A Distributed Intrusion Alert System

Chih-Yao Lin, Hsiang-Ren Shih, and Yomin Hou

Taiwan National Computer Emergency Response Team[*]

{chinyao, shr, yominhou}@twncert.org.tw

Abstract

In this paper, a distributed intrusion alert system which is based on Honeypot technology is proposed. It is used to monitor unexpected actions appearing in different organizations. The motivation of this project comes from the hardness of detecting malicious activities without further assistance. The main advantage of this system is that it can monitor many IP addresses in different organizations at the same time to find unexpected actions. This system is named DIAS and has two parts. One of them consists of a number of Intrusion Alert Systems (IASs). Each Intrusion Alert System (IAS) is connected to the intranet of an organization to detect unexpected actions. The other part is Alerts Analyzing System (AAS) which is used for data collecting and analyzing. In this paper we not only discuss the system model but also the implementation of this system. The practical experiment shows the benefit of this system. The future works to improve this system are also discussed in this paper.

Key words: Honeypot, IAS, AAS, DIAS

1. Introduction

One of the missions of TWNCERT [1] is to coordinate among relevant agencies and organizations to identify pertinent response and actions in case of security incident. From the incident handling experience, most of the organizations have already had firewalls or devices for packet filtering in the perimeter, and anti-virus software on their computers. Some of them have also installed intrusion detection systems (IDS) or intrusion prevention systems (IPS) in the network. However, there are still intrusion incidents because of the incaution of employees, the new-found vulnerabilities of the operating systems or application software, new-type attacks, etc... Moreover, in the intrusion incidents, most intruders tried to further attack other computers in the intranet.

In order to solve the problem discussed above, we proposed a system named Distributed Intrusion Alert System (DIAS) which is based on Honeypot technology [2, 3, 4, 5, 6]. DIAS consists of two parts. One of them consists of a number of Intrusion Alert Systems (IASs), and the

[*] Taiwan National Computer Emergency Response Team (TWNCERT) is supported by Institute for Information Industry, *http://www.iii.org.tw*.

other one is Alerts Analyzing System (AAS). The main idea of IAS is to occupy some or all of unused IP addresses in the organization's intranet. It is designed to be placed in an organization's intranet but not to interact with other hosts, so that any connections initialized by other hosts to those IP addresses are naturally unexpected and probably generated by an intruder. It is a simple idea and can be implemented easily. Moreover, the role of AAS is a platform to collect data from different organizations for further analyzing.

During the practical experiment, some of the alerts were identified to be worms, firewall configuration mistakes, and intrusions. The result shows the benefit of the system. It is of practical use for detecting intrusion with low cost.

The rest of the paper is organized as follows. Section 2 describes the concept used to build our DIAS architecture. Section 3 presents the proposed DIAS framework, details of its building, its advantages and its limitations. Section 4 details a proof-of-concept experiment utilizing a DIAS implementation. Finally, we provide the conclusion in Section 5.

2. System Model

In this section, we discuss the system model of DIAS architecture. To take a quick view of DIAS, the requirements of it are listed below.

1. The system should be able to detect unexpected activities in the network.
2. The system must not become a stepping stone used by intruders to attack others.
3. The system should be able to collect and analyze the unexpected activities from different organizations, and provide early warnings of attacks

