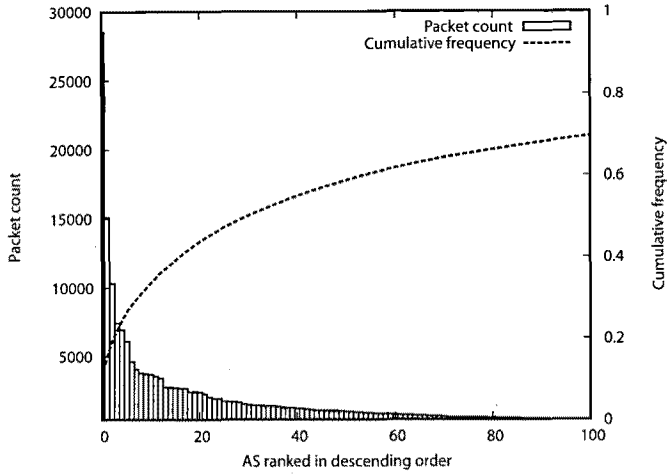

Fig. 1. Count of witty worm packets per AS and its cumulative frequency.

The packet traces of the witty worm dataset were obtained between Mar 20 04:01:40 UTC 2004 and Mar 25 07:01:40 UTC 2004. Witty worm packets have source port user datagram protocol (UDP)/4000. The packets in the trace contain complete IP and UDP headers and partial payload, up to a total of 82 bytes. This dataset contains packets from infected nodes from all over the Internet arriving at the UCSD network telescope.

We investigated the packet count per source autonomous system (AS) number (the AS number corresponding to the source IP address in the packet). Fig. 1 shows the ranking of the ASs by packet count. The X-axis shows the AS-rank. Only the top 100 ASs out of a total 2,203 are shown. The left Y-axis shows the packet count for the corresponding AS and the right Y-axis shows the cumulative packet count. This figure shows that the variance of packet counts per AS was large. The sum of packets from top five ASs was 25% of total packets from all 2,203 ASs.

Next, we examined the packet count per hour for each AS. Fig. 2 shows the count of witty worm packets per hour of each top-5 AS. The X-axis gives time t and the Y-axis gives the count. In this figure, we found that the hourly packet count for AS 8517 dropped sharply. The count between Mar 20 14:01:40 and 15:01:40 was 1462, and the count between Mar 20 15:01:40 and 16:01:40 was 0. After that, there were no packets from AS 8517. We denote t_0 as this starting time of the duration with no packets from AS 8517 (t_0 = 03/20 15:01:40). Probably AS 8517 detected the witty worm event at an early stage and quickly deployed counter measures just before t_0. Other domains did not show this sharp drop.

Finally, we examined the propagation activity of witty worm. Fig. 3 shows the cumulative count of countries which sent witty worm packets to the UCSD network telescope. The X-axis gives time t and the Y-axis gives the count. It is clear that within the first hour, most countries had been infected.

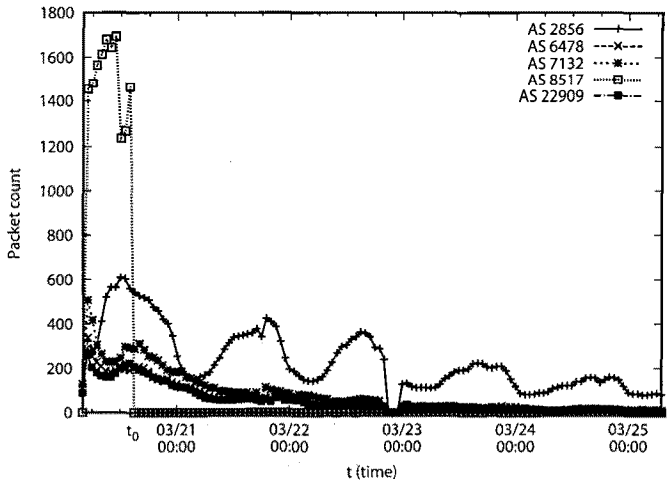

Fig. 2. Count of witty worm packets per hour of each top-5 AS.

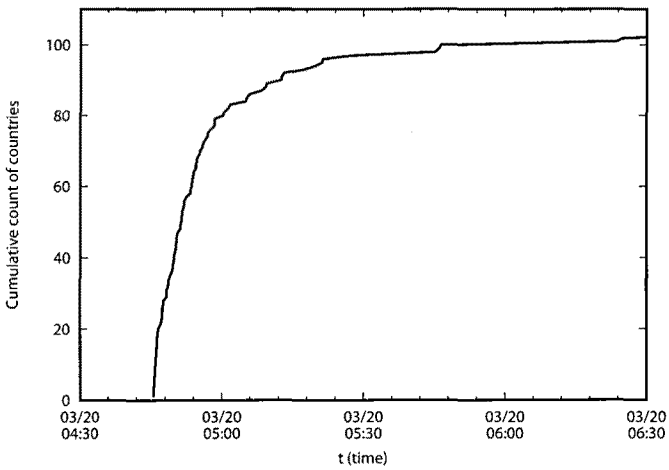

Fig. 3. Cumulative count of countries which have sent witty worm packets.

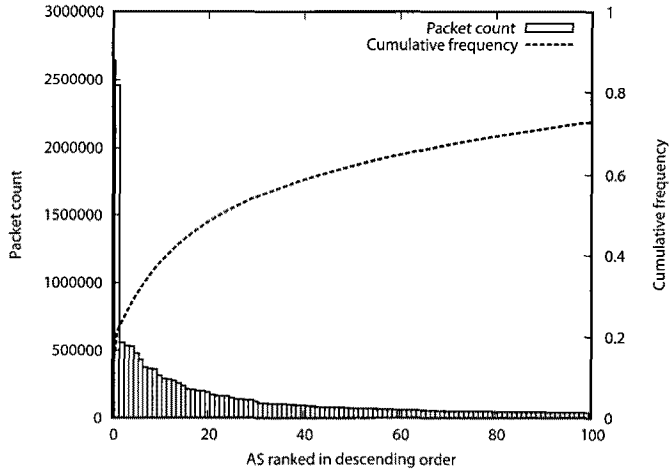

Fig. 4. Count of backscatter packets per AS and its cumulative frequency.

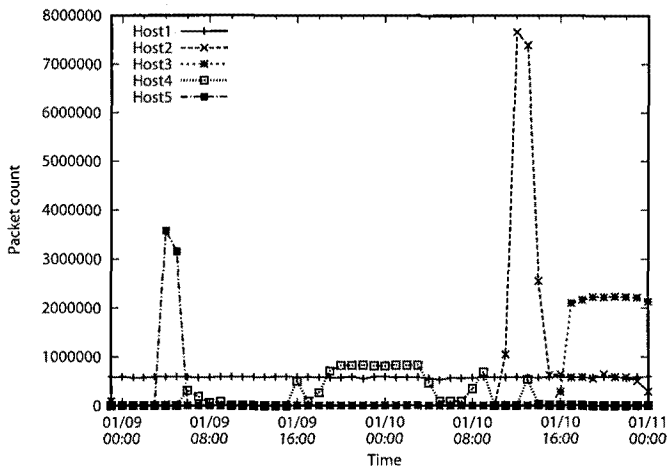

Fig. 5. Count of backscatter packets per hour of each top-5 host.

