네트워크 취약점 검색공격 탐지 시스템을 위한 안전한 통신 프레임워크 설계

유일선[†]·김종은[††]·조경산[†††]

요약

본 논문에서는 취약점 검색공격 탐지시스템 DS-NVSA(Detection System of Network Vulnerability Scan Attacks)에서 서버와 에이전트들 사이의 상호연동을 위한 안전한 통신 프레임워크를 제안한다. 기존 시스템과의 상호연동을 위하여 제안 프레임워크는 IETF의 IDWG에서 제안한 IDMEF와 IAP를 확장 적용하였다. 또한 공개키 기반의 환경을 지원하지 못하는 네트워크 시스템을 위해 대칭키 기반의 암호화 통신 프로토콜 SKTLS(Symmetric Key based Transport Layer Security Protocol)를 제시하였다. 제안된 프레임워크는 DS-NVSA 이외에도 기존의 이기종 침입탐지 시스템의 재사용과 탐지 영역의 확대를 제공하며, 또한 기업내 통합 보안환경시스템 ESM(Enterprise Security Management) 시스템에도 적용될 수 있다.

A Secure Communication Framework for the Detection System of Network Vulnerability Scan Attacks

Il-Sun You[†] · Jongeun Kim[††] · Kyungsan Cho[†††]

ABSTRACT

In this paper, we propose a secure communication framework for interaction and information sharing between a server and agents in DS NVSA (Detection System of Network Vulnerability Scan Attacks) proposed in [1]. For the scalability and interoperability with other detection systems, we design the proposed framework based on IDMEF and IAP that have been drafted by IDWG. We adapt IDMEF and IAP to the proposed framework and provide SKTLS (Symmetric Key based Transport Layer Security Protocol) for the network environment that cannot afford to support public-key infrastructure. Our framework provides the reusability of heterogeneous intrusion detection systems and enables the scope of intrusion detection to be extended. Also it can be used as a framework for ESM (Enterprise Security Management) system.

키워드 : 침입 탐지(Intrusion Detection), 네트워크 취약점 분석(Network Vulnerability Analysis), 네트워크 보안(Network Security)

1. Introduction

Computer hackers must conduct a lot of research to successfully gain privileged access to computers over the network. This attitude to gather intelligence before attempting to break in is called network vulnerability scan attack [1].

Today, the number of automated network vulnerability scanners such as mscan, sscan and nmap is constantly increasing and more attacks are successfully initiated [2-4]. To protect from those automated scanners, many tools such as scanlogd, snort and RTSD have been developed. However, they have problems in detecting slow scans, coordinated scans, slow coordinated scans, vulnerable port scans and so forth because they depend on the specified number of packets received from one host in a specified period of time [2, 4]. Also they cannot provide a hierarchical detection and response capability to counter attacks occurring across large-scale networks because they are almost stand-alone systems. In [1], DS-NVSA (Detection System of Network Vulnerability Scan Attacks) was proposed to solve the problems mentioned above. However, DS-NVSA has security threats such as masquerade, modification of messages, fabrication of intrusion alert messages or response messages, repudiation, denial of service in exchanging messages between its server and its agents. Moreover,

※ 이 연구는 2003학년도 단국대학교 대학 연구비의 지원으로 연구되었음.

† 정회원 : 단국대학교 대학원
†† 종신회원 : 단국대학교 대학원
††† 종신회원 : 단국대학교 정보컴퓨터학부 교수

논문접수 : 2002년 9월 9일, 심사완료 : 2002년 12월 18일

it lacks the standardized way to enable interoperability with other intrusion detection systems and allow users to mix-and-match the deployment of these systems according to their strong and weak points to obtain an optimal implementation.

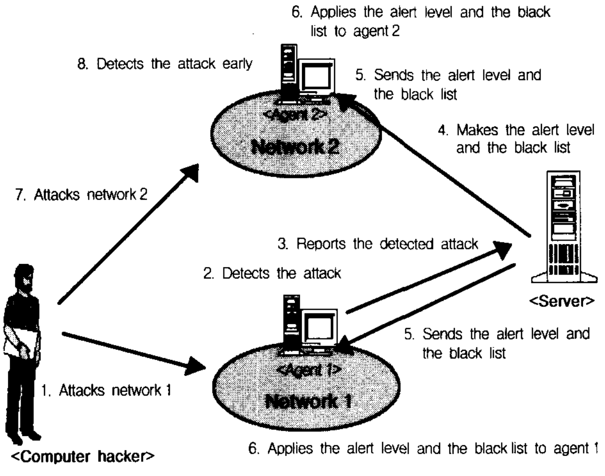
In this paper, we propose a secure communication framework for interaction and sharing information between a server and several agents in DS-NVSA, which stands against the threats mentioned above and provides a standardized way.

The rest of the paper is organized as follows. Section 2 describes the related work. In section 3, we describe DS-NVSA, and then define the requirements of a secure communication framework for DS-NVSA. In section 4, we propose a secure communication framework. Section 5 analyzes the proposed framework based on the requirements defined in section 3. Finally, section 6 provides our conclusions and outlines future work.

