The Secure Transfer of the Cooperative Data in Network-Based Intrusion Detection System*

Fan Jun(樊隽)[1], Jin Ye(金烨)

(Southeast University, Computer Science Department., 210096, Nanjing, PRC)

Abstract: This paper discusses the design and implementation of a secure transfer of the cooperative data in a network-based intrusion detection system. It makes the data transfer of security cooperation available, and ensures the security of all the communicative channels between different cooperation spots as well as supplying the base of ensuring IDS to accurately describe occurred security incidents to some extent. It provides reliable transfer services for security detection and security tracing that it becomes an indispensable part of the network-based intrusion detection system.

Key words: cooperative detection, IDXP (Intrusion Detection Exchange Protocol), XML(Extensible Markup Language), TLS(Transport Layer Security), IDS(Intrusion Detection System)

CLC number: TP393;　**Document code**: B;　**Article ID**: 1672－1292(2002)03－0042－04

0 Introduction

With the rapid development of the network, network-based intrusions become severer and severer. So it is imperious to develop reliable network-based intrusion detection system. Since the dependence between diverse networks become severe, it is essential to import distributed cooperative mechanism into the IDS, which need to exchange cooperative data among different detection spots. To ensure the reliability and correctness of cooperation, the transfer of the cooperative data must be carefully controlled. Thus cooperative detection transfer is the base of the implementation of cooperative detection. In general it is divided into two layers: the lower layer is used to transfer data and ensure the communication channel's security of diverse cooperative spots; while the upper layer uses XML encoding program to encapsulate the security events, which is obtained from lower layer, and provides the results for cooperative detection system.

Untill now the most famous researches in network security detection are CIDF(Common Intrusion Detection Framework) which is held up by U. C. Davis and IDMEF(Intrusion Detection Message Exchange Format) produced by IDWG(Intrusion Detection Work Group) of IETF. Here the design of cooperative detection transfer is based on IDWG.

IDWG released "draft-ietf-idwg-idxp-01" to give a detailed description of IDXP (Intrusion Detection Exchange Protocol) in 2001 and in the next year released the newest version of "draft-ietf-idwg-idxp-04". IDXP is an application-level protocol for exchanging data between intrusion detection entities. It supports mutual-authentication, integrity, and confidentiality over a connection-oriented protocol. The protocol enables the exchange of IDMEF messages, unstructured text, and binary data. It is independent from data representation and makes the data transfer safe in high-speed IP network. Here we use IDXP as the lower

* **Received date**: 2002－09－15.

Foundation item: The project supported by the 863 program of China under grant No. 863317013399.

Biography: Fan Jun, born in Nov. 1978, female, master graduate, Southeast University, Computer Science Dept., Network Center, major in network security.

layer transfer mechanism of TCP. It supports the transmission of the alarm from channel sensor or analyzer to the manager. The security of IDXP is provided by TLS (Transport Layer Security). The security detection framework of IDWG is demonstrated by Fig. 1. IDMEF messages is a canonical data representation that should be used for the structured representation of the intrusion detection data. We use XML to implement IDMEF because XML is flexible and there are many tools which are able to deal with XML.

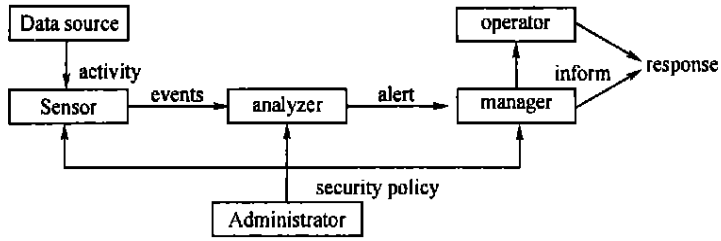

Fig. 1 Security detection framework of IDWG

1 Overall design of cooperative detection transfer

Cooperative detection transfer is built upon IP layer and under application layer, a connection-oriented transfer based on TCP. It is divided into two parts: client and server. The server is a dependant process which supports TLS and IDXP. It can verify the request to ensure whether the applicant owns a trusted certificate. It is the same that the applicant named client also supports TLS and IDXP. The client validate the server to ensure whether the server has a trusted certificate. The overall design is just shown by Fig. 2.