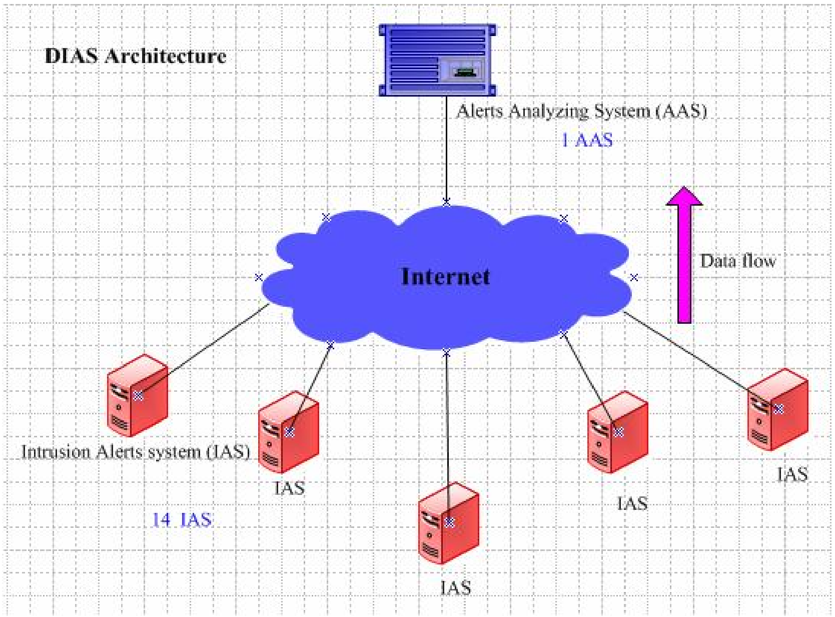


Figure 1. The architecture of DIAS

To achieve the requirements mentioned above, the architecture of DIAS shown in Figure 1 has been designed. There are two parts of DIAS. One of them consists of a number of IASs, and the other one is AAS. The first two requirements are carefully considered when designing IAS and the last requirement is accomplished by AAS. Each IAS is expected to be installed in an organization in order to detect unexpected connections initialized by intruders. AAS is used as a platform to collect alerts from different organizations for further analyzing. Back to Figure 1, the arrow sign from the bottom toward the top indicates data transferring from each IAS to AAS. All IASs are unable to communicate with each other since they may belong to different organizations. AAS is the only place to collect and analyze all data from IASs.

2.1. Architecture of IAS

As mentioned in the first two requirements, DIAS not only has to detect intrusions but also prevent itself from being a stepping stone at the same time. Those requirements are seriously considered while implementing IAS. For those considerations, there are several components designed for each IAS. As shown in Figure 2, there are four components of IAS that are used to achieve those goals. Those are “Security Protection Component”, “Virtual Systems Responding Component”, “Packets Capturing Component” and “Data Transferring Component”. Those components are listed as the flow of data; all packets are supposed to move according to the order. The following is the individual discussions of those components.

“Security Protection Component” is the component of IAS used to control which packet can pass into the IAS or not. It is the key point to prevent IAS from being a stepping stone as discussed in the second requirement. This component adopts a host based firewall to protect each IAS by blocking any connections to the IAS itself. That is, there is no way that attackers can connect to IAS

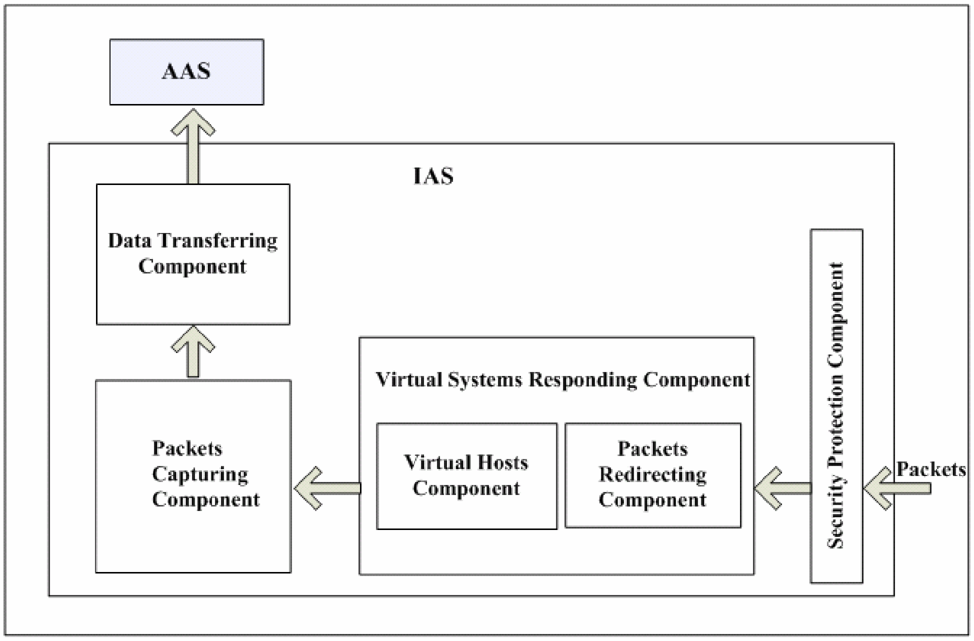

AAS
IAS
Data Transferring Component
Virtual Systems Responding Component
Packets Capturing Component
Virtual Hosts Component
Packets Redirecting Component
Security Protection Component
Packets