2. Related Work

As intrusions and other attacks become more widespread and more sophisticated, it becomes beyond the scope of a stand-alone intrusion detection system to deal with them [4]. Thus, the need arises for systems to cooperate with one another and to manage diverse attacks across networks. As a result, several distributed intrusion detection systems such as DIDS, GRIDS, EMERALD, AAFID were proposed [11]. But there are still some problems. First, these systems have focused primarily on homogeneous components, with little attention toward standardization. Second, there are threats such as masquerade, modification of messages, fabrication of intrusion alert messages or response messages, repudiation, and denial of service when messages are exchanged among different intrusion detection systems. Works to solve such problems have been made as follows. In 1997, a group of research projects funded by DARPA began a collaborative effort called the Common Intrusion Detection Framework (CIDF). The motivation of CIDF was to provide an infrastructure that allows different intrusion detection and response (IDR) components to interoperate and share information and resources[4, 13, 18]. A communication framework and a common intrusion specification language (CISL) are provided to assist interoperation among CIDF components [14]. The Common Intrusion Specification Language (CISL) is a way for intrusion detection systems to express information about events, attacks and responses. It is designed to be flexible and efficient for the application programmer and uses a syntax called S-expressions. Some of the ideas involved in CIDF have encouraged the creation of an Internet Engineering Task Force (IETF) working group, named the Intrusion Detection Working Group (IDWG) [10]. Though inspired by the desire to share the ideas of CIDF in a wider community, IDWG is now a separate activity and may or may not use the results of CIDF. The purpose of IDWG is to define data formats and exchange procedures for sharing information [12].

After defining requirements of data formats and exchange procedures, the IDWG proposed the Intrusion Detection Message Exchange Format (IDMEF) as a standard data format that intrusion detection systems can use to report alerts about events that they deem suspicious, and two protocol specifications (IAP and IDXP) fulfilling the IDWG transport protocol (IDP) requirements for communicating IDMEF messages [5, 7, 12, 17].


6. Applies the alert level and the black list to agent 2
8. Detects the attack early
5. Sends the alert level and the black list
<Agent 2>
Network 2
4. Makes the alert level and the black list
7. Attacks network 2
3. Reports the detected attack
2. Detects the attack
<Server>
5. Sends the alert level and the black list
<Agent 1>
Network 1
1. Attacks network 1
<Computer hacker>
6. Applies the alert level and the black list to agent 1

(Figure 1) Hierarchical detection processing of DS-NVSA

3. DS-NVSA (Detection System of Network Vulnerability Scan Attacks)

In this section, we provide an overview of DS-NVSA, and then redefine the requirements of a secure communication framework for DS-NVSA.

3.1 Overview of DS-NVSA

DS-NVSA improves conventional algorithms based on the specified number of packets received from one host in a given period of time and provides a hierarchical detection and response capability to counter attacks occurring across large-scale networks as shown in (Figure 1) [1].

The server computes alert levels using the attack information reported from agents within the given period. The alert level is measured by the attacked agents and also by the frequency of the attacks reported. If the alert level indicates the critical status of the network, the server notifies the alert level to the agents. In addition, the server can send black list to agents when it is requested by administrators or depending on the alert level. The black list is composed of IP addresses and port numbers that are largely used in the recent attacks or recommended to be watched by the administrators. Receiving a response message, the agents adjust the alert level and apply the black list to their detection policies. As a result, they watch packets more closely and get higher probability of detecting packets with the IP addresses or port numbers in the black list.

3.2 Requirements of A Secure Communication Framework for DS-NVSA

DS-NVSA needs a secure communication framework that handles the security threats and provides a standardized way to enable interoperability with other intrusion detection systems. Before presenting the secure communication framework for DS-NVSA, we redefine the following requirements of the framework based on IDWG transport protocol (IDP) requirements for communicating IDMEF messages [12, 15].

3.2.1 Reliable Transmission

As DS-NVSA relies on the alert or response messages sent, the framework should make sure that the messages are delivered reliably. The IDWG specified that IDP be based on TCP.

3.2.2 Operate through Firewalls without Compromising Security

Since it is expected that firewalls will often be deployed between the components of DS-NVSA, the framework should have the ability to send messages through firewalls without compromising security.

3.2.3 Mutual Authentication/Assurance of Message Origin

Components of DS-NVSA must be able to verify the identity of their peer. Assurance of message origin involves a way to prove which messages came from which component of DS-NVSA.

3.2.4 Integrity and Confidentiality

The framework should guarantee both the integrity and confidentiality of its data.

3.2.5 Resist DOS Attacks

A common way to defeat secure communication systems is through resource exhaustion, and it can prevent any communication at all. It is desirable that the framework resists such denial of service attacks.

3.2.6 Resist Malicious Duplication of Messages

A common way to impair the performance of secure communications mechanisms is to duplicate messages being sent, even though the attacker might not understand them. It is desirable that the framework resists such message duplication.

3.2.7 Interoperability

The framework should provide the standard message formats and message exchange procedures that provide interoperability with other systems.

3.2.8 Flexible Secure Protocols

In addition to the secure protocols (such as TLS) based on public-key infrastructure, the framework should provide secure protocols for the network system that cannot afford to support public-key infrastructure.