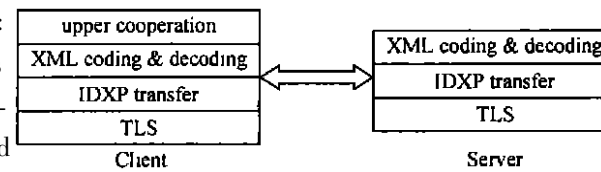

Fig. 2 Overall design

As an applicant the client authenticates the cooperative detection server first to obtain the server's trustable certificate. After coding the security events which cooperative policy requires to cooperate with by XML, the client sends it to the diverse cooperative spots through IDXP to get positive or negative affirm.

When the server receives cooperative detection request sent by the client, it first uses XML decoding program to get the security events needed to verify. Then it queries the event in the security events database to find whether there is a matching. If so the server will code the first matching security events by XML and give a positive reply to client. Otherwise it will give a negative reply. The flow chart is demonstrated by Fig. 3.

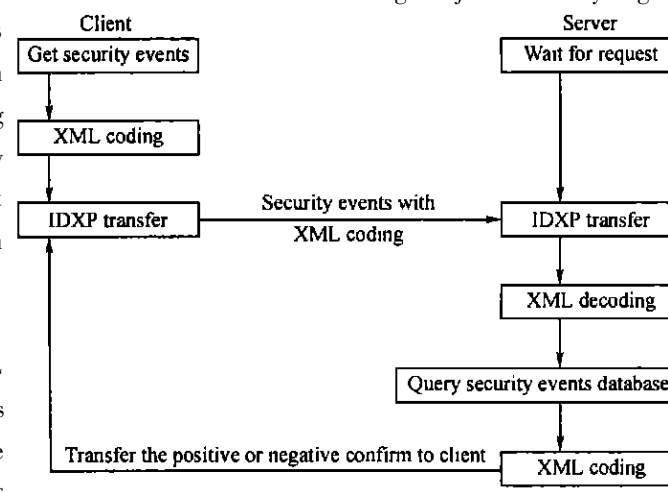

Fig. 3 Flow chart of cooperative detection transfer

2 Implementation of cooperative detection transfer

The cooperative detection transfer system includes server and client. The server is the manager and client is the sensor and analyzer, which we call it SA. In the implementation of the system, we use IDXP-v4

and TLS-v1. We divide the system into two parts: the base surrounding supports the data transfer and the upper interfaces. That base surrounding is IDXP transfer.

2.1 IDXP transfer mechanism

The cooperative detection transfer module is implemented in C programming. It provides an API for the upper layer to use. The mechanism of design is based on IDXP, a protocol used to transport encapsulated data and TLS, a protocol used for security authentication. The framework of the module is shown in Fig. 4.

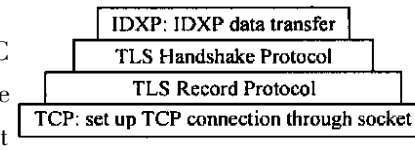

Fig. 4 Basic framework

The procedure of data transfer between cooperative detection spots can be separated into four parts: Protocol setup、Security setup、Secured data transport and termination.

2.1.1 Protocol setup

We can use IDXP peers to refer to the intrusion detection entities using IDXP to transfer data. That is to say, server and client are some kinds of peers. First, the two peers build a normal TCP connection by using socket. Then two peers exchange some transfer parameters by using request-response form. An IDXP peer wishing to establish IDXP communications with another IDXP peer does so by opening a BEEP channel, which may initiate a BEEP (Blocks Extensible Exchange Protocol) session.

