

# Intelligent Distributed Intrusion Detection Systems

by: Rajesh T Sivanandan, 10/14/2004

<http://www.securitydocs.com/library/2641>

## Abstract

An Intrusion Detection System always helps the “second in the queue”, in other words; any Detection System can only say that there is an attack which is ongoing. Now based on the facts delivered from them, the second attempt (or may be the 100th attempt) may be neutralized. Now what is there that is missing in these algorithms? Answer is simple, Intelligence! Gone are the days where a computer is thought to be a piece of toy which will just do what you ask it to do. Time demands more than that, and so is the topic of bringing *intelligence* to Intrusion Detection Systems than to go with traditional Detection Systems.

A system that has to be built which diligently closes a door to an unknown person carrying a gun. We will try to see some of the possible and explicable areas of building Intelligence to Intrusion Detection Mechanism.

## Introduction

This paper is an attempt to concentrate more on these techniques and to assume that we aren't aware of any Intrusion blocking mechanism yet. Well, that helps towards the orientation of thinking unprejudiced. Questions to be asked and answered are;

1. How a Network IPS would ensure the health of Complete Network, sitting at one place and doesn't know what its neighbor is upto?
2. How would a Host IPS ensure the health of the Host alone while the host is talking to numerous known and unknown peers?
3. How can we get rid of annoying intrusions ahead of time unless you know them?
4. How can we eliminate cumbersome job of Forensic Analysis, which takes hours and hours of time just to find that all the time spent was in vein since those are False Positives?

## Need for Hierarchically distributed Intrusion Detection

### Problems

Most networks today employ Intrusion Detection Systems which sits in the network and passively monitor the traffic. If there are 3 sensors (Sensor A, Sensor B and Sensor C] monitoring a subnet 10.0.0.0, then any update to the knowledge base of the sensors is 3 fold. Take an example of Signature based IDS, if there is a new signature update then the owner of the network needs to do the signature update 3 times so that all the sensors gets updated. Even if there is a mechanism to update them together in a multithreaded fashion, still you wouldn't prefer it. The reason is that during signature update if the sensor A needs to reboot to rebuild itself, you want Sensor B to be active at that time and monitor the network. Even for a fraction of the second you don't want to leave your network unattended (Ahh... Thanks to the hacking Community). But still the organizational policy would say something like “all the latest patches and updates needs to be updated within 24 hours of its availability”

Now it is a clear factor that it doesn't happen all that well! If at all the Sensors were intelligent enough

to talk to each other and get to know the happenings, then it could have been managed in a better manner. So let's look at "Sensor to Sensor Communication"

## **Sensor to Sensor Communication**

If we have sensor to sensor communication then the Administrator doesn't have to worry about updating all the sensors in the network and also the timings/down times. All he needs to do is to delegate the work to one of the sensor and all other sensors would follow.

Also if we had intelligent communications between sensors, they would be diligently adjusting their sensing capabilities to suit Network Dynamics. All the sensors don't need to be running the same set of instructions at the same time. From now we will concentrate on Signature based Intrusion Detection Systems and try to conceive some basics on how to make it an intelligent Signature based Intrusion Detection Systems and also Intrusion Detection Systems.

Any Intrusion Detection Engine essentially consists of three parts;

- Application Layer Interface
- Middleware Interface
- Machine Level Interface

### **Application Layer Interface**

This is the interface which is used to configure the sensing options and other Network Layer parameters. Mostly GUI/HTML application.

### **Middleware Interface**

This is the interface where all the objects communicate, mostly the software components that make Intrusion Detection possible.

### **Machine Layer Interface**

This is the one which actually makes the system work to the bit level instruction-processing.

We will concentrate more on the improvements that can be brought into the Middleware Component where the distributed hierarchy can be fixed in. Different IDS should communicate to themselves and so it is more convenient to leave the Network Layer Communications to a Middleware framework which will take care of it. So the only object that needs improvement would be for the IDS-Message conversation.