4. Design of A Secure Communication Framework for DS-NVSA

In this section, we propose a secure communication framework for DS-NVSA in accordance with the requirements mentioned above. For the scalability and inter operability with other systems, we design the framework based on IDMEF and IAP that have been drafted by IDWG. First, we design the message formats and message exchange procedures for DS-NVSA, and then modify IDMEF and IAP for them.

4.1 Message Format

Messages exchanged between the server and agents in DS-NVSA are classified into heartbeat messages, alert messages and response messages.

4.1.1 Heartbeat Message

Heartbeat messages are what the agents use to indicate their current status of starting, running or stopping to the server. Heartbeat messages are supposed to be sent at regular interval, say every ten minutes or every hour. The lack of a heartbeat message indicates that the agent or its network connection has failed. A heartbeat message is structured as (Figure 2) shows.

Sender ID	Current Time	HFlag

* HFlag = one of {HStart, HStop, HNormal}

(Figure 2) Heartbeat message

4.1.2 Alert Message

Alert messages are what the agents use to report the detected attacks to the server and are structured as (Figure 3) shows.

Sender ID	Current Time	Attack Type	Attack Inform

(Figure 3) Alert message

4.1.3 Response Message

Server uses Response Messages in (Figure 4) to recommend agents to apply the alert level and black list to their detection policies.

Sender ID	Current Time	Alert Level	Black List

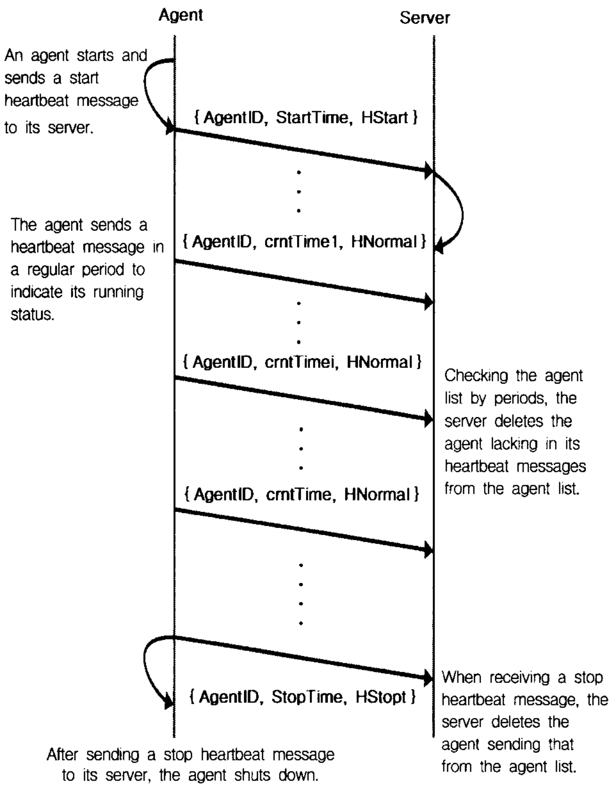
(Figure 4) Response message

4.2 Message Exchange Procedure

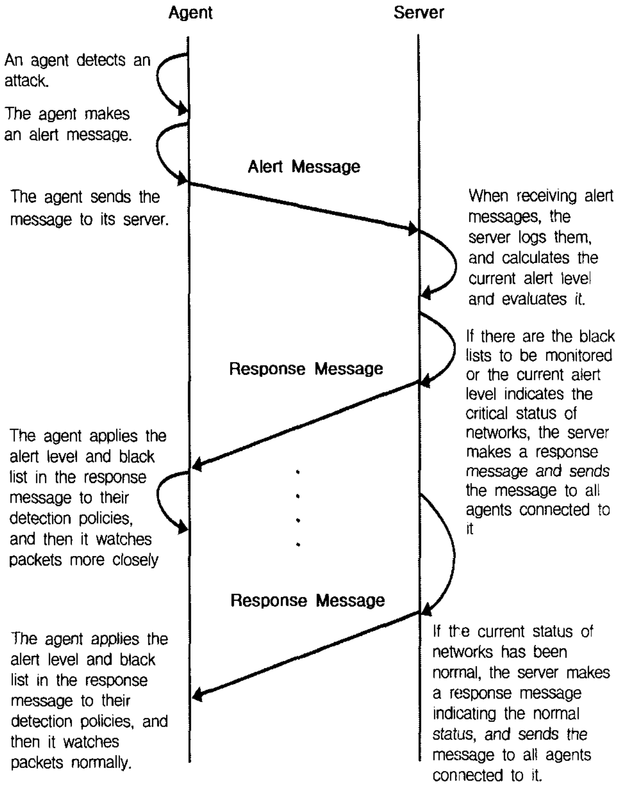
Message exchange procedure is composed of heartbeat message exchange procedure and alert/response message exchange procedure. (Figure 5) shows heartbeat message exchange procedure and (Figure 6) shows alert/response message exchange procedure.

4.3 Adaptation of IDWG Framework

For the interoperability with other systems, we adapt IDMEF and IAP proposed by the IDWG to our proposed framework. Because of XMLs extensibility and wide availability of software tools for parsing and validating XML, XML-based IDMEF is preferred for this work [5]. Furthermore, as IAP can fulfill all requirements defined in section 3 except the requirement of flexible secure protocols [16]. However, we cannot apply IDMEF and IAP directly to our message format and message exchange procedure defined above, because IDMEF does not support the heartbeat message and response message defined in 4.1. As based on TLS (Transport Layer Security Protocol), IAP does not support the network system that cannot afford to deploy public-key infrastructure. We improve IDMEF and IAP to go beyond this limitation and provide SKTLS (Symmetric Key based Transport Layer Security Protocol) as an alternative to TLS.