Figure 2. The system architecture of IAS components

itself. If attackers want to connect the IAS itself in order to compromise it, it will be rejected by the firewall. Packets going to “Virtual Systems Responding Component” of IAS are the only way to pass through the component. There are only virtual systems provided by “Virtual Systems Responding Component”.

“Virtual System Responding Component” on IAS is used to provide several virtual systems which are used to interact with intruders. For intruders, it is not allowed to connect to the IAS itself due to “Security Protection Component” except those virtual systems. This component is designed to satisfy the first requirement of being able to detect unexpected activities on the network. There are two subcomponents named “Packets Redirecting Component” and “Virtual Hosts Component”. The former is used to redirect packets to the virtual systems on IAS and the latter is used to interact with intruders. The reason of using the first subcomponent is as follows. When some hosts are in the same broadcast domain, if one host wants to send packets to another one with a different IP address, it needs the MAC address of that host. In order to get that, it will first check its own current ARP entry to see whether it has that host’s MAC address. If there is no the record in the ARP entry, it will send ARP requests to the broadcast MAC address in order to get an ARP reply from that host. After having the MAC address of that host, the host can then send packets to that host. Because the combination of the IP address and the MAC address might change, the ARP entry of one host is dynamic by default in order to suit the current situation. That is, ARP requests and ARP replies would appear in the local LAN quite often. In order to be able to detect unexpected actions, this subcomponent is configured to redirect packets by replying ARP requests to specific unused IP addresses which we want to monitor. For example, if we have installed an IAS with 192.168.2.2 as its real IP address and we want to monitor packets going to an unused IP address which is 192.168.2.3, we use this subcomponent to reply the ARP request to this IP address. This would make that host which sends the ARP request in order to get the ARP reply of 192.168.2.3 think the host is alive but it is one virtual system of IASs actually. Similarly, the reason of using the second subcomponent is to make intruders think the virtual host is real by using Honeypot technology. The usage of Honeypot system is to simulate product servers to deceive attackers but in fact they are not. Since the first requirement is to monitor unexpected actions in the network not only a host. This subcomponent is designed to be able to simulate many hosts at the same time. This is the key point to detect unexpected activities more possibly. For example, one can be configured to monitor up to hundred of IP addresses at the same time with all virtual hosts defined to several types of operating systems. Since many threats happen on Windows systems, all virtual hosts of each IAS are configured to act as Windows 2000 systems now.

“Packets Capturing Component” is used to capture packets going to those virtual systems on IAS and IAS itself. In order to collect packets coming to or leaving from the IASs, packets sniffing tools are used for this purpose. This component is used to satisfy the first requirement too. When it is used with “Virtual System Responding Component”, IAS is able to detect packets attracted by it and record those packets for further analyzing. Those packets are naturally suspect since all virtual systems on IAS is not exist before using IAS. In this component, all packets captured by one interface of one day on each IAS are stored in one file. If there are more than one interfaces, more than one file are generated. Files are then sent to AAS via “Data Transferring Component”.

“Data Transferring Component” takes the responsibility to transfer captured packets by IASs to AAS every midnight automatically. The reason of using this component is to help satisfy the last requirement. That is, after collecting packets, further analyzing of them and early warning of attackers derived from them might be possible.

In sum, only packets going to virtual systems on IAS are accepted and those packets will be handled by “Virtual Systems Responding Component” which includes two subcomponents, “Packets Redirecting Component” and “Virtual Hosts Component”. The former subcomponent is used to reply with ARP replies for those virtual hosts in order to make other hosts think those virtual hosts are real and sent packets to them. The latter subcomponent is used to create some virtual systems on IAS for attackers to interact with. The number of virtual hosts is changeable in order to suit different situation. Then, “Packets Capturing Component” is done by using packets sniffing tools to capture packets appearing in the previous component. Finally, “Data Transferring Component” is used to transfer captured packets in each IAS to AAS on a daily basis in order to collect, analyze those packets. This is used to comply with the last requirement.