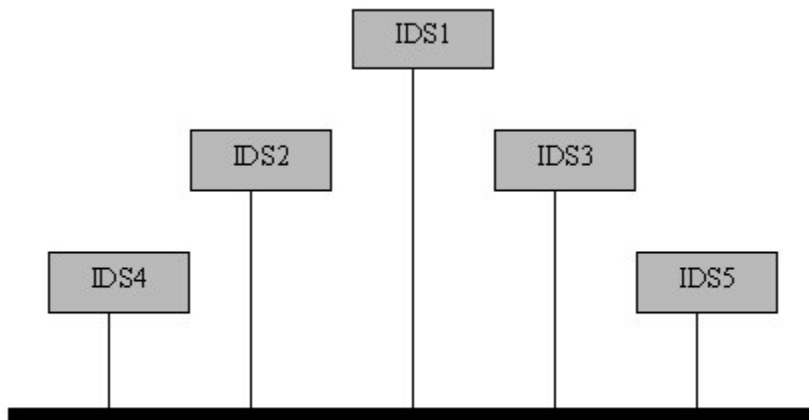
## **Message Oriented Middleware**

The concept here is about an infrastructure which will basically provide a plug-in for all the parties involved for middleware level communications. What will be the inbuilt services taken care by the MOM? We'll see them in a moment.

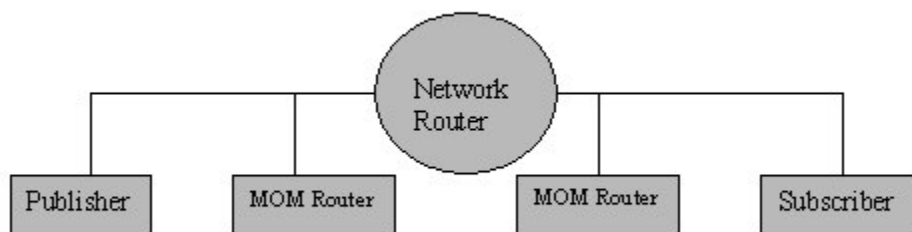
- Network Communications (End to End):
  - This can be further classified into different modes of communications as "One-to-One", "One-to-Many", "Many-to-Many"
  - Support for "Unicast", "Multicast" and "Broadcast" communications.
- Basic Client-Server architecture:

- For easy and structured communications between different ends.
- Message Oriented Communication:
  - This is the key word for communication between different ends. The advantage is that there is no IP Address involvement, No Subscription initiation is involved. It all happens in an “as-we-go” approach.

So we now will have to integrate this MOM to the IDS Middleware. Looking at the picture below;



There are 5 IDS boxes in the diagram connected to a Network Bus. It can all be in the same segment or in different network segments based on the size of the network. All the IDS boxes have a middleware which is having an integrated MOM. The software module of a MOM can act as both “Publisher” and a “Subscriber” based on the configurations. If there are different Network Segments, then MOM module can also act as message-routers to other subnets. It will basically be something like below;



**Publisher:** A publisher will publish data in the form of “Message” to the network.

**Subscriber:** A Subscriber will get registered to the “Message” and will receive any messages that are being delivered by the “Publisher”.

In the above picture, lets take “IDS1” be the publisher and all others are subscribers. For a better understanding of MOM, let us see how this works. First IDS1 creates 2 messages which are

“SigUpdate” and “PolicyChange”. All the subscribers should be configured to subscribe to these messages. All the MOM modules are configured for “Multicast” mode of communication. That’s it! The configuration part is done. Now whenever there is a policy change or signature update, IDS1 will propagate appropriate message using the message-names defined previously and other IDS boxes would just follow as instructed.

### **Signature Update in Distributed IDS Environment**

Let’s take an example; new signature update information is being delivered to IDS1. First IDS1 determines the information on where to download the update from then downloads it and installs it. It then sends out a message which will contain the “Heading”, “Availability” and a “Time Stamp” using the previously configured message-name (SigUpdate).