Agent
Server
An agent starts and sends a start heartbeat message to its server.
{ AgentID, StartTime, HStart }
The agent sends a heartbeat message in a regular period to indicate its running status.
{ AgentID, crntTime1, HNormal }
{ AgentID, crntTimei, HNormal }
Checking the agent list by periods, the server deletes the agent lacking in its heartbeat messages from the agent list.
{ AgentID, crntTime, HNormal }
{ AgentID, StopTime, HStopt }
When receiving a stop heartbeat message, the server deletes the agent sending that from the agent list.
After sending a stop heartbeat message to its server, the agent shuts down.

(Figure 5) Heartbeat message exchange procedure


Agent
Server
An agent detects an attack.
The agent makes an alert message.
Alert Message
The agent sends the message to its server.
When receiving alert messages, the server logs them, and calculates the current alert level and evaluates it.
If there are the black lists to be monitored or the current alert level indicates the critical status of networks, the server makes a response message and sends the message to all agents connected to it
Response Message
The agent applies the alert level and black list in the response message to their detection policies, and then it watches packets more closely
Response Message
If the current status of networks has been normal, the server makes a response message indicating the normal status, and sends the message to all agents connected to it.
The agent applies the alert level and black list in the response message to their detection policies, and then it watches packets normally.

(Figure 6) Alert/Response message exchange procedure

4.4 Improving IDMEF

To be applied to our work defined in 4.1 and 4.2, IDMEF must fulfill the following requirements. First, heartbeat messages should support status such as HStart, HStop and HNormal. Second, in addition to alert messages and heartbeat messages, it should support response messages that are composed of alert level and black list.

4.4.1 Extending Heartbeat Message

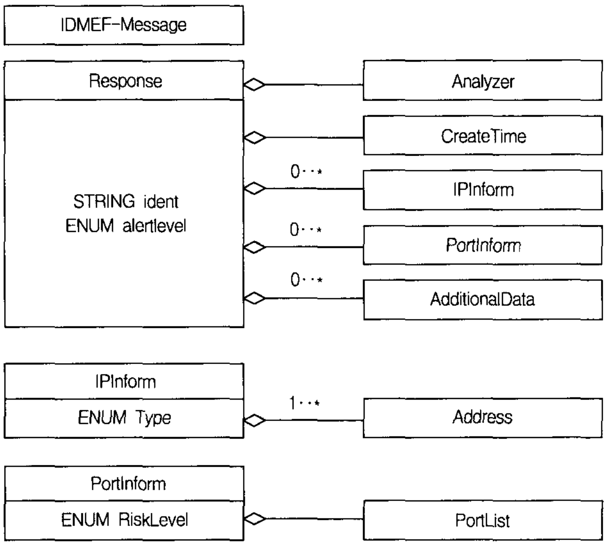
We add an hflag attribute to heartbeat messages for supporting HStart, HStop and HNormal. The DTD of heartbeat messages is modified as shown in (Figure 7).

```
<!ELEMENT Heartbeat                    (
        Analyzer, CreateTime, AnalyzerTime?, AdditionalData*
)>
<!ATTLIST Heartbeat
        ident          CDATA          '0'
        hflag               (hstart|hnormal|hstop)        'hnormal'
>
```

(Figure 7) DTD of modified heartbeat message

4.4.2 Definition of Response Message

We define a response message class which inherits from IDMEF-Message class and consists of Analyzer, CreateTime, IPInform, PortIn-form and AdditionalData class as (Figure 8) shows.



(Figure 8) Response message class

The DTD of response messages is shown in (Figure 9). Analyzer is the class for the server sending response messages to agents and CreateTime is the class showing when a response message is created. IPInform is the class for black list and its DTD is structured as (Figure 10) shows. If the IPType attribute of IPInform is black, agents add the addresses in IPInform to their own black list. If the IPType attribute of IPInform is normal, agents delete the addresses in IPInform from their own black list. PortInform is what the server sends to notify the risk level of the specified port number to its agents and is represented in the XML DTD as shown in (Figure 11). Response messages include an ident attribute and an alertlevel attribute. The alertlevel attribute shows the current alert level of networks.

```
<!ENTITY % attvals.alertlevel
    "( AlertLevel 1 | AlertLevel 2 | AlertLevel 3 | AlertLevel 4 |
     AlertLevel 5) "
>
<!ELEMENT Response (
    Analyzer, CreateTime, IPInfrom*, PortInform*, AdditionalData*
)>
<!ATTLIST Response
    ident        ID                 #IMPLIED
    alertlevel   %attvals.alertlevel ; 'AlertLevel 1'
>
```

(Figure 9) DTD of response message

```
        <!ELEMENT IPInform (Address +) >
        <!ATTLIST IPInform IPType (black|normal) 'black'>
```

(Figure 10) DTD of IPInform

```
<!ELEMENT PortInform (portlist) >
<!ATTLIST PortInform
     RISKLevel (RiskLevel 1 | RiskLevel 2 | RiskLevel 3 | RiskLevel 4 |
              RiskLevel 5)
     'RiskLevel 1'
>
```

(Figure 11) DTD of PortInform

4.5 Modifying IAP

We modify IAP to support various secure protocols and provide response messages and heartbeat messages.