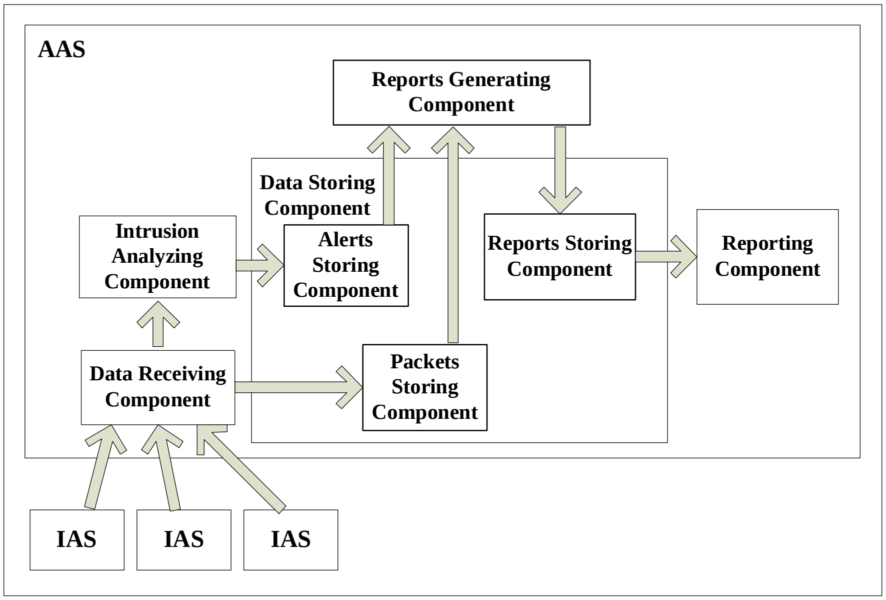
2.2. Architecture of AAS

Beside the first requirements, the third requirement of DIAS is mostly related to AAS. AAS is used to store data coming from each IAS, analyze packets and show the result on a Web interface. Its architecture is shown in Figure 3. There are several components on AAS. They are “Data Receiving Component”, “Intrusion Analyzing Component”, “Data Storing Component”, “Reports Generating Component” and “Reporting Component”. The following is their individual discussions.

“Data Receiving Component” is used to receive to data come from all IASs. Since all data recorded on each IAS is stored in files, this component just keep those files in the related directories on AAS in their original format. Further processes are done in other components and this component only receives and stores valuable data day by day.

“Intrusion Analyzing Component” is used to generate intrusion alerts. One easy way to verify if there are malicious activities among some packets is by using IDS with its patterns to check if there are any problems. As the last requirement mentioned before, the purpose of this component is to analyze packets stored by “Data Receiving Component” in order to find intrusion events. This component is launched daily since all new data is come to AAS once a day. After having any intrusion alerts, they are sent to “Data Storing Component”.

“Data Storing Component” is used to store data. This component mainly uses database packets and designed programs to insert data to the database and query data from the database. There are three subcomponents of it. Those are “Alerts Storing Component”, “Packets Storing Component” and “Reports Storing Component”. The first subcomponent is used to store alerts generated by



AAS
Reports Generating Component
Data Storing Component
Intrusion Analyzing Component
Alerts Storing Component
Reports Storing Component
Reporting Component
Data Receiving Component
Packets Storing Component
IAS
IAS
IAS

Figure 3. The system architecture of AAS components

"Intrusion Analyzing Component". The second subcomponent is used to store packets information extracted directly from files received by "Data Receiving Component". This subcomponent has its responsibility to insert data extracted from those files to the database flawlessly. The reason of using this subcomponent is because there is not enough of only having alerts stored in the first subcomponent. Although alerts generated by IDS are useful, it would be better to have information about connections too. That is, if we have this information in addition to related alerts, we would have a better view about real actions. Those two subcomponents then wait "Reports Generating Component" to query data stored in them. The last subcomponent of this component is used to store final reports generated from "Reports Generating Component". Those subcomponents are all useful to satisfy the last requirement of DIAS.