2.1.2 Security setup

After the protocol initiating, a BEEP security profile offering the required security properties should initially be negotiated. In our model, trust is placed exclusively in the IDXP peers. A BEEP security profile is used to establish end-to-end security between pairs of IDXP peers. Only after successful negotiation of the underlying security profile are IDXP peers to be trusted. Here we use TLS - v1.0 as the BEEP security profiles to secure a BEEP session containing channels using the IDXP profile. In the implementation we use s-server and s-client modules provided by Openssl 0.9.6 to complete TLS handshake. When the two peers verify each other the client query whether the server has a certificate. If so the client will validate the certificate to determine whether it is trustable. The server also authenticates the certificate of the client. If the authentication succeeds the initial of the protocol and the security authentication complete. Thus following the successful negotiation of the BEEP security profile, IDXP greetings are exchanged and connection provisioning proceeds. Now the security session is ready for the data transfer. Here we use PKI(Public Key Infrastructure) and PGP(Pretty Good Privacy) to manage the private and the public key and the certificate is applied from NENC CA. For some real-time systems which need high level security, we can use OCSP (Online Certificate Status Protocol) mechanism to authenticate the certificate.

2.1.3 Secured data transport

According to UML modeling mode, we use IDMEF to code the transfer data so that it meets the requirements of semantics interoperability. The receiver decodes it by using a XML parse. Between the pair of IDS entities communicating over a BEEP session, one or more BEEP channels may be opened using the IDXP profile. We can use one channel to transfer the control data and some other channels to transfer diverse information. If desired categorization and prioritization of data sent between IDXP peers can be set, however, in most situations additional channels using the IDXP profile should be opened within an existing BEEP session, as opposed to provisioning a new BEEP session, containing the additional channels using the IDXP profile.

2.1.4 Termination

Both of the IDXP peers may choose to terminate the communication. The peer wanting to terminate

sends a "close" element on channel zero indicating which channel is to be closed to the other peer and waits for the response. An IDXP peer may also choose to close an entire BEEP session by sending a "close" element indicating that channel zero is to be closed. After receiving the response it can terminate the channel.

2.2 Upper interface

The reason for constructing the cooperative detection transfer module is just to provide an API for security cooperation. The format of the client API is just like this:

int handle-request(int* length , struct sockaddr _ in sa)

If fails the returning result is 0, otherwise the returning result is 1 and the response of server is put in the buffer.

The format of the server API is just like this:

int handle-request(int length)

It is used to handle the intrusion alert from the client and send the process results back to the client. If succeed the returning result is 1, otherwise the returning result is 0.

3 Conclusion

For today's network, the importance of describing the security events all-around becomes more and more prominent and the request of identity authentication becomes more and more veracious. The article is just a try towards these problems. The cooperative detection transfer system can transport the data coded by IDMEF between client and server on a secure channel. Enhancement of capability of some other cooperative functions such as co-adjust will help us greatly in network research and provide important basis for network security.

[**References**]

[1] Gong Jian, Lu Sheng, Wang Qian. Computer Network Security Conspectus[M]. First Edition. Published by Southeast University, Nanjing: Song Zengmin, 2002, 1~ 290.

[2] Douglas E Comer, David L Stevens. Internetworking with TCP/IP Vol. III: Client-Server Programming and Applications [M]. Second Edition, Prentice Hall Inc, 1998, 1~ 403.

[3] Charlie Kaufman, Radia Perlman, Mike Speciner. Network Security: Private Communication in a Public World[M]. First Edition. Prentice Hall Inc, 1995, 1~ 205.

[4] Liu Jianhang. The PKI based CA[Degree dissertation][D]. Preserved in Nanjing: Southeast University. 1999.

[5] J Callas. OpenPGP Message Format[DB/OL]. http:// www. ietf. org, 1998/ 2002.

[6] B Feinstein. The Intrusion Detection Exchange Protocol (IDXP) [DB/OL]. http:// www. ietf. org, 2002/ 2002.

网络入侵检测中协同信息的安全传输

樊隽, 金烨

(东南大学计算机系, 210096, 南京)

[摘要] 介绍了一个面向网络入侵检测协同信息的安全传输系统的设计与实现, 它能够支持检测协同底层的数据传输并保证各协同站点间的信道安全, 从而实现不同 IDS 之间的信息安全传输, 为监测协同和安全追踪提供可靠传输服务, 是基于入侵网络检测系统中必不可少的一个组成部分。

[关键词] 协同检测, IDXP, XML, TLS, IDS

[中图分类号] TP393; [文献标识码] B; [文章编号]: 1672- 1292(2002) 03- 0042- 04

[课题支持] 本项目受国家"863- 317- 01- 33- 99"课题资助

[责任编辑: 蓉]