A.2 Backscatter of DoS Attacks

When a denial of service (DoS) attack victim receives an attack packet, it responds to the source address in the packet. If the source address is spoofed, the response is directed to the spoofed address. These responses constitute what is known as *backscatter* [17]. A part of backscatter will be destined to the darknet. It is possible to collect data of these backscatters by monitoring packets destined to the darknet at border gateways. The backscatter 2007 dataset comprises packets that arrived at the UCSD network telescope and packets that were collected as part of *day in the life of the internet* (DITL) project [18]. We used packet traces for the time span between Jan 8 23:00:00 UTC 2007 and Jan 11 00:59:59 UTC 2007. Fig. 4 shows the ranking of the ASs by packet count. The X-axis shows the AS-rank. Only the top 100 ASs out of a total 10,350 are shown. The left Y-axis shows the packet count for the corresponding AS and the right Y-axis shows the cumulative packet count. It is clear that a small number of ASs were sending a large number of backscatter packets. These ASs have very likely been exposed to some DoS attacks.

Fig. 5 shows a host-wise (source IP-wise) count of backscatter packets per hour. It shows the count for the top 5 hosts only. The count for host1 maintains a steady value. A large number of backscatter packets indicate the likelihood of one or more DoS attacks. The steady backscatter seen from host1 could indicate that the managers of the domain containing host1 are probably not aware of the attack.

B. Application Scenario of Event Information Sharing

From our analysis, it appears that a large number of managers do not detect worm propagation or DoS attacks, while a relatively small number of managers appear to have detected and reacted to these events. In the following, we demonstrate a scenario where event information sharing helps. In this scenario, we consider each AS as an independent management domain and these ASs can share event information with each other.

B.1 Restraining Worm Propagation

In subsection II-A.1, we infer that AS 8517 detected the witty work event at an early stage. Let us consider the hypothetical case where AS 8517 publishes this event information along with the features of the witty worm traffic. Other ASs would have a

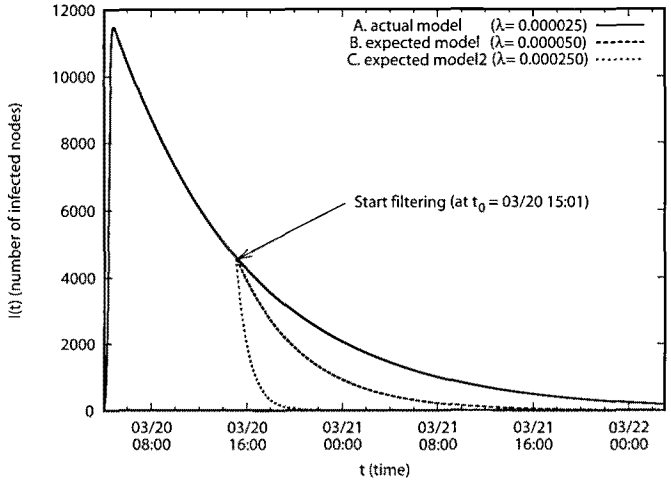

Fig. 6. Propagation model of witty worm.

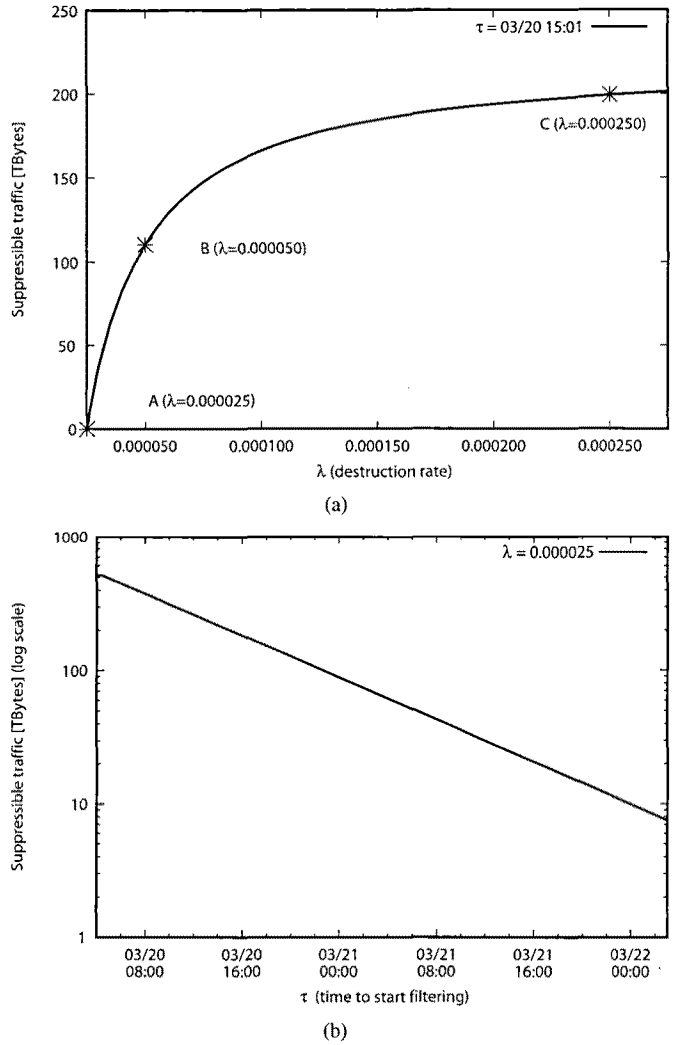

Fig. 7. Suppressible amount of witty worm traffic: (a) Variable destruction rate λ and (b) variable time to start filtering τ.

better chance to be aware of the witty worm at a relatively early stage and at least some of those would quickly deploy counter measures. Our argument is that if the event information detected by AS 8517 was shared among other management domains the effect of the witty worm would have been more effectively contained.

To estimate the impact of event information sharing, we first introduce the propagation model of witty worm. In [10], the propagation model of witty worm traffic was investigated using the same dataset of CAIDA and another real witty worm traffic contemporary dataset captured in *the Internet motion sensor* [19]. In the model, the variation of the number of infected nodes with time is numerically analyzed. Fig. 6 shows the propagation model of witty worm given in [10]. The X-axis represents time t and Y-axis gives the number of infected nodes $I(t)$. We focus on one of the parameters of the model, the *destruction rate* λ. It is the average rate at which hosts infected by witty worms crash and stop sending packets. In this model, λ is given a constant value of 0.000025.

The filtering of witty worm packets at a domain boundary has the same effect as the crashing of *all* infected nodes in the domain as the worm packets and thus the propagation activity from the domain ceases. Therefore, in the model we can use the destruction rate λ as a parameter to study the effect of active filtering of witty worm packets. We changed λ to a larger value at t_0 in the propagation model under the hypothesis that AS 8517 published the filtering information for witty worm and some ASs made use of the information to filter witty worm packets from their outgoing traffic at t_0. Curve B and curve C in Fig. 6 are plotted with $\lambda = 0.000050$ and $\lambda = 0.000250$, respectively. In Fig. 6, we can see that for a larger λ a larger number of infected hosts were isolated. The amount of witty worm traffic is roughly proportional to the count of infected hosts. Thus, the area enclosed by curve A and curve B gives the amount of witty worm traffic that would be suppressed if the destruction rate increased two-fold, while the area enclosed by curve A and curve C gives the amount of witty worm traffic that would be suppressed if the destruction rate increased ten-fold. We can also see that the closer t_0 is to the origin, the larger is the amount of the suppressed traffic S. In other words, the earlier the information is published and made use of, the greater is the suppressed traffic or effectiveness of the information.

Fig. 7 estimates the amount of witty worm traffic that could be suppressed. X-axis of Figs. 7(a) and 7(b) give destruction rate λ and filter start time τ, respectively. The Y-axis gives the estimate of the suppressed witty worm traffic S. It is calculated by $S = I(t)\eta l$, packet sending rate $\eta = 1200$ [packet per sec] and average packet size $l = 1000$ [bytes]. In Fig. 7(a), τ is constant t_0 and λ is variable. In Fig. 7(b), λ is constant 0.000025 and τ is variable. Fig. 7(a) shows that the amount of suppressed traffic would increase non-linearly with λ, and Fig. 7(b) shows that the amount of suppressed traffic would decrease exponentially (linearly in log scale) with τ. The results show that both *quick* and *wide* event information sharing are effective and *quick* event information sharing is more effective than *wide* event information sharing.