"Reports Generating Component" is used to query packets and alerts stored in "Data Storing Component". Because false positives might happen before this stage, it needs manual intervention to analyze them and find the reasonable explanations. Engineers have to see and verify the data stored in "Data Storing Component" and insert the final reports to "Data Storing Component" again.

"Reporting Component" is used for engineers to query reports stored in "Data Storing Component" and produces reports. Those reports are then sent to system administrators of organizations for them to verify if the information is harmful or harmless. Since false positives might still appear in the final reports, manual intervention of system administrators of each organization is also necessary.

In sum, when data come from IASs via their "Data Transferring Component", it is received by "Data Receiving Component" and stored on AAS for further handling. There are two ways to process the data. One is to produce alerts by using “Intrusion Analyzing Component” and pass them to “Data Storing Component”. Another way is to store packets information in “Data Storing Component”. To be more specific, the subcomponent named “Alerts Storing Component” is used to store alerts come from “Intrusion Analyzing Component”, and another subcomponent named “Packets Storing Component” is used to store packets information (source IPs, destination IPs, source ports, destination ports, protocol, etc...). After that, “Reports Generating Component” of AAS is used to combine alerts and packets information together and provide an interface for AAS users to verify if the data is reasonable. Reports generated are then stored in the subcomponent named “Reports Storing Component” of “Data Storing Component”. Finally, when managers of AAS need to see the reports, “Reporting Component” organizes them and produces reports. Those reports are then sent to related system administrators.

3. Implementation

In this section, we show how we implement DIAS by mentioning packages that we use. First, the implementation of IAS is discussed. Then, the way that we build AAS is illustrated. Finally, the advantages and limitations of our implementation are listed.

3.1. The Building of IAS

In this section, all IAS components and packets used by them are as follows.

“Security Protection Component” uses *iptables* [7] as the host firewall to protect each IAS. The only way to pass through this component is to go to the virtual systems running on IAS.

“Virtual System Responding Component” on the IAS is used to provide virtual systems which are free for attackers to interact with. The first subcomponent named “Packets Redirecting Component” is accomplished by using *Arpd*. The second subcomponent named “Virtual Hosts Component” is accomplished by a low interactive Honeypot system called *Honeyd* [8] which is open source software released under GPL [9]. One of its good features is the ability to simulate many hosts at the same time. We use this feature to monitor unexpected actions happen on more than one host. Because all virtual hosts provided by *Honeyd* are not real, it is very difficult for attackers to compromise the real operation system via *Honeyd* especially when it is not run with the root privilege. That is, the IAS is quite secure by default.

“Packets Capturing Component” is accomplished by using a package called *Snort* [10]. The way to store data is as the file type which is *Libpcap* [11] format.

“Data Transferring Component” is done by SCP. It is designed that every data is transferred from IAS to AAS without system administrators’ assistance. It can be used to transfer data automatically from SCP client to SCP server if SCP server has SCP client’s public key in place. After this, all further SCP connections will be successfully created without inputting the password personally. Cron table on IAS is used to transfer IAS logs to AAS via SCP periodically.

3.2. The building and operation of AAS

In this section, all AAS components and packets used by them are as follows.

"Data Receiving Component" is used to receive to data come from IASs. Since each IAS uses SCP client to transfer data, SCP server is used here to get the data.

"Intrusion Analyzing Component" is used to generate intrusion alerts. *Snort* is used as IDS in to interpret files received in "Data Receiving Component" and generate security alerts according to its intrusion detection patterns.

"Data Storing Component" is used to store data. This component mainly uses *MySQL* [12] as the database and *PHP* [13] as the language to insert data to the database and query data from the database. Since the original files are *Lipcap* format, *PHP* needs *Phpcap* [14] to know this format.

"Reports Generating Component" is used to query packets and alerts stored in "Data Storing Component". Web interface provided by *Apache* [15] and *PHP* is used to provide GUI for TWNCERT engineers to see and verify the data stored in "Data Storing Component" and insert the final reports to "Data Storing Component" again.

"Reporting Component" is used to query reports stored in "Data Storing Component". *Apache* and *PHP* are used again to produce HTML format reports.

