Incentives of Agent-Based Distributed Intrusion Detection Systems on the Open Internet

Ting-Chien Huang and Yuh-Jong Hu
Emerging Network Technology Lab.
Department of Computer Science
National Chengchi University, Taiwan
{g8913,jong}@cherry.cs.nccu.edu.tw
886-2-29393091#62066

Abstract—

We propose an agent-based distributed intrusion detection system (ADIDS) to enhance the conventional network based or host based IDS on scalability, flexibility, protection and trustworthiness. The agent communication language for digital certificates are encoded in XML format to ensure the most flexibility on the message encryption, decryption, authentication, and authorization. In this paper we demonstrate what are the possible incentives for ADIDS for intrusion detection when compared with regular IDS on the open Internet.

Keywords—

Distributed Intrusion Detection System, agent-based DIDS, XML encryption, XML signature, XML certificate.

1. INTRODUCTION

We had implemented a non-agent IDS to contrast with the agent-based distributed IDS (ADIDS) for the possible incentives on the open Internet. In the non-agent IDS, the offensive side launched some well-known attacks, such as nmap, dictionary attack, IIS bug, NetBus, synflood, teardrop etc, to validate the robustness of IDS. The generic IDS in this simulation is Bro [1] and the alert messages were encoded in IDMEF XML format to provide the maximum portability. After this simulation, we found out several problems for this non-agent information warfare. For example, we had a false alarm rate about 30.4% and intrusion detection rate about 80.4%. The reasonable causes for these side effects were shown as the followings:

1. System isn't closed and the unexpected attack occurs.
2. DoS attacks are durative attacks therefore the detection events were confused.
3. The factitious normal traffics and a large number of attacks cause the network blocked so that the alert can't be collected.
4. The services on the host and the IDS might be crashed so that we can't count the intrusion events.

Therefore we had an idea that the agent technology can be introduced in this IDS to solve the above problems. An ADIDS was set up to find out the possible incentives of agent technology in IDS and to support our idea.

2. RELATED WORKS

The distribution of IDS is not a new idea. There were some well-known IDSs, such as EMERALD, CSM and GrIDS to have this capability (see table 1) [8][9][10][11]. AAFID project is an agent based distributed IDS with multi-agent to monitor specific intrusion events [12]. But we highlight the cooperation and integration among different IDS. Centralized IDS (CIDS) monitors and coordinates a number of distributed Intrusion Detection Agent (IDA) using secure agent communication language on the open Internet. It is very important for the secure message passing between CIDS and IDA on the open Internet. But most of Distributed Intrusion Detection System (DIDS) does not have this communication mechanism. Using XML to encode the communication message for encryption, signature, and digital certificate that enhances the portability of these messages. And these XML well-formatted messages are suitable for heterogeneous IDS communication. Agent level communication is to reduce the workload of IDS. One of the incentives to use agent to detect the intrusion event is to minimize the

response time [7]. In case IDS is crashed at the run time, the responsible agent will reboot that IDS to guarantee the normal operations. Of course agent can react in time to avoid some malicious complex attack to break down the entire network.

Table 1: Distributed IDS

	network based	*host based*	*agent based*	*counterattack*	*security*
CSM		●		●	
EMERALD	●	●		●	
GrIDS	●	●			●
AAFID	●	●	●		
ADIDS	●	●	●	●	●

3. INTEGRATING IDS WITH MAS

In the experimental phase, we build our ADIDS system for inter-platform and run with RedHat and Windows operating systems on the Intranet. This ADIDS system will be migrated into Internet in the near future. The ADIDS framework is shown as the following Figure 1:

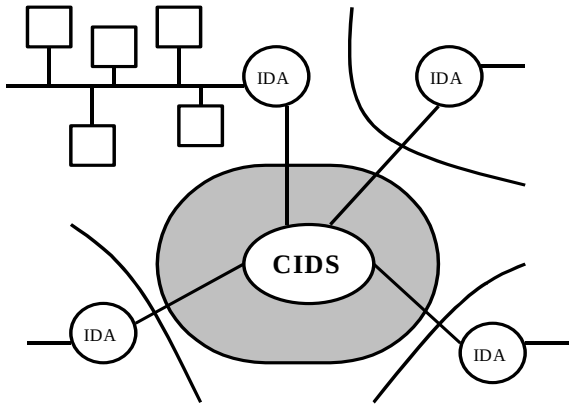

Figure1: ADIDS Framework

We rely on the multi-agent system (MAS) to enforce the communication among IDS. Intrusion detection policy is sent from CIDS to the specific IDA to renew or upgrade the outdated policy. IDA controls IDS that has the high trust on CIDS. The security of communication and authority control will be guaranteed by using XML well-formatted digital certificate. Thus the network security important criteria, such as confidentiality, authentication, integrity and non-repudiation are satisfied.

4. COMMUNICATION AND SECURITY MODULE

4.1 Communication mechanism

It is much simpler to use function call or system call among DIDS module than use agent communication language in our ADIDS. But the message syntax and semantics are limited in those system call communication. We design some agents to control different services to address the overloading with packets flooding and large-scale IDS architecture problem. Bro can only monitor some special services such as FTP or Telnet in the same network segment but our ADIDS distributes the overload services workload into multi-IDS among different network domains to construct the large-scale IDS architecture. In the ADIDS, multi-agent platform is based on FIPA-OS framework with IIOP as inter-platform communication RMI as the intra-platform communication [2][3][4]. We use FIPA standard communication acts, such as AGREE, INFORM, CONFIRM as agent outer communication language to pass control or alert message in the ADIDS.

4.2 Security module

We check the OS logs and patch the bugs to establish the agent authentication architecture that will be discussed later. There is not available for secure agent module in FIPA-OS to support our ADIDS to have a reliable IDS, so we developed from scratch. We integrated the J/Crypto cryptography module with RSA, DES, ECC, SHA, etc [5]. FIPA-OS supports RMI over SSL communication mechanism but that was only for the intra-platform communication. So we implemented our security mechanisms in the inner content level of agent communication language to have inter-platform secure communication. We used XML as inner content language and that contained XML encryption, XML signature, XML identity certificate and XML authorization certificate shown as the following (see table 2)[5]

$E_{A1 \to A2}$ = *Encrypted_data||Signature*
Encrypted_data = $E_{A1pub}[K]||E_K[ID_{A2}||$
$ID_{A1}||C||P||T]$
Signature = $E_{A2pri}[P]||H[E_{A2pri}[P]]$

Table 2: Attributes

Name	Description	Name	Description
A1	Agent1	A2	Agent2
ID_{A1}	A1's ID	ID_{A2}	A2's ID
$A1_{pub}$	A1's public key	$A2_{pri}$	A2's private key
K	Session key	C	Certificate
P	Plaintext	T	Timestamp
H	Hash	E	Encrypt

Content data is encoded with XML including encrypted data, signature and certificate.

```
<encrypted_key id="ek" carried_key_name="sk"
xmlns=…>
  <encryption_method algorithm=…/>
   <ds:key_info xmlns:ds=…>
    <ds:key_name>sk</ds:key_name>
     </ds:key_info>
     <nonce>x43gfh6gfjc</nonce>
    <time_stamp>3xydfg3bc</time_stamp>
<cipher_data>x2yd45gsfdg4c…</cipher_data>
 </encrypted_key>
</main_data>
</encrypted_data>
<signature id="ida" xmlns=…>
   ……
</signature>
<certificate id="ida" xmlns=…>
……
</certificate>
```

4.3 Identity and Authorization certificate

The authentication and authorization certificates are based on the agent-oriented PKI and encoded with XML. The following identity certificate was used for agent authentication [6].