B.2 Detecting DoS Attack by Backscatter Packets

Another scenario is DoS attack detection. Monitors in the darknet can detect backscatter packets easily but it is very dif-

ficult for managers to tell a backscatter packet from a non-backscatter packet, so data monitored in the darknet is worth sharing with managers. It enables managers to detect attacks easily. Monitors in the darknet measure the count of backscatter packets per AS and the results are made accessible in a controlled manner. Managers regularly check the statistics corresponding to their respective ASs. From the statistics, the manager can figure out whether a DoS attack is in progress in his/her domain.

C. Problems and Expectations in Cross-Domain Event Information Sharing

As shown in Fig. 3, the worm did not spread to all countries instantaneously. It took about an hour for the worm to spread to most of the countries. Clearly, some effective national level action to preempt the attack, could have been put in place within this time window if there was a quick and wide event information sharing mechanism. There are two major issues to be addressed for realizing automatic event information sharing. One is data format, and the other is access control.

In the hypothetical scenario (subsection II-B.1), the manager in AS 8517 detects the worm event but does not have an appropriate data format for sharing it. The packet trace itself, in this case monitored by the UCSD network telescope, cannot be used as it is, as it contains sensitive data, so the information for these events cannot be used automatically in real-time. One alternative would be to sanitize the data manually, depending on the querying AS and other factors. But that would be a time consuming tedious process that is impractical.

In the hypothetical scenario (subsection II-B.2), though the data monitored in darknets are useful for managers, publishing all the details would in effect reveal the address [20] of the darknet monitors. These addresses are a closely guarded secret as knowledge about these could be used to subvert the darknet monitoring system or poison the information collected, thereby rendering the darknet monitor data useless for analyzing the Internet. Thus arises the requirement that published darknet data must not contain sensitive data and the level of detail offered to a person will depend on the level of trust in that person. In other words, a proper access control mechanism is needed.

To summarize, the following two issues need to be addressed:

C.1 Data Format

Existing systems employ independent data presentation and transport. This is a problem from the extensibility, reusability, and scalability point of view.

If a new system needs to be introduced, the existing data will need to be converted, or the new system will need to be limited to the format of the legacy system. A standardized, extensible data format is necessary for mechanical processing and information sharing. Examples of data to be converted are packet traces for analysis of attacks and application level data like spam mail.

C.2 Security Model

Information sharing gives rise to security concerns. As much information as possible should be made available to the right person for effective information utilization. However, most existing systems drastically restrict access to prevent sensitive data from being revealed to the wrong person. Instead of this all or nothing policy, security policy should have the flexibility to deal with users with different levels of privileges. We have to design an appropriate *security model* to support such security policy.

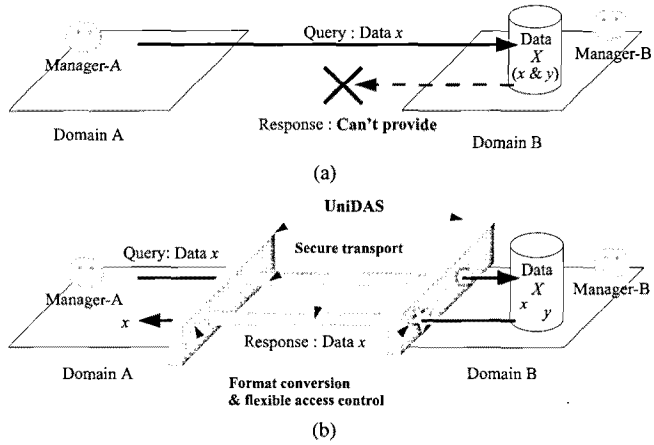

Fig. 8. A conceptual model of UniDAS: (a) Existing architecture and (b) UniDAS architecture.

III. UNIDAS: UNIVERSAL DATA ACQUISITION SYSTEM FOR EVENT BASED MANAGEMENT

A. Conceptual Scheme

In this section, we describe universal data acquisition system for event based management (UniDAS) that provides a generalized framework for an event information sharing system that supports automation of event information sharing and its utilization. This system is a generic application. We will focus on the realization of the applications described in Section II. Fig. 8 shows a conceptual model of UniDAS. UniDAS supports flexible access control to event information according to a security model described in subsection III-C.

Let us consider the example of a spam mail incident. Say manager-A of domain A sends a query to manager-B of domain B asking for the body of a spam mail. Domain B manager records spam mail in a message per file format. The bodies of the spam mail are not private or security sensitive. But the message also contains the receiver's address, which may be considered private. The message may also contain the address of the relay servers which served the message. These addresses may also be considered to be security sensitive. As such manager-B will probably reject the query for the message body. Manager-B could offer the files themselves to manager-C of domain C with whom the necessary security/privacy related agreements are drawn up. For example, DNS-based blackhole list (DNSBL) systems [21], [22] provide black-listed spam senders that do not contain any sensitive contents without access control. Apart from anti-spam systems, there are many systems which are intended to provide non-sensitive contents for other domains without access control such as ISDAS [23], Internet storm center (ISC) [24] and so on.

UniDAS is more flexible as it has a more fine-grain level of access control over the event information. Data X is structured

and the components have corresponding access controls defined. If access control requirements are met, the query is accepted and only data x from data X is offered.

The UniDAS model is independent of the database entity or data format of the back-end system. Data from the backend must be translated to the common data format in which it will be sent over a secure transport channel to the target system. The key points of UniDAS are its data format, security model and transaction model.

B. Data Format

A standard format is required that satisfies the requirements of various domains, is reasonably easy to use and has extensibility. UniDAS employs a standardized data format. IODEF is a standardized presentation protocol for exchanging security incident and related data. Its data structure is generic and versatile for security incidents. We adopt IODEF for our system. The main reason for our choice is that IODEF is already standardized for exchanging event information as well as representing various data. Additionally, it is transport-independent and has extensibility built in. The contribution of this work lies in the proposed additional data classes for network and application information. IODEF supports various data classes like *EventData class* and primitive types like *STRING* natively. In addition, all required extensions for data presentation can be supported by *AdditionalData class* and *STRING* data. These classes support description of any type of event information. It is an extensible markup language (XML) document, therefore, it can be processed easily with XML libraries which are widely available and on many platforms.

In UniDAS - UniDAS messages (u-messages) are exchanged. A u-message is an IODEF formatted message. It may be a query or a response. Figs. 9 and 10 show samples of IODEF description for u-messages in our system. Each u-message contains several pieces of *AdditionalData*. EventDisplay-QUERY contains a query for event information. EventDisplay-SenderInfo contains information about the sender of this IODEF document. EventDisplay-QueryResult contains "true" if there is the queried event and "false" if there isn't. EventDisplay-OFFLINE contains time series data to show the traffic graph related to the event. EventDisplay-TopN contains port-wise traffic of a host or network related to the event. These descriptions of *ApplicationData* class is our extension within the standard IODEF schema. Event information is provided in the above standard IODEF format over a secure transport.

C. Security Model

Within a local domain, e.g., branch offices connected via VPN, detailed data can be offered to concerned persons who have the appropriate access privileges. Otherwise several factors will need to be considered. In this paper, we discuss a specific genre of events which are observed at several network sites and the nature of the event is not obvious, at early stages, from the happenings at a single site. The event related information will include the following.