4.5.1 Upgrade Request

To support various secure protocols in addition to TLS, we extend the Upgrade : header that is sent with iap-upgrade-request and the version of TLS to be used.

We replace "Upgrade : TLS/1.0" CRLF with "Upgrade : " Protocol CRLF.

4.5.2 Alert Content

iap-content-url specified as /iap/alert/ in iap-content-request cannot support response messages and heartbeat messages. To solve that, we replace iap-content-url = "/iap /alert/" with iap-content-url = ("/iap/alert/" | "/iap/heartbeat/" |"/iap/response/")

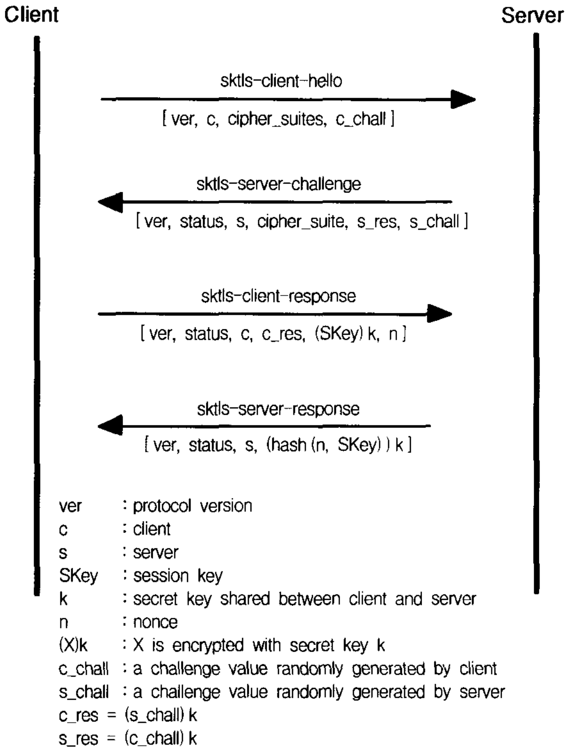
4.6 SKTLS(Symmetric Key based Transport Layer Security Protocol)

We present SKTLS to replace TLS as a secure protocol. Based on the symmetric key, SKTLS is not computationally expensive and complex. Also, it does not have difficult challenges such as verification of certificates, securely storing a private key, implementing and managing PKI, training users and so forth. Therefore, SKTLS enables our proposed framework to support network systems that cannot afford to support PKI due to some challenges mentioned above.

SKTLS has two phases ; connection setup and data exchange.

4.6.1 Connection Setup

SKTLS connection setup phase is composed of mutual authentication, selecting cipher algorithms and exchanging session keys. Its detailed procedure is as shown in (Figure 12). For mutual authentication, this phase uses a challenge-response type of OTP (One Time Password) based on secret keys shared between peers.

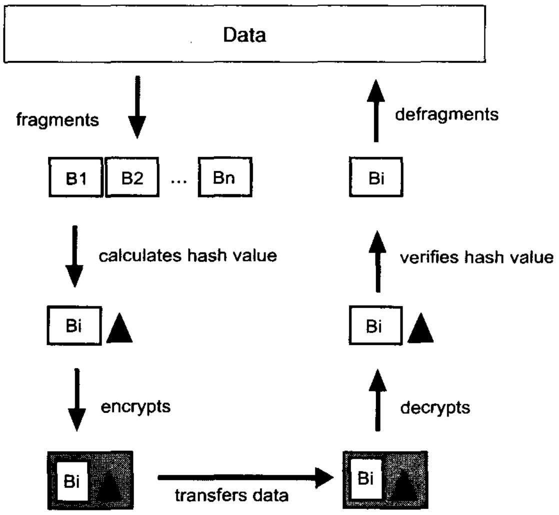

Client
Server
sktls-client-hello
[ver, c, cipher_suites, c_chall]
sktls-server-challenge
[ver, status, s, cipher_suite, s_res, s_chall]
sktls-client-response
[ver, status, c, c_res, (SKey) k, n]
sktls-server-response
[ver, status, s, (hash (n, SKey)) k]

ver : protocol version
c : client
s : server
SKey : session key
k : secret key shared between client and server
n : nonce
(X)k : X is encrypted with secret key k
c_chall : a challenge value randomly generated by client
s_chall : a challenge value randomly generated by server
c_res = (s_chall) k
s_res = (c_chall) k

(Figure 12) The connection setup of SKTLS

4.6.2 Data exchange

After the connection setup is completed, the data exchange phase begins. (Figure 13) shows the detailed procedure.