3.3. Advantages and Limitations

From the implementations of DIAS, we can directly deduce its advantages and limitations as described below.

Advantages:

1. It is quite secure because all services on each virtual host hosted by *Honeyd* are not real, and all ports of IAS itself are blocked by *iptables*. There is no way for attackers to connect to the IAS itself expect the virtual hosts on it.
2. It is very possible to find unusual actions launched by attackers or mis-configured hosts since it is expected to see no packets at all on each IAS. In addition, by using *Honeyd* and *Arpd*, it is able to watch many IP addresses at the same time.
3. It is possible for DIAS to find an overall view of activities from all IASs. This might be useful in sharing trend information to different organizations.

Limitations:

1. There is no way to find any intrusions when there are no packets going to IASs. The common limitation of Honeypot system happens on DIAS too. That is, if there are no actions launched against a Honeypot system, it is impossible for a Honeypot system to find those actions.
2. When it detects unexpected activities, there is no chance for it to get full processes launched by attackers from the beginning to the end of an attack. Because there are no real services provided by IASs, it is also a limitation for not being able to find a full attack process. Since the services used to deceive attackers are not real, there are not many things they can do with them.

Therefore, there are fewer things can be recorded and learned.

4. Experiment Result

In this section, we show the experiment result of DIAS. First, the expected result is discussed. Then, we show the particular experiment results derived from different IASs and all experiment result in an entire view from DIAS. All experiments are followed by related discussions.

DIAS has been built since April 2004. In this part, data come from each IAS before the end of year 2004 is used to discuss. Although it is believed that each result would comply with the expected result, unexpected results still happened. The reasons of those situations are interesting and deserve more explanations.

Before showing the experiment results, here are some items need to be clarified. First, every report on AAS is stored on a daily basis. That is, with data being automatically managed by AAS and manual analyzing, every report standing for a report of one day is sent to system administrators of the organizations. Second, there are two kinds of diagrams for explaining the two expected results shown in the previous part. The first one shows the result of each IAS. Although there are 14 IASs and there should be 14 diagrams by default, it might not be useful to spend time on many similar results. After verifying all diagrams, four of them are chosen. They are four typical results need to be discussed thoroughly. In addition, the second kind of diagram shows an overall view of DIAS.

4.1. Four typical diagrams from all diagrams

As we can see in Figure 4, the X-axis stands for the time from June to December, and the Y-axis stands for the number of IP addresses which have ever seen in the packets going to IAS. In this diagram, IAS encountered many hosts which sent packets to it. According to the figure, there is no record between August to the end of October. The reason was that IAS faced a hardware fault during that time. After replacing a new machine, it also faced an unusual network where many