1) Start time and end time (duration)
2) Observation point (geographical and network topological)
3) Observed network activity information

```
<?xml version="1.0" encoding="UTF-8"?>
<iodef:IODEF-Document xmlns:iodef="draft-ietf-inch-iodef
-043.xsd" version="0.40" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance" xsi:schemaLocation="draft-ietf-
inch-iodef-043.xsd draft-ietf-inch-iodef-044.xsd">
<iodef:Incident>
<iodef:EventData>
<iodef:AdditionalData iodef:dtype="string">
<EventDisplay-QUERY>
<iodef:ID>130.34.38.130.9-query-1200481357588</iodef:ID>
<iodef:StartTime>2008/0/16</iodef:StartTime>
<iodef:EndTime>2008/0/16</iodef:EndTime>
<iodef:Source>Any</iodef:Source>
<iodef:Type>icmpInMsgs.0</iodef:Type>
<iodef:EventTime>1200481334254</iodef:EventTime>
</EventDisplay-QUERY>
</iodef:AdditionalData>
<iodef:Contact>
<iodef:Name>Tohoku University</iodef:Name>
<iodef:Email>admin@example.com</iodef:Email>
<iodef:AdditionalData iodef:dtype="string">
<EventInformation-PGP-Signature>
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (MingW32)

iEYEABECAAYFAkhfBmsACgkQch2U714+ul6AYQCfd5223YcEWW2/P
1tEcdzB2C6U/BAAnAyt3Y9PpBW8T7GWIXeNlBonpyvC
=PfhQ
-----END PGP SIGNATURE-----
</EventInformation-PGP-Signature>
</iodef:AdditionalData>
</iodef:Contact>
</iodef:EventData>
</iodef:Incident>
</iodef:IODEF-Document>
```

Fig. 9. A sample of IODEF description of an event query.

Table 1. An example of security requirement in typical environment.

Type	DATA -CONTENT	CONSUMER Local	Companion	Law enforcement authority	Public
Packet trace	In/out src port	Y	Y	Y	N
	In/out dst port	Y	Y	Y	N
	In/out src addr	Y	Y	Y	N
	In/out dst addr	Y	Y	Y	N
	In/out protocol	Y	Y	Y	N
	In/out other header	Y	Y	Y	N
	In/out payload	Y	N	Y	N
	Timestamp	Y	Y	Y	N
Attacked organization	Organization type	Y	Y	Y	Y
	Organization name	Y	Y	Y	N
	Geographical region	Y	Y	Y	N
Spam mail	Header: To	Y	Y	Y	N
	Header: From	Y	Y	Y	Y
	Header: Subject	Y	Y	Y	N
	Header: Received	Y	Y	Y	N
	Header: Date	Y	Y	Y	Y
	Message body	Y	Y	Y	N
	Hash value of the body	Y	Y	Y	Y

Instances of such events are DDoS attacks, worm propagation, spam mail and so on. The security model described in this section will cover such events. For example, the manager of domain A may provide some network information to a manager of domain B if the two domains have agreed to cooperate in network management. Often these conditions are implicit. One contribution of our system is the development of a model that

```
<?xml version="1.0" encoding="UTF-8"?>
<iodef:IODEF-Document xmlns:iodef="draft-ietf-inch-iodef
-043.xsd" version="0.40" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance" xsi:schemaLocation="draft-ietf-
inch-iodef-043.xsd draft-ietf-inch-iodef-044.xsd">
<iodef:Incident>
<iodef:EventData>
<iodef:AdditionalData iodef:dtype="string">
<EventDisplay-SenderInfo>
<iodef:Sender>130.34.38.184</iodef:sender>
<iodef:OrganizationType>Academic Institution
</iodef:OrganizationType>
<iodef:OrganizationName>Tohoku University
</iodef:OrganizationName>
<iodef:Country>JP</iodef:Country>
<iodef:City>Sendai</iodef:City>
</EventDisplay-SenderInfo>
</iodef:AdditionalData>
<iodef:AdditionalData iodef:dtype="string">
<EventDisplay-QueryResult>
<iodef:status>true</iodef:status>
</EventDisplay-QueryResult>
</iodef:AddtionalData>
<iodef:AdditionalData iodef:dtype="string">
<EventDisplay-OFFLINE>
<iodef:Mo>ifInOctets.3</iodef:Mo>
<iodef:Host>cpMonitor</iodef:Host>
<iodef:Domain>LocalNet</iodef:Domain>
<iodef:Time>1169602477656</iodef:Time>
<iodef:Param time="1169600070" value="3860985519"/>
        :
<iodef:Param time="1169602476" value="3862119289"/>
</EventDisplay-OFFLINE>
</iodef:AdditionalData>
<iodef:AdditionalData iodef:dtype="string">
<EventDisplay-TopN>
<iodef:FileName>192.168.108.1</iodef:FileName>
<iodef:DATA>
192.168.108.1,993,47020,0,47020,0,0,1168300805,1168302585
192.168.108.1,143,1726,0,1726,0,0,1168300810,1168302570
192.168.108.1,25,1465,0,1465,0,0,1168300800,1168302590
192.168.108.1,53,140,140,0,0,0,1168300800,1168302590
192.168.108.1,22,12,0,12,0,0,1168300800,1168302590
192.168.108.1,80,9,0,9,0,0,1168300810,1168302535
192.168.108.1,111111,5,5,0,0,0,1168300800,1168302590
192.168.108.1,123,1,1,0,0,0,1168300820,1168302565
192.168.108.1,135,1,0,1,0,0,1168302140,1168302460
</iodef:DATA>
</EventDisplay-TopN>
</iodef:AdditionalData>
</iodef:EventData>
</iodef:Incident>
</iodef:IODEF-Document>
```

Fig. 10. A sample of IODEF description including various types of data.

Table 2. An example of security requirement in darknet.

Type	DATA -CONTENT \ CONSUMER	Local	Companion	Law enforcement authority	Public
Packet trace	Incoming src port	Y	Y	Y	Y
	Incoming dst port	Y	Y	Y	Y
	Incoming src addr	Y	Y	Y	N
	Incoming dst addr	Y	N	Y	N
	Incoming protocal	Y	Y	Y	Y
	Incoming other header	Y	Y	Y	N
	Incoming payload	Y	Y	Y	N
	Timestamp	Y	Y	Y	N

describes such conditions explicitly as a *security requirement*. We propose to represent the security requirement as a 3-tuple:

1) **DATA-CONTENT:** This describes the nature of the data, e.g., the Src address of the incoming packet is contained in the data.
2) **CONSUMER:** This describes who will use the data, e.g., some data may be opened for public consumption.
3) **PROBE-POINT:** This describes the point at which the data was observed, e.g., UCSD network telescope.

Managers have to define their security requirement in the context of the PROBE-POINT. Security and privacy requirements will differ with the PROBE-POINT, e.g., home networks, office networks and darknets. For each PROBE-POINT, the security requirement is represented by a matrix of DATA-CONTENT and CONSUMER. Actually, how the DATA-CONTENT should be defined is a major issue. We mainly focus on packet traces because they have been used as data sources of many analysis systems like Snort [25], which is a de facto standard of intrusion detection system (IDS)/ intrusion prevention system (IPS). Rules of Snort are informative about how to define DATA-CONTENT in packet traces. In practice, network managers should define their security requirement to support general rules of Snort.

Table 1 shows an example of security requirements in a typical environment. The left column shows the DATA-CONTENT of the information items. This example applies to packet traces and spam mail. Organization information is useful for applications which handle information that is geographically dispersed and diverse; for example the office of national security as described in section I. In the table, **Y** indicates that access privileges to the information item can be granted and **N** means access will be denied. A packet trace is very useful for network analysis, but it is one of the data types that clearly cannot be offered publicly, because it contains several pieces of potentially sensitive data. Generally, all DATA-CONTENT of packet traces shouldn't be open to public except under special circumstances. If there is some special relationship like research collaboration, the data may be offered to the collaborators for research purposes in a sanitized form. For example, only the header of each packet may be offered for traffic flow analysis, while payload will not be offered. Additionally, depending on the country, the complete packet trace will have to be made available to law enforcement authority if formally requested. The name and address details of an attacked organization are also not open to public because the news of the attack may have a negative impact on the organization.