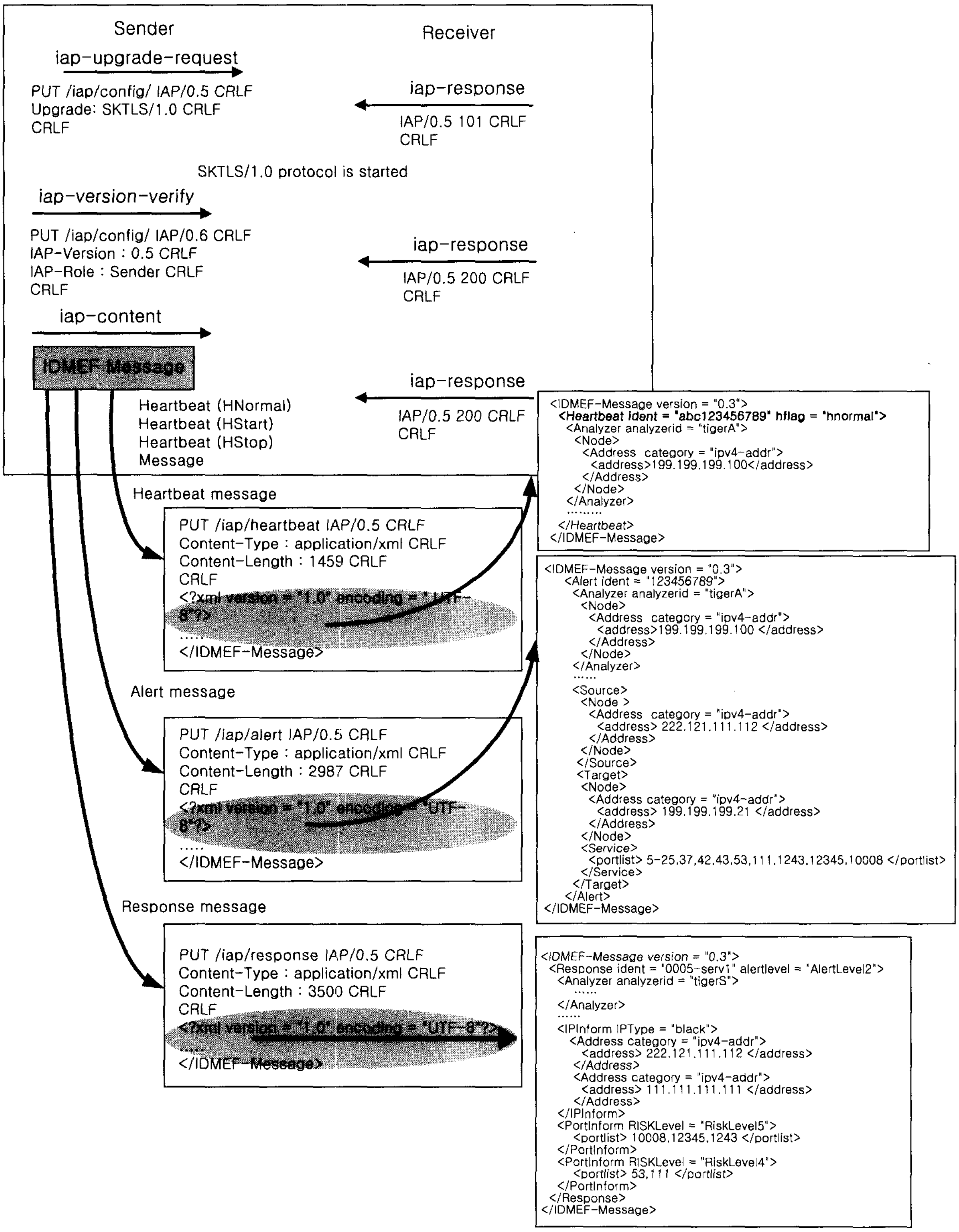
(Figure 13) Data exchange phase of SKTLS

5. Analysis of the proposed framework

In this section, we will look into the appliance of improved IAP and IDMEF in the proposed framework. Then on the basis of the requirements mentioned at section 3.2, a comparative analysis on our proposal will be presented.