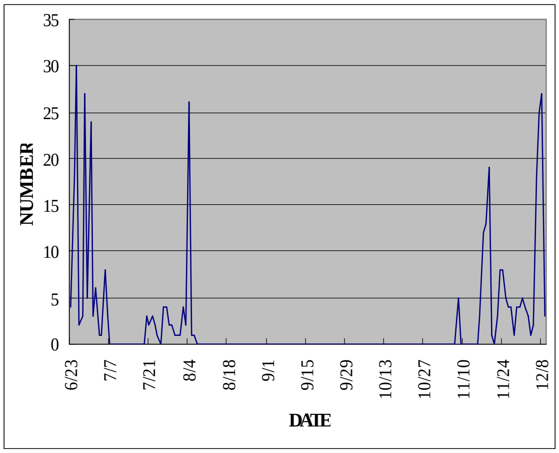


Figure 4. IAS result #1

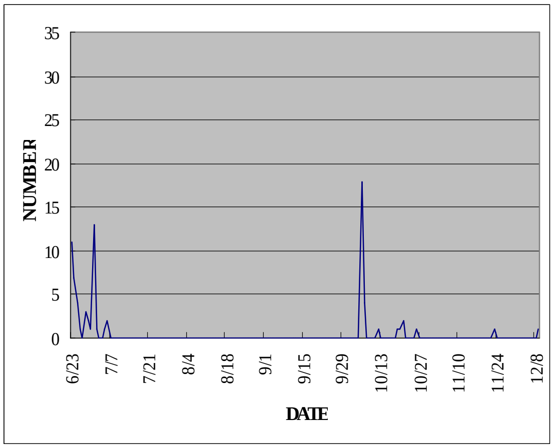


Figure 5. IAS result #2

uninvited hosts on the network sent packets to it. For a deeper discussion, it is required to talk about why those uninvited hosts want to interact with it. Unfortunately, the reason is hard to explain due to the nature of IAS. As the limitation shown above, it is hard to tell what each packet stands for. In addition, after discussing with the system administrators of this organization, some of those hosts on record can be figured out, while some of them are still hard to explain.

From this case, something could be learned. First, DIAS complies with the limitation discussed above. Second, the success of DIAS mostly depends on the cooperation of system administrators of other organization. It is hard for TWNCERT engineers to find a truth happens on IAS without the system administrators' help. Finally, DIAS does help system administrators of the organization to find unexpected activities and have a better view about their environment. That would be helpful for them to find potential problems.

In Figure 5, it is a typical example of mis-configured setting. In the beginning, although IAS is claimed to be placed inside their perimeter firewall, the firewall is not configured correctly. This means packets from the Internet can reach IAS without any difficulty. After discussing with the system administrators, this situation finally changed at the beginning of July. The flat situation had lasted for three months before the firewall setting was set wrong again at the beginning of October. As it is shown in this figure, except the time of mis-configured setting of the firewall, there are few records can be seen. On the other hand, IAS in this case is useful to find mis-configured firewall setting because of the dramatic change in the figure.

In Figure 6, an example of particular hosts which are defined to scan every host on the network for a particular purpose and some unknown hosts send packets to IAS is shown. Since there is a particular host placed on the network, there are many records with number just equals to 1. Beside this, some uninvited hosts have ever sent packets to IAS. Some of them are because of some users like to find their neighbors, or worms' spreading.

In Figure 7, a typical example of no particular hosts and no unknown hosts' activities is shown. Except of some curious users want to find other hosts, there is no packets sent to IAS. The rest of diagrams which are not shown have the same characteristic of this figure. That is, there is no record

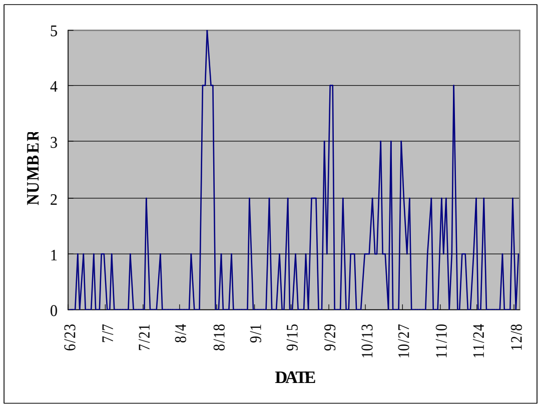


Figure 6. IAS result #3

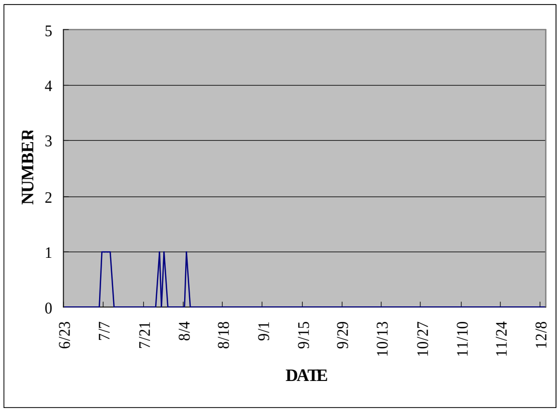


Figure 7. IAS result #4

most of the time. If there are records, the amounts are all quite low.

After seeing those diagrams, some similar key points suit all organizations are discussed here.

- DIAS is proved to be able to detect unexpected actions or is useful to find worms' activities, though; the reasons of the actions should be verified by system administrators of each organization. It is not always easy to verify them.
- DIAS has encountered IP addresses owned by some IASs are used by other hosts several times, it is reported that those situations are due to accidentally IP setting by some users.
- It is believed that it is hard to find attacks which are not meant for scanning all hosts on the network. It needs added features for attracting attackers.