Table 2 shows an example that is relevant to a darknet. In a darknet, there are no outgoing packets. Not only the destination address of incoming packets, but also precise timestamps and full packet header including IP addresses should not be openly accessible. These may be used to infer the address of the darknet monitors. Perhaps only src/dst port and protocol number can be offered to the public. Of course, destination address of incoming packets may be offered only to a closed group of researchers/investigators.

The security requirements will depend on the application or data type. For spam mail sharing, managers have to consider mail files rather than packet traces. The lower part of Table 1 shows an example of security requirement related to spam mail. DATA-CONTENT is represented by the message body and each header value like to-header, from-header, subject-header, received-header, date-header, and so on. The to-header and the message body may contain some private information like name

or address, so they are not offered to public. A hash value of the message body may be offered to the public while the message body itself may be offered only to a restricted group of co-researchers.

There are two aspects of realizing a security model. One is the authentication mechanism. Depending on the identity of the requestor, which are authenticated, the access rights of the requestor vis-a-vis the data will be derived. IODEF supports a *contact class* in which the sender of the message can be specified. Contact class has provisions of certificates which can be appended as AdditionalData that can be used for authentication at the application layer. A mapping between the CONSUMER specified in the security model and the contact class is required to effect the access control specified in the security model. For a global system, individual identity access control does not scale. In our system, the individual is authenticated, his/her role is determined, and his/her access privileges are computed. Our security model is a new system design using existing technologies. The other aspect is data filtering. Effective filtering from chunks of data is needed. For each CONSUMER, a filtering script will filter out the inappropriate components. In this respect, XML documents are much easier to handle than packet traces because an XML document is a well structured text file. XSLT [26] is one of the filtering methods. Of course, a perl script or a compiled program can be used for the same purpose.

In this section, we focus on the design of the access control scheme. Practical security will be ensured with the following technologies for protecting event information. In the transport layer, public key infrastructure (PKI) functions like secure sockets layer (SSL) can be used to prevent unauthorized access and for transporting event information securely. In the application layer, XML security technologies are applicable for u-message. XML signature [27] can be used for preventing unauthorized access and XML encryption [28] can be used for storing transported event information securely.

D. Transaction Model

The first reason is that the proposed security model will not work well in a centralized architecture. All domains will be required to provide their data to the centralized server; this is unacceptable from the security point of view. The second problem is that of scalability of the data storage. Event information can be voluminous. All information cannot be stored in a single server.

In our system, data is not centralized. Our system realizes domain-to-domain information sharing. There are two models of transactions: the pull model and the push model. In the pull model, a manager sends a query to another domain and gets a response from the domain according to the security requirements between these domains. When a manager wants to analyze an event in detail, he/she will send a query to other domains and get more detailed data for the event analysis. In the push model, a manager can send advertisement, notification or warning information to other domains. While pushing the information the same security mechanisms will be used to determine what information will go to whom. Our system can support flexible data acquisition and publication with a well defined security model.

In this transaction model, all messages are transported over a standardized secure transport. There are two requirements of the secure transport: Encryption of data and authentication of sender/receiver. These requirements are important in order to realize our security model.

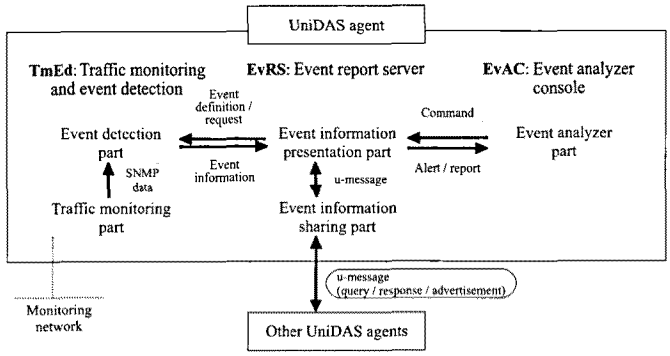

Fig. 11. An overview of modules in the UniDAS agent.

For this transaction model, the scalable name resolution scheme for the event name, domain name and data name is required. Here, we can simply utilize existing schemes. In the area of web services, the combination of web digital subscriber line (WSDL) [29], universal description discovery and integration (UDDI) [30] and SOAP [31] have been discussed and tested for this purpose. Traditional directory services like domain name system (DNS) or lightweight directory access protocol (LDAP) [32] can also be used. Our system assumes that all the domains share the name list. They know where they can send a query and how to specify the required data. An event and its data are specified by protocol, port number, IP address, timestamp, and so on. An example of EventDisplay-QUERY is shown in Fig. 9.

IV. DESIGN AND IMPLEMENTATION

A. Overview of the UniDAS Agent

In this section, we describe the design and implementation of the UniDAS agent. The UniDAS agent works in a local domain and realizes event information querying and response. The implementation described in this section is designed to deal with traffic data for traffic monitoring and analysis. Traffic data is an important element in network management. Fig. 11 shows an overview of the UniDAS agent. It consists of three modules. The event report server module (EvRS) is related to the security model. The *event information presentation part* is in charge of IODEF translation, and the *event information sharing part* is in charge of filtering translated IODEF documents based on security requirements and authentication.

B. TmEd: Traffic Monitoring and Event Detection Module

Traffic monitoring is carried out by a simple network management protocol (SNMP)-based manager-agent method. We employed the CpMonitor [33] technology. The CpMonitor captures all packets on the monitored links and provides various statis-

tics based on fields in the packet headers. It works as a SNMP agent and generates counter-type managed objects (MOs) corresponding to these statistics. An abstract representation of the statistics is defined in the CpMonitor management information base (CpMonitor-MIB) module. It enables monitoring of IP address-wise traffic, port-wise traffic, count of unique IP addresses etc. The traffic monitoring part in TmEd module works as a SNMP manager. It collects and stores time-series traffic data from the CpMonitor-MIB.

The event detection part in the TmEd module finds an objective event from these time-series data using pre-defined event detection rules. Detecting events is not our focus in this work. The implemented system supports only simple arithmetic operations for MO values. For example, "*cpmSnapStatsV4IcmpMsgs.0* $\geq$ *1000*" is an event detection rule. *cpmSnapStatsV4IcmpMsgs.0* is the name of an MO in CpMonitor-MIB, which counts the number of ICMPv4 packets seen on the monitored link in a pre-defined interval. For more practical event detection, implementation of any other event detection algorithm is possible and left for future work.

Generally, a traffic monitoring system needs very large storage. To handle a large amount of monitored data efficiently, we adopt two policies of storing. One is an event-based storing. Monitored data will be stored for a long time only when an event related to the data is detected in a local domain or a query related to the data is delivered from other domains. Otherwise the monitored data will be discarded after a certain period, e.g., a few days. The other policy is distributed data storing. Data should not be centralized and be requested only when needed.

C. EvRS: Event Report Server Module

As we showed in Fig. 10, any type of event information can be written in an IODEF XML file as a u-message. In the pull model, a u-message is built automatically based on the query as shown in Fig. 9. Each element in the EventDisplay-QUERY element specifies the required event information. The EvRS module that receives it can parse these elements and generate the reply u-message. This translation of event information is processed by the event information presentation part in the EvRS module. While offering the detected event information, it encodes the time-series traffic data of the event into XML format.

The event information sharing part in the EvRS module deals with u-message query/response/advertisement between UniDAS agents. There will be some choices in transportation mechanisms of the XML document. We employed e-mail as a secure transport in this system. E-mail can satisfy the secure transport requirements described in Fig. 8, e.g., message encryption and sender authentication by pretty good privacy (PGP) or secure/multipurpose Internet mail extensions (S/MIME). It can satisfy the transaction model described in subsection III-D.