5.1 Appliance of the Extended IAP and IDMEF

(Figure 14) shows an application of our proposed framework. While the agent is identified by the domain name 'A.IDS.NET' and the ID 'tigerA' with the network address 199.199.199.100, the server is specified by the domain name 'S.IDS.NET' and the ID 'tigerS' with the network address



```
PUT /iap/config/ IAP/0.5 CRLF
Upgrade: SKTLS/1.0 CRLF
CRLF
```

```
IAP/0.5 101 CRLF
CRLF
```

```
PUT /iap/config/ IAP/0.6 CRLF
IAP-Version : 0.5 CRLF
IAP-Role : Sender CRLF
CRLF
```

```
IAP/0.5 200 CRLF
CRLF
```

```
IAP/0.5 200 CRLF
CRLF
```

```
PUT /iap/heartbeat IAP/0.5 CRLF
Content-Type : application/xml CRLF
Content-Length : 1459 CRLF
CRLF
<?xml version = "1.0" encoding = "UTF-
8"?>
.....
</IDMEF-Message>
```

```
<IDMEF-Message version = "0.3">
 <Heartbeat ident = "abc123456789" hflag = "hnormal">
  <Analyzer analyzerid = "tigerA">
   <Node>
    <Address  category = "ipv4-addr">
     <address>199.199.199.100</address>
    </Address>
   </Node>
  </Analyzer>
  .........
 </Heartbeat>
</IDMEF-Message>
```

```
PUT /iap/alert IAP/0.5 CRLF
Content-Type : application/xml CRLF
Content-Length : 2987 CRLF
CRLF
<?xml version = "1.0" encoding = "UTF-
8"?>
.....
</IDMEF-Message>
```

```
<IDMEF-Message version = "0.3">
   <Alert ident = "123456789">
    <Analyzer analyzerid = "tigerA">
     <Node>
      <Address  category = "ipv4-addr">
       <address>199.199.199.100 </address>
      </Address>
     </Node>
    </Analyzer>
    ......
    <Source>
     <Node >
      <Address  category = "ipv4-addr">
       <address> 222.121.111.112 </address>
      </Address>
     </Node>
     </Source>
     <Target>
     <Node>
      <Address category = "ipv4-addr">
       <address> 199.199.199.21 </address>
      </Address>
     </Node>
     <Service>
      <portlist> 5-25,37,42,43,53,111,1243,12345,10008 </portlist>
     </Service>
    </Target>
   </Alert>
</IDMEF-Message>
```

```
PUT /iap/response IAP/0.5 CRLF
Content-Type : application/xml CRLF
Content-Length : 3500 CRLF
CRLF
<?xml version = "1.0" encoding = "UTF-8"?>
.....
</IDMEF-Message>
```

```
<IDMEF-Message version = "0.3">
 <Response ident = "0005-serv1" alertlevel = "AlertLevel2">
  <Analyzer analyzerid = "tigerS">
     ......
  </Analyzer>
  ......
  <IPInform IPType = "black">
    <Address category = "ipv4-addr">
      <address> 222.121.111.112 </address>
     </Address>
     <Address category = "ipv4-addr">
      <address> 111.111.111.111 </address>
     </Address>
  </IPInform>
  <PortInform RISKLevel = "RiskLevel5">
     <portlist> 10008,12345,1243 </portlist>
  </PortInform>
  <PortInform RISKLevel = "RiskLevel4">
     <portlist> 53,111 </portlist>
  </PortInform>
 </Response>
</IDMEF-Message>
```

(Figure 14) An application of the secure communication framework

199.199.199.130. We take SKTLS as a secure protocol. Messages are transferred through our proposed framework.

5.2 Analysis of the Proposed Framework

<Table 1> shows a comparative analysis of our proposal with the original IAP and IDMEF on the basis of the requirements of the secure communication framework.

As shown in [7, 16, 17], both original IAP and extended IAP can satisfy the requirements such as Reliable Transmission, Operation through Firewalls without Compromising Security, Mutual Authentication and Assurance of Message Origin, Integrity and Confidentiality, Resist DOS attacks, and Resist Malicious Duplication of Messages.

Our proposal can support various secure protocols in addition to TLS, and make it possible to express diverse messages mentioned above. So it is sure that our proposed framework is more appropriate to embody DS-NVSA. However, if we use SKTLS as a secure protocol, we cannot expect perfect service for non-repudiation due to the innate characteristics of the symmetric key structure. But in exchanging messages for intrusion detection and response, dissimilar to the case of E-commerce, we usually put emphasis on the point that the server could identify the message sender, and accordingly we can meet the need for non-repudiation by the mutual authentication in a sufficient level.

5.3 Using IDXP (Intrusion Detection eXchange Protocol)

At the 50th IETF, presentations were made on IAP, IDXP, a comparison of IAP and IDXP and TUNNEL (a BEEP profile used along with IDXP). The direction of the IDWG with regards to the future of these protocols was then discussed. A consensus reached at the meeting was that IDXP should become the IDWG message transfer protocol. Thus, IAP should remain as an Internet Draft as it is unless IDXP and TUNNEL prove themselves unworthy of meeting IDWG requirements and member expectations. Because such decisions were made during our research and IDXP had several problems with the implementation [17], we could not adopt IDXP as IDMEF message transfer protocol for our work. But it is not difficult to apply IDXP for our proposed framework. As implemented using the BEEP (Block Extensible Exchange Protocol) framework, IDXP provides flexibility in selecting a protocol for securing transport connections and support heart beat message and response message using heartbeat type attribute and config type attribute of streamType options in the IDXP profile without additional modification. The only thing to do for the proposed framework is to define and register SKTLS profile.

6. Conclusions and Future Work

In this paper, we propose a secure communication framework for interaction and information sharing between a server and several agents in DS-NVSA. After defining requirements of the secure communication framework for DS-NVSA, we design message formats and message exchange procedures for DS-NVSA in accordance with the requirements. Then, for the scalability and interoperability

〈Table 1〉 Comparing extended IAP and IDMEF with existing ones

Requirement		original	ours
Interoperability	Support standard format	○	○
	Heartbeat message	△	○
	Alert message	○	○
	Response Message	×	○
Reliable Transmission		○	○
Operate through Firewalls without Compromising Security		○	○
Mutual Authentication / Assurance of Message Origin		○	○
Integrity and Confidentiality		○	○
Resist DOS attacks		○	○
Resist Malicious Duplication of Messages		○	○
Flexible Selection of Various Secure protocols	PKI based	△	○
	SymmetricKey based	×	○

with other intrusion detection systems, we propose the framework to be based on IDMEF and IAP drafted by IDWG. Because IDMEF and IAP have some limitations to be applied to our proposal, we improve them. Furthermore, we provide SKTLS as an alternative to TLS for the flexible secure protocol. Based on symmetric key, SKTLS does not have shortcomings of PKI and supports the network system that cannot afford to support PKI.