4.2. Entire view of DIAS records

In this part, packets collected by all IASs are added together. The data is shown in Table 1.

Table 1. The top 10 TCP and UDP ports of DIAS

Order	Protocol	Port	Number	Protocol	Port	Number
1	**TCP**	**139**	24740	**UDP**	**137**	2785
2		**445**	17898		**161**	1081
3		**2745**	2832		**138**	116
4		**3127**	2036		**2967**	109
5		**1080**	1786		**53**	91
6		**80**	1650		**1026**	76
7		**3128**	1537		**1027**	61
8		**1025**	1324		**38293**	55
9		**135**	1267		**111**	55
10		**6129**	1065		**1434**	48

In Table 1, "Order" means the order from 1 to 10 according to the number; "Protocol" means the port next to it is belonged to TCP or UDP; "Port" shows the port number which is one port of IAS's virtual system; "Number" shows the number of each related port. As it is shown in Table 1, TCP port 139 and 445 have larger number than others and UDP port 137 and 161 are the first one and second one of all UDP ports. According to this situation, the possible reasons are as follows.

- There are many Windows machines next to IASs and some of them are infected by worms and start to infect other machines via TCP port 139 or 445.
- The system administrators want to watch if all Windows machines are on line. They install some software or place some hardware devices which are configured to send packets to TCP port 139, 445 and UDP port 137 of other machines.
- There are network managing devices which send SNMP packets to UDP port 161 of all machines for monitoring and managing the network.

5. Conclusion

In this paper, DIAS is used by TWNCERT to help some organizations in Taiwan. It is proved to have good effects in those organizations. For example, it is able to detect unexpected actions on more than one service or host. The information produced by it is shared to the system administrators of the organization for them to find problems and solve them. Although the system is currently unable to confirm happening problems, it is actually the system administrators' responsibility to deal with them. In addition, it is harder and harder to catch sophisticated attacks since attackers are unwilling to scan every host on the network. It is really a challenge. The future research of DIAS is to provide a more real not still secure system and a smarter system which the alert threshold limit can be set in order to reduce the amount of alerts for example. Finally, the more important thing realized from this research is the factor of human. The cooperation between people involve in the framework is really important to find problems and solve them for a more secure environment. On the other hand, if IAS could be distributed to more organizations, it would be better for finding potential problems on the whole environment in Taiwan. One way to promote IAS more easily is to provide an easier installation method. Currently, RedHat Anaconda package has been successfully used to build a customized bootable Linux CD which includes all necessary packages and some shell scripts used to provide an easy installation and configuration interface. It is believed that it would help distribute this system to as many organizations as possible and get the benefits of it.

References

[1] Taiwan National Computer Emergency Response Team (TWNCERT). http://www.twncert.org.tw/en/index.php

[2] B. Cheswick. An evening with Berferd in which a cracker is lured, endured, and studied. *In Proceedings of USENIX Winter 92 Conference*, January 1992.

[3] L. Spitzner. *Honeypots: Tracking Hackers.* Addison-Wesley, September 2002.

[4] L. Spitzner. The honeynet project: Trapping the hackers. *IEEE Security and Privacy Magazine.* March/April 2003.

[5] The honeynet project. Know Your Enemy : Learning about Security Threats (2nd Edition). Addison-Wesley, May 2004.

[6] The honeynet project. Whitepapers. http://www.honeynet.org/papers/ index.html

[7] The netfilter/iptables project. IPtables. http://www.iptables.org/

[8] N. Provos. Honeyd. http://www.honeyd.org/

[9] The GNU Project. GNU General Public License. http://www.gnu.org/copyleft/gpl.html

[10] M. Roesch. Snort. http://www.snort.org/

[11] TCPDUMP/Libpcap project. Libpcap. http://www.tcpdump.org/

[12] MySQL AB. MySQL. http://www.mysql.com/

[13] The PHP group. PHP. http://www.php.net/

[14] R. Julien. Phpcap. http://alcane.newffr.com/phpcap/

[15] The Apache Software Foundation. Apache. http://www.apache.org/