D. EvAC: Event Analyzer Console Module

This is the interface for the human manager to analyze subjective events with lots of event information reported both from their own domain and other domains.

The manager will generate a query for an event through this module. Generally, the manager will receive a report of an event from a local TmEd module. The manager will want to get more

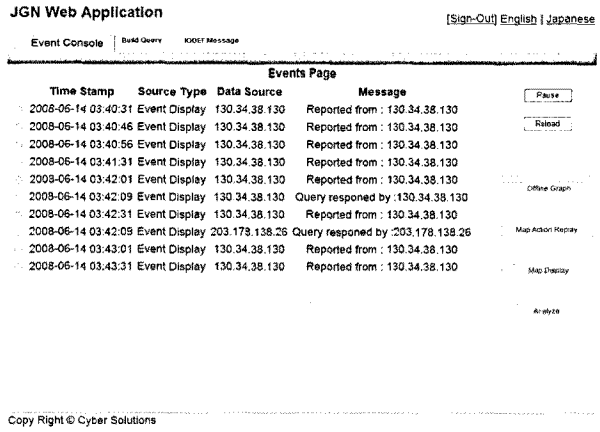

Fig. 12. Screenshot of web-based event analyzer console.

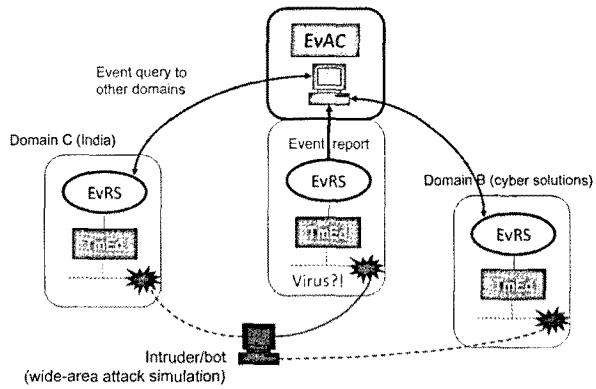

Fig. 13. Experimental environment.

information from other domains, and he/she will send an appropriate query and receive reports on whether the event was observed in other domain and if it was observed, it will probably receive additional event information.

We implemented a web-based graphical user interface (GUI). An EvAC module should have many functions of analysis and automation to help the human manager through the discovery of new events. Our implementation can show offline traffic graphs and border gateway protocol (BGP) network maps. Fig. 12 shows the screenshots of a web-based event analyzer console. It shows a list of events.

E. System Deployment

We deployed our system in the Internet and demonstrated a scenario of event information sharing and analysis. Fig. 13 shows the experimental environment. First, TmEd in domain A detected an artificial event. We operated an EvAC in domain A and sent queries of the event to the EvRSs of domain B and domain C. After a few seconds, these EvRSs returned the response to the EvRS of domain A.

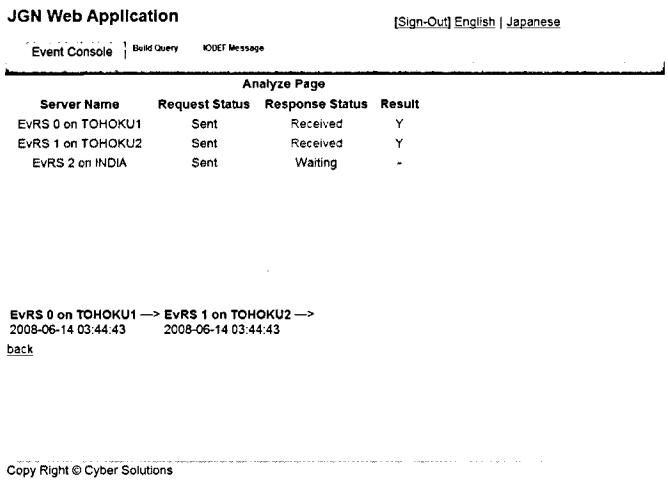

Fig. 14. The result of query.

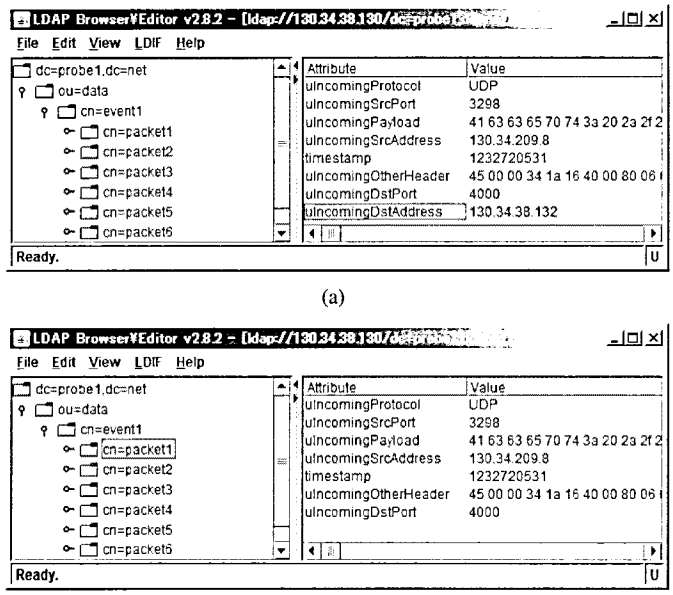

Fig. 15. The result of accesses: (a) A **Local** user and (b) a **Companion** user.

Fig. 14 shows the result of the query. Each row represents the ID of the EvRS. The result shows whether each EvRS detected the queried type of events and its detected time, if they were detected. Each query took 110–160 seconds to respond. In this demonstration, event information represented in IODEF was simple and no complex additional data was used. It took only the time of the transportation of the query and response. Simple mail transfer protocol (SMTP) was used for sending queries and responses, and post office protocol version 3 (POP3) was used for receiving queries and responses. The interval time of polling to POP3 server was 10 seconds. This processing time is practical enough because worm propagation and backscatter in the Internet can continue for several hours or days. The advantage of this implementation is that it uses an off the shelf transport system. Optimization and/or efficiency of the transport system for u-messages was left out of the scope of this work.

In our security model, security requirements and access control schema are important aspects. However, we will leave out the implementation details as u-messages are easy to implement in any XML compatible system. We implemented the access control based on the security requirements described in Table 1 using LDAP. First, a user is authenticated in the login page of the EvAC. When the authenticated user accesses UniDAS, the user's role, such as local, companion, and public, is computed from the user's credentials. Subsequently, role-based access control can be realized with LDAP easily.

V. EVALUATION

A. Methodology

In this section, we discuss the UniDAS framework (not the implemented UniDAS agent which is described above) from two points of view: Availability and sufficiency of our security model. Though it is possible to deploy UniDAS in several domains, for experimental purposes, it is impractical to wait for a global attack or cyber terrorism to happen. So, we use some simple innocuous probes as attacks to explain how the problems of applications mentioned in Section II can be solved, and show that the UniDAS meets the requirements of security and effectiveness.

B. Realization of Security Model

First, we confirmed that our security model can be realized with LDAP based access control implementation described in the previous section. Figs. 15(a) and 15(b) show the result of access from a **Local** user and a **Companion** user, respectively. In this experiment, PROBE-POINT which was identified by DN (distinguished name) of "dc=probe1,dc=net" is a darknet. We set up the security requirement which was defined with *organization type*, *organization name* and *geographical region* in Table 1 and all DATA-CONTENTs in Table 2. In Fig. 15(a), a user with credentials of "cn=nagao,ou=operator,o=tohoku-univ,c=jp" and **Local** role accessed *probe1*. He could access all the attributes. In Fig. 15(b), another user with credentials of "cn=glenn,ou=operator,o=cysol,c=jp" and **Companion** role accessed *probe1*. He could not access *incoming dst addr*. As shown in Table 2, it is the only difference between **Local** and **Companion** roles. In another experiment, an anonymous user with **Public** role accessed *probe1* and could not access many DATA-CONTENTs in accordance with the security requirement. In this manner, the access control can be realized by role-based access control offered by LDAP.