1. *identity:* The identity of certificate owner that is recognized and registered with Directory Facilitator and Agent Management System.
2. *public_key:* The key of authentication claimant.
3. *option:* Optional information.
4. *validation:* Validation period of identity certificate.
5. *signature:* Certificate signature signed by the CA's private key

$IC_{CA \to IDA}$ = (ID_{IDA}, Pu_{IDA}, *Option, V,* Sig_{CA})

Authorization certificate has a field named validation and that causes the periodic authentication. In the initial stage, the authorization field only contains fewer arguments, such as monitoring privilege and system maintenance. Authorization certificate has five fields [6]:

1. *public_key:* The key of issuer to grant authorization.
2. *issuer:* Authorization grantor.
3. *authorization:* Expression of authorization.
4. *validation:* Validation period of authorization certificate.
5. *signature:* Certificate signature signed by grantor's private key

$AC_{CIDS \to IDA}$ = (Pu_{CIDS}, *Issuer, Authorization, V,* Sig_{CA})

5. INCENTIVES

In our ADIDS system implementation, we discovered some very important incentives of introducing agent technology for DIDS. These incentives are really important to enhance the robustness and effectiveness of DIDS shown as the followings:

5.1 Evaluation

The detection rate and false alarm rate are important statistic measures to evaluate the effectiveness of DIDS in the information warfare simulation model. The detection rate and false alarm rate can be defined as the following:

$$D = \frac{\textit{Formal detected events} - \textit{unexpected attacks}}{\textit{Total formal attack events}}$$

$$FA = \frac{\textit{Unmatched events} - \textit{unexpected attacks}}{\textit{Total detected events} - \textit{unexpected attacks}}$$

In our experiment, we found that many global attacks such as IIS Unicode attacks might disturb the intrusion detection events collected on the open Internet. The unexpected intrusion events outside our ADIDS framework will be counted

in our statistical measurement that results in the wrong detection rate and false alarm rate. We use IDA to filter out those unrelated intrusion attack events. Furthermore, IDA might normalize some of intrusion attacks in the measurement calculation shown as the following:

1. *Timing estimation:* DoS attacks or scan port attack usually take a long period to result in the influence so that we need to filter out the intrusion event in the same attack with those specially mixing up factitious normal traffics.
2. *Strategic attacks:* FTP bounces attack or FTP bugs always include sensitive field and data, but that is not obvious.
3. *Global attacks:* DDoS attacks are difficult to detect, but we can use an agent level policy to handle the special attacks.

5.2 Bootstrap

IDS should log the system operation events, and the system service thread should be run continuously. Thus, if the IDS crashed, IDA can discover this situation and handle the unexpected event by executing the system command script. CIDS also can boot the IDS from remote side by the following speech acts:

IDA → CIDS: request

```
<speech act>
 <precondition>IDS_CRASHED</precondition>
 <action>REQUEST</action>
 <content>
  <statement>REBOOT_ABNOMALLY</statement>
  <attachments>NONE</attachments>
 </content>
 <expect>CIDS_COMMAND</expect>
</speech act>
```

CIDS → IDA: agree

```
<speech act>
 <precondition>IDS_CRASHED</precondition>
 <action>AGREE</action>
 <content>
  <statement>REBOOT_ABNOMALLY</statement>
 <attachments>NONE</attachments>
 </content>


 <expect>IDS_OPERATE</expect>
</speech act>
```

IDA → CIDS:confirm

```
<speech act>
 <precondition>IDS_OPERATE</precondition>
 <action>CONFIRM</action>
 <content>
  <statement>IDS_OPERATED</statement>
  <attachments>IDS_STATUS</attachments>
 </content>
 <expect>NONE</expect>
</speech act>
```

The FIPA-OS MAS interoperates with the Bro IDS by executing the following system commands script:

in the Bro policy:

```
system(fmt("%s %s land_attack",
id$orig_h,id$act) );
```

in the FIPA-OS:

```
exec("sh /root/MAS/boot.bat");
```

5.3 Policy maintenance

CIDS can modify or update the intrusion detection policy at remote IDA dynamically.