By comparing with original IAP and IDMEF, we show that our proposed framework fulfills the requirements defined and is more appropriate for DS-NVSA.

Our framework provides the reusability of heterogeneous intrusion detection systems and enables the scope of intrusion detection to be extended. Also it can be used as a framework of ESM (Enterprise Security Management) system.

As a future work, we will apply IDXP to our proposed framework and implement it. Also, in order to show the usefulness of the framework, we will implement interfaces for other intrusion detection systems such as Snort, scanlogd and RTSD.

References

[1] Il-Sun You and Kyungsan Cho, "An Improved Detection System for the Network Vulnerability Scan Attacks," *The KIPS Transactions : Part C*, Vol.8-C, No.5, pp.543-550, 2001.

[2] Korea Information Security Agency, "Analysis of Large Scale Network Vulnerability Scan Attacks and Implementation of the Scan-Detection tool," 1999, http://www.certcc.or.kr.

[3] Korea Information Security Agency, "2001 Security incident Statistic in Korea," 2001, http://www.certcc.or.kr.

[4] Clifford Kahn, Don Bolinger and Dan Schnackenberg, "A Common Intrusion Detection Framework," 1998, http://www.isi.edu/~brian/cidf/drafts/communication.txt.

[5] D. Curry and H. Debar, "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition," <draft-ietf-idwg-idmef-xml-09.txt>, 2002.

[6] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0," RFC 2246, 1999.

[7] Dipankar Gupta, "IAP : Intrusion Alert Protocol," <draft-ietf-idwg-iap-05.txt>, 2001.

[8] Fielding, et al., "Hypertext Transfer Protocol-HTTP/1.1," RFC 2616, 1999.

[9] G. Mansfield and D. Curr, "Intrusion Detection Message Exchange Format Comparison of SMI and XML Implementations," <draft-ietf-idwg-xmlsmi-01.txt>, 2000.

[10] http://www.isi.edu/~brian/cidf/.

[11] J. Kim and P. Bentley, "The Artificial Immune Model for Network Intrusion Detection," 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT '99), http://www.cs.ucl.ac.uk/staff/J.Kim/publication.html, 1999.

[12] Mark Wood and Michael Erlinger, "Intrusion Detection Message Exchange Requirements," <draft-ietf-idwg-requirements-10.txt>, 2002.

[13] Peng Ning, Sushil Jajodia and Sean Wang, "Abstraction-based Intrusion Detection in Distributed Environments," *ACM Transactions on Information and System Security (TISSEC)*, Vol.4, Issue.4, pp.407-452, 2001.

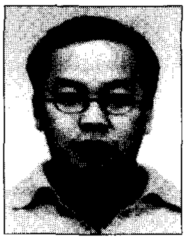
[14] Rich Feiertag, et al., "A Common Intrusion Specification Language (CISL)," http://www.isi.edu/~brian/cidf/drafts/language.txt, 1999.

[15] Stephen Northcutt, "Intelligence Gathering Techniques," http://www.microsoft.com/technet/security/intel.asp.

[16] Pollock, et al., "Implementing the Intrusion Detection Exchange Protocol," *Proceedings of the 17th Annual Computer Security Applications Conference*, http://www.acsac.org/2001/papers/67.pdf, 2001.

[17] B. Feinstein, G. Matthews, and J. White, "The Intrusion Detection Exchange Protocol (IDXP)," <draft-ietf-idwg-beep-idxp-07.txt>, 2002.

[18] Wenke Lee, et al., "A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions," *Proceedings of the 3rd International Workshop on the Recent Advances in Intrusion Detection*, pp.49-65, 2000.



유 일 선

e-mail : qjemfahr@security.co.kr

1995년 단국대학교 전산통계학과(이학사)

1997년 단국대학교 일반대학원 전산통계학과(이학석사)

2002년 단국대학교 일반대학원 전산통계학과(이학박사)

1997년~2000년 (주)한조엔지니어링 연구원

2000년~현재 (주)인터넷시큐리티 선임연구원

관심분야 : 침입탐지, 네트워크보안, 사용자 인증 및 접근통제

김 종 은

e-mail : semico@dankook.ac.kr
1995년 단국대학교 전자계산학과(이학사)
1997년 단국대학교 대학원 전산통계학과
(이학석사)
1998년~현재 단국대학교 대학원 박사과정
관심분야 : 컴퓨터 네트워크, 게임, 분산 시뮬레이션

조 경 산

e-mail : kscho@dankook.ac.kr
1979년 서울대학교 전자공학과(학사)
1981년 한국과학원 전기 및 전자공학과
(공학석사)
1988년 텍사스 대학교(오스틴) 전기 전산
공학과(Ph.D.)
1988년~1990년 삼성전자 컴퓨터부문 책임연구원
1990년~현재 단국대학교 정보컴퓨터학부 교수
관심분야 : 컴퓨터 시스템, 컴퓨터 네트워크, 성능 분석