C. Contributions of Application Scenarios

C.1 Witty Worm Scenario

In this scenario, a lack of common data format is the main problem. To solve the problem, we proposed an IODEF based data format. In our introduced application, only incoming src port and incoming protocol of each packet is required for sharing and explaining witty worm activity. These fields are not sensitive and can be open for public (Table 2). Packet traces are not appropriate for automatic event information sharing, because

Table 3. Another example of security requirement about traffic statistics in darknet.

Type	DATA -CONTENT	CONSUMER Local	Companion	Law enforcement authority	Public
Traffic statistics	Count per country	Y	Y	Y	Y
	Count per AS	Y	N	Y	N
	Count per address	Y	N	Y	N
	Count of consumer's own domain	Y	Y	Y	N

Table 4. The summary of the evaluation.

Scenario	Advantage	Related DATA-CONTENT
1. Witty worm	Prevent worm activity	Incoming src port
2. Backscatter	Find victims quickly	Stats of own domain

some fields may contain sensitive data. In fact, packet traces in many datasets have anonymized their IP addresses and also packet traces of the UCSD network telescope are offered to only restricted researchers. In contrast, u-message in our proposed scheme can be used for offering these sensitive or non-sensitive information automatically. Filtering information in a u-message can be used for worm filtering in real-time.

C.2 Backscatter Scenario

In this scenario, there is a security concern about unintentionally revealing the addresses of one or more darknet monitors. The addresses of a darknet monitor must not be revealed. This requires care in publishing the information detected at the darknet monitor. Raw AS-wise counts may be used as a clue to infer darknet monitor addresses. To avoid giving out such clues, we can adopt two steps for offering event information detected at darknet monitors.

Table 3 shows the security requirement of statistical DATA-CONTENTs in a darknet. Monitors in the darknet measure the count of backscatter packets per country. The aggregate nature of the data makes it difficult to use probe packets to unveil darknet monitor addresses. If a manager sees a suspicious activity in his/her country, he/she may send a query about the count pertaining to his/her own AS to the manager of the darknet monitor. If the requesting manager has the access privileges, he/she will receive the response and will figure out whether there is DoS attack in his/her domain.

D. Discussion

Sharing of event information is useful for event based management. In Section I, we have presented feasibility related issues namely, the issues of scale, security and privacy. We argue that our design overcomes the problems. The issues of scale are addressed by the distributed DB design which become feasible due to the use of a common incident exchange format. The issues of security and privacy are addressed by a flexible security model where by access control is defined and exercised based on the DATA-CONTENT, CONSUMER and PROBE-POINT. We have demonstrated how the security model can be employed for packet traces and have discussed how it may be implemented using standard access control mechanisms as are available in LDAP. We have shown a few examples where sharing of event information is easily achieved.

Table 4 shows the summary of the evaluation. We showed that our proposed scheme helps in inhibiting witty worm activity and finding DoS attack victims quickly. The witty worm scenario can be extended to generic worm propagation and can be applied to worms like code red [9]. Although similar results may be realized using other means, our method has the advantage of usability in the wide Internet where access controls are exercised based on a flexible security model. This will prompt more managers to share event information and make the system more viable.

VI. RELATED WORKS

A. Concept of Event

Event is a commonly-used concept not only in network management area but also in many other areas. Many researches have used the term *event* but the definition, if any, and usage are different. In the area of mathematical time series analysis, an event is defined as a *change-point* in time series data [34]. This type of objective event definition is suitable for automatic event detection and processing. On the other hand, there are many subjective definitions, e.g., scene change in motion-image sequence analysis [35]. In this case, automated event detection may lead to a false positive or a false negative.

In this paper, we indicate the former event as an *objective event* or simply an *event*, and the latter event as a *subjective event*. In the network management area, examples of objective events are SNMP [36] alert, IDS signature matching, firewall filter matching, and so on. A subjective event is subjectively judged by a human manager by correlating and analyzing these objective events and their related information. Examples of subjective events are flash crowd, attack, intrusion, and so on. We intend to share objective events and their event information about UDP packets or backscatter stats, and to analyze subjective events like witty worm or DoS attacks.

B. Event based Network Management

Many network management methods and systems work based on events, especially fault management, performance management and security management [1]. A good example of an event based management system is an IPS, in which managers define some signatures of malformed packets previously and packets that match a signature are automatically dropped. It has been mainly used for protecting a network against infection of viruses and worms. In this case, the detection of a packet that matches a signature is an objective event, and the act of dropping the packet is one of the security management tasks. Many spam filtering and anomaly detection systems [37] have a similar scheme. These management methods have their own event definitions, and event information in general includes traffic data and service state.

C. Information Sharing for Event based Network Management

Managers are primarily concerned about the behavior of their own networks. The Internet being a very large distributed conglomeration of autonomous systems, events in the networks that

comprise the Internet affect each other, so event information sharing, is very useful for management.

There are some information sharing systems. Vipul's Razor [38] is a network system for collaborative spam detection and filtering. It shares information on spam to make the spam-filtering rules of each organization more effective and precise. DNSBL systems [21], [22] have been used to share information about black-listed spam senders. DShield [39] has been used to share firewall log information from many networks to reveal the global trend of malicious traffic.

As we mentioned in subsection II-C, these existing information sharing systems are designed for specific applications and as such have problems with extensibility and reusability for more generic applications. In contrast, IODEF [4] is standardized as RFC 5070, which is a format for representing computer security information commonly exchanged between computer security incident response teams (CSIRTs). Our proposed system is not limited to a particular data type because of the generic nature of IODEF. Moreover, our system can be used in an open environment; it has provisions for security and privacy as described in subsection III-C whereby one can offer sensitive data to restricted members and non-sensitive data to the public at large.

D. Access Control for Information Sharing

UniDAS adopted a role-based access control (RBAC) [40] scheme with LDAP implementation because it is scalable and practical. Recently, more advanced temporal access control like temporal role-based access control (TRBAC) [41] have become available. Currently, we think that RBAC is enough for UniDAS, but we may adopt more advanced technologies in the future.

VII. CONCLUSION

In this paper, we addressed the issue of generalization and automation of cross-domain event information sharing. We used data sets which represent actual attacks in the operational Internet to understand the dynamics of the attacks and then showed the effectiveness of sharing incident related information to contain these attacks. The effectiveness of the scheme is measured in terms of the quickness of preventive measures and the consequent reduction in damage and disruption. We proposed an event information sharing scheme that handles events generated from a network management system. We implemented the scheme and showed that it is effective in real event analysis and is practical as it satisfies the basic security and privacy requirements.

We will start event information sharing with UniDAS among some laboratories in Tohoku university, Japan. Progressively, we will extend the sharing environment to other universities, CSIRTs like JPCERT and finally various domains in the world.

REFERENCES

[1] J. P. Martin-Flatin, G. Jakobson, and L. Lewis, "Event correlation in integrated management: Lessons learned and outlook," *J. Netw. Syst. Manage.*, vol. 15, no. 4, pp. 481–502, 2007.

[2] K. McCloghrie, D. Perkins, and J. Schoenwaelder, "Structure of management information version 2 (SMIv2)," RFC 2587, Apr. 1999.

[3] "Information technology—open systems interconnection—structure of management information: Guidelines for the definition of managed objects," ISO/IEC 10165-4:1992 / ITU-T X.722, 1992.