CIDS → IDA: Inform

```
<speech act>
 <precondition>NONE</precondition>
 <action> INFORM </action>
 <content>
  <statement>UPGRADE_POLICY</statement>
  <attachments>POLICY_PACK1</attachments>
 </content>
 <expect>UPGRADE</expect>
</speech act>
```

IDA → CIDS: Confirm

```
<speech act>
 <precondition>POLICY_UPDATE</precondition>
 <action> COFIRM </action>
 <content>
  <statement>POLICY_UPDATED</statement>
  <attachments>POLICY_STATUS</attachments>
 </content>
 <expect>NONE</expect>
</speech act>
```

5.4 Pre-processing

The pre-processing of highly concentrated incoming alert messages for DoS attacks can avoid overloading the IDS. Thus the communication messages between IDA and CIDS will reduced rapidly too. Because DoS attacks always make tremendous number of intrusion alert events for a single attack action,

and that we should not warn this intrusion repeatedly.

5.5 Counterattack

Appropriate counterattacks for IDS are required to protect the network by using hosts.deny and hosts.allow operations to ban some attacking host for a moment. We have some drafts of the counterattack scenario as below:

```
if (DoS attacks)
      if (It is a repeated offender)
        Banned for a longer moment or forever
      else
        Banned for a moment
      End if
   end if
```

The methods are such as:

```
in.telnetd: somewhere.location.target
in.telnetd: someone@somewhere.target
in.ftpd: somewhere.location.target
```

And so does the followings:

1. Ban IP.
2. Block account.
3. Terminate program.
4. Terminate connection.
5. Resource restriction.

5.6 Communication

Heterogenous IDSs are not able to communicate with each other. But we can overcome this problem by using agent level communication shown in our ADIDS framework.

```
expression := <speech_act>
<speech_act> :=
<precondition><action><content><expect>
<content> := <statement><attachments>
```

5.7 Recovery

If services are crashed, IDA can restart the specific daemons or flush the blocked buffer by checking out if the events are related to specific attacks. Therefore, IDA will recover the services and countermeasure on attacks.

```
if (the specific services are crashed)
  if (attacks detecting)
    check and counterattack
    …
  end if
  recovery the services
end if
```

The methods are such as:

```
killall -HUP httpd
killall -HUP ftpd
```

5.8 Cooperation

We can integrate the IDS with firewall even with the system protected-purpose services to resist the attacks. IDA can communicate with IDS or execute the system commands to reconfigure the firewall dynamically.

6. IMPLEMENTATION

Our ADIDS is designed to integrate HIDS (Host based IDS) and NIDS (Network based IDS) with multi-agent platform running on top of them. The system architectures are shown as the followings:

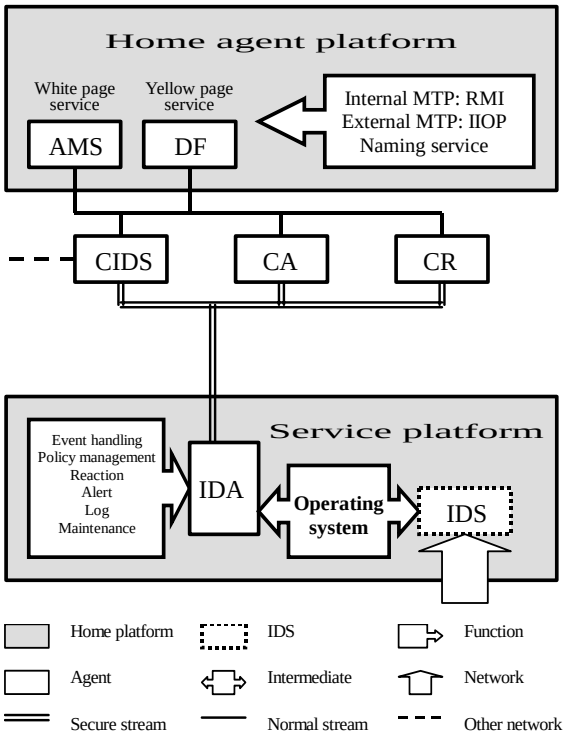

Figure2: ADIDS architecture

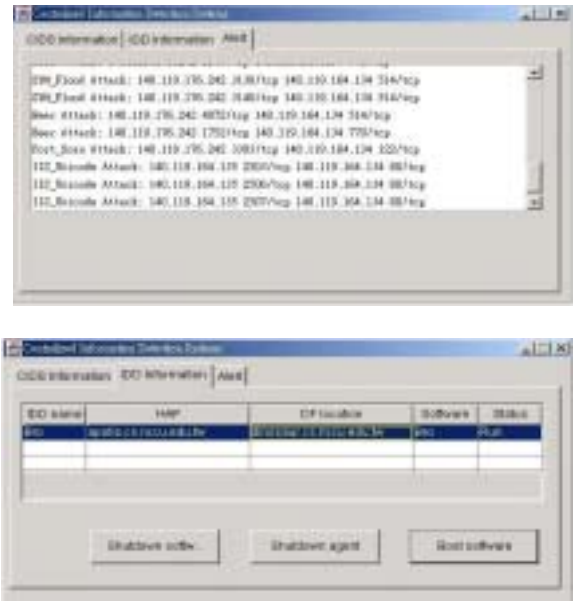

Figure3: System illustrations

7. FUTURE WORK

This ADIDS system is an initial prototype system so it still needs to be enhanced with the following features:

1. Need more emulation and testing to gather more statistical data for detection rate and false alarm rate
2. Agent functionality needs to be subdivided
3. Porting to more IDS systems
4. Developing more robust security module.

8. CONCLUSIONS

We first developed a non-agent based distributed IDS in our information warfare and find out some problems with this system. So we proposed an agent based distributed IDS (ADIDS) to resolve those problems and had discovered very important incentives by using multi-agent technology on DIDS. The problems faced by non-agent based distributed IDS are: evaluation, IDS management, counterattack, policy bootstrapping, communication, recovery, and cooperation. We also find out that the agent level communication provides the capacity to bind heterogeneous IDS systems and format a DIDS system. The integration mechanisms between multi-agent platform and IDS in this paper are based on system calls provided the operating systems. We envision that wrapper approach should be the solution in the near future.

Security is also an important issue in our ADIDS to ensure the reliable communication on the open Internet. Unfortunately existing multi-agent platform lacks of secure agent communication language to achieve our goal. Thus we implemented the secure agent communication features in the agent inner content language. The security criteria were indicated as identity certificate and authorization certificate with XML encoded format. The secure agent communication can protect our alert and control command message within ADIDS on the open Internet.

9. REFERENCE

[1] Paxson V., *Bro: A System for Detecting Network Intruders in Real-Time*, Computer Networks, 31(23-24), pp. 2435-2463, 14 Dec. 1999.

[2] FIPA-OS V2.1.0 Distribution Notes.

[3] FIPA Standards, http://www.fipa.org.

[4] Poslad S., Buckle P. and Hadingham R.G. (2000). *The FIPA-OS agent platform: Open Source for Open Standards*, Proceedings of PAAM 2000, Manchester, UK, pp. 355-368.

[5] Stallings W., *CRYPTOGRAPHY AND METWORK SECURITY: Principles and Practice*, Prentice Hall, 1999.

[6] Y. J. Hu, *Trusted Agent-Mediated E-Commerce Transaction Services via Digital Certificate Management, Journal of Electronic Commerce (JER), May, 2002, Kluwer Academic.*

[7] Curtis A. Carver, Jr., John M.D. Hill, John R. Surdu and Udo W. Pooch, *A Methodology for Using Intelligent Agents to provide Automated Intrusion Response*, Proceedings of the 2000 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 6-7 June, 2000

[8] J. S. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, *An architecture for intrusion detection using autonomous agents*, Tech. Rep. 98/05, Purdue University, 1998.

[9] Stefan Axelsson. *Intrusion detection systems: A taxonomy and survey*. Technical Report 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[10] P. A. Porras and P. G. Neumann, *EMERALD: Event monitoring enabling*

responses to anomalous live disturbances, In Proceedings of the 20th National Information Systems Security Conference, pages 353–365. National Institute of Standards and Technology, 1997.
[11] S. R. Snapp et al.. DIDS (Distributed Intrusion Detection System) - Motivation, Architecture and Early prototype. Proc., 14th National Computer Security Conf., Washington, D.C., Oct. 1991.
[12] AAFID http://www.cerias.purdue.edu/homes/aafid/