[4] R. Danyliw, J. Meijer, and Y. Demchenko, "The incident object description exchange format," RFC 5070, Dec. 2007.

[5] C. Shannon and D. Moore. (2004, Mar.). The CAIDA dataset on the witty worm. Cisco Systems, Limelight Networks, the US Department of Homeland Security, the National Science Foundation, DARPA, Digital Envoy, and CAIDA Members. [Online]. Available: http://www.caida.org/data/passive/witty_worm_dataset.xml

[6] C. Shannon, D. Moore, and E. Aben. (2007, Jan.–Nov.). The CAIDA backscatter-2007 dataset. [Online]. Avilable: http://www.caida.org/data/passive/backscatter_2007_dataset.xml

[7] CAIDA. [Online]. Available: http://www.caida.org/home

[8] The darknet project. [Online]. Available: http://www.team-cymru.org/Services/darknets.html

[9] C. C. Zou, W. Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *Proc. the 9th ACM Conf. Comput. Commun. Security*, 2002, pp. 138–147.

[10] C. C. Zou. (2004). Witty worm propagation modeling. [Online]. Available: http://tennis.ecs.umass.edu/ czou/research/wittyModel.html

[11] J. Kim, S. Radhakrishnan, and S. K. Dhall, "Measurement and analysis of worm propagation on Internet network topology," in *Proc. ICCCN*, 2004, pp. 495–500.

[12] C. Shannon and D. Moore, "The spread of the witty worm," *IEEE Secuity and Privacy*, vol. 2, no. 4, pp. 46–50, 2004.

[13] N. Weaver and D. Ellis, "Reflections on witty: Analyzing the attacker," *login:*, vol. 29, no. 3, pp. 34–37, 2004.

[14] C. C. Zou, D. Towsley, and W. Gong, "On the performance of internet worm scanning strategies," *Perform. Eval.*, vol. 63, no. 7, pp. 700–723, 2006.

[15] M. A. Rajab, F. Monrose, and A. Terzis, "On the effectiveness of distributed worm monitoring," in *Proc. SSYM*, Berkeley, CA, USA, 2005, pp. 15.

[16] S. Wei and J. Mirkovic, "Correcting congestion-based error in network telescope's observations of worm dynamics," in *Proc. IMC*, New York, NY, USA, 2008, pp. 125–130.

[17] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," in *Proc. the 10th Usenix Security Symp.*, 2001, pp. 9–22.

[18] Day in the life of the Internet. [Online]. Available: http://www.caida.org/projects/ditl

[19] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson, "The Internet motion sensor: A distributed blackhole monitoring system," in *Proc. NDSS*, Feb. 2005.

[20] Y. Shinoda, K. Ikai, and M. Itoh, "Vulnerabilities of passive internet threat monitors," in *Proc. SSYM*, Berkeley, CA, USA, 2005, pp. 14.

[21] MAPS RBL. [Online]. Available: http://www.mail-abuse.com

[22] RBL.JP. [Online]. Available: http://www.rbl.jp/index-e.php

[23] Internet scan data acquisition system (ISDAS). [Online]. Available: http://www.jpcert.or.jp/isdas

[24] SANS Internet storm center. [Online]. Available: http://isc.sans.org

[25] Snort. [Online]. Available: http://www.snort.org

[26] XSL transformations (XSLT). [Online]. Available: http://www.w3.org/TR/1999/REC-xslt-19991116

[27] "XML signature syntax and processing (second edition)," W3C Recommendation, 2008.

[28] "XML encryption syntax and processing," W3C Recommendation, 2002.

[29] "Web services description language (WSDL) version 2.0 part 1: Core language," W3C Recommendation, 2007.

[30] UDDI Version 3.0.2. (2004). [Online]. Available: http://uddi.org/pubs/uddi_v3.htm

[31] "SOAP version 1.2 part 1: Messaging framework (second edition)," W3C Recommendation, 2007.

[32] M. Wahl, T. Howes, and S. Kille, "Lightweight directory access protocol (v3)," RFC 2251, Dec 1997.

[33] CpMonitor. [Online]. Available: http://www.cysols.com/products/cpmonitor/index.html

[34] V. Guralnik and J. Srivastava, "Event detection from time series data," in *Proc. KDD*, New York, NY, USA, 1999, pp. 33–42.

[35] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 8, pp. 873–889, 2001.

[36] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple network management protocol (SNMP)," RFC 1157, May 1990.

[37] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. ACM SIGCOMM*, Philadelphia, Aug. 2005.

[38] Vipluś Razor. [Online]. Available: http://razor.sourceforge.net

[39] DShield. [Online]. Available: http://www.dshield.org

[40] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," in *Proc. the 15th National Comput. Security Conf.*, 1992, pp. 554–563.
[41] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal role-based access control model," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 191–233, 2001.

Masahiro Nagao received his Ph.D in Information Sciences from Tohoku University, Japan in 2009. His research interests include network management mechanisms, traffic analysis, log analysis, etc. He was a JSPS Research Fellow from April 2008 to March 2009.

Kazuhide Koide received his Ph.D. degree in Information Sciences from Tohoku University, Japan, 2006. He is currently the Assistant Manager of IP Network Department in KDDI Corporation. His research interests include IP network operation, monitoring, and management. He received Highly Commended Paper Award of AINA2006. He is a member of IEICE.

Akihiro Satoh received his Master's degree in Information Sciences from Tohoku University, Japan in 2008. Now he is in the Ph.D. program in Graduate School of Information Sciences (GSIS), Tohoku University. His research interests include network management mechanisms, traffic analysis, etc. He is a Student Member of IPSJ.

Glenn Mansfield Keeni received his Ph.D. in Logic programming, from Tohoku University, Japan. He is currently a Senior Visiting Researcher at the Research Institute of Electrical Communications, Tohoku University, Sendai, Japan and is President/CEO of Cyber Solutions Inc., Sendai, Japan. His research interests include expert systems, computer networks and their management, security, etc. He is a Member of the ACM, the IEEE Communications Society and is an Active Member of the IETF.

Norio Shiratori is currently a Professor at Research Institute of Electrical Communication (RIEC), Tohoku University, Japan. Before moving to RIEC in 1993, he was the Professor of Information Engineering at Tohoku University from 1990 to 1993. Prior to that, he served as an Associate Professor and Research Associate at RIEC, Tohoku University, after receiving his Doctoral degree from Tohoku University in 1977. He has also served as the Vice-Director of RIEC, and IFIP representative of Japan. Now he is the President of IPSJ (Information Processing Society of Japan). He is a Fellow of IEEE, IPSJ and IEICE. He also contributes through serving in various capacities, such as: General Chair of the 9th IEEE ICOIN-9(1994), IFIP Joint International conference FORTE/PSTV'97, and 12th IEEE ICOIN-12 (1997); Program Chair of ICPADS'96 (1996) and ICPP-99 (1999). He was one of the leaders in Japan Gigabit Network (JGN) national project and headed several national projects such as, SCOPE—funded by Ministry of Internal Affairs and Communications and Dynamic Networking project—sponsored by JSPS. He is currently leading two other national projects. Besides that, in 2006, the proposed idea of his research group on Mobile IPv6 was approved and standardized by IETF. He has proposed a new concept of Flexible Computing and still working in this direction. His recent research interest is in Ubiquitous and Symbiosis computing. He has published more than 15 books and over 400 refereed papers in computer science and related fields. He was the recipient of IPSJ Memorial Prize Wining paper award in 1985, Telecommunication Advancement Foundation Incorporation Award in 1991, Best Paper Award of ICOIN-9 in 1994, IPSJ Best Paper Award in 1997, and many others including the most recent Outstanding Paper Award of UIC-07 in 2007.