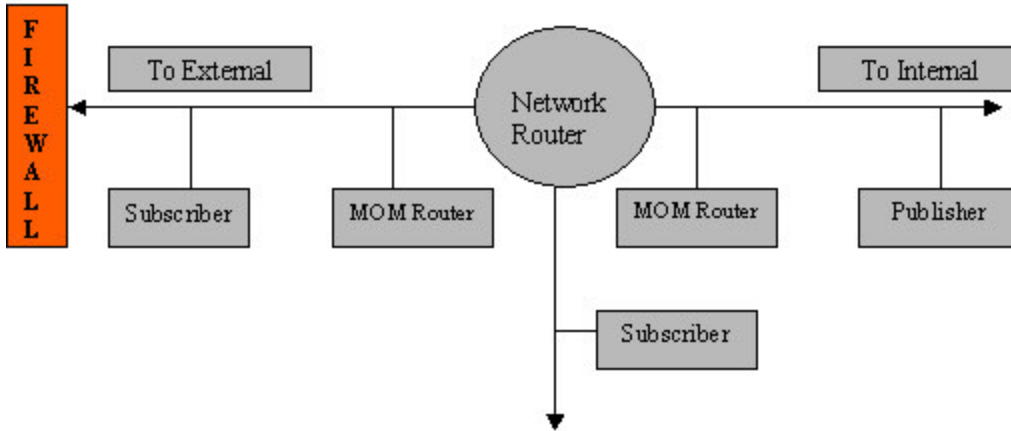
- **Heading:** This should contain the Signature Update Package Name.
- **Availability:** This should contain from where it can be downloaded from.
- **TimeStamp:** This will contain the timestamp when IDS1 is producing this message (After installing the update to itself first)

Each subscriber (who is subscribed to this message-name) receives this message and will have arbitrary back off time before installing the update. This is to make sure that no two sensors are trying to install the update at the same time. At any time all the sensors should have taken action before a “reasonable” time and send its status back to IDS1 or its Management Center in the form of “Status” messages. The Management Centers/Monitor Centers will update the local database to reflect the updated information about each sensor.

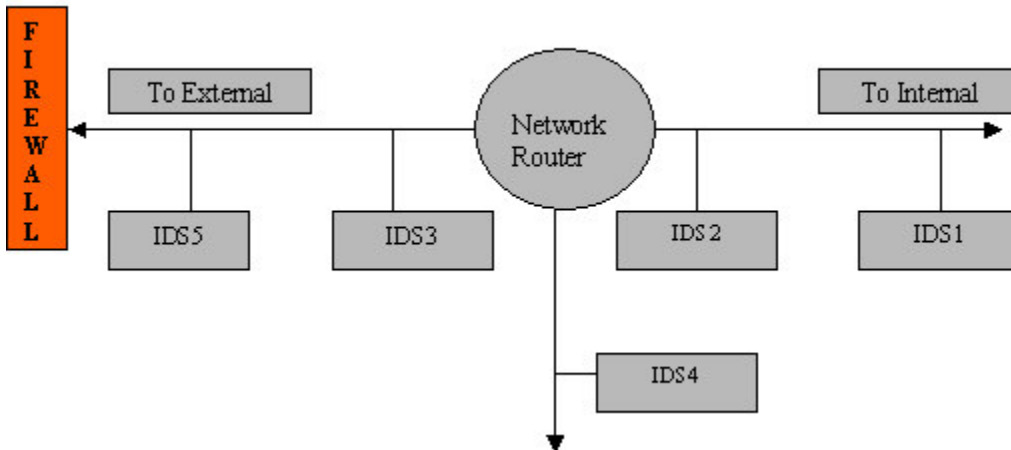
### **Policy Update in Distributed IDS Environment**

Referring to the diagrams below; there is a Publisher IDS in the internal network behind the Network Router. There are 5 subscriber IDS including the two MOM Routers which also do the message-routing along with acting as subscriber IDS. Message is transferred from Publisher to the local network using multicast. The MOM Router on the internal network passes this information to the MOM Router on the external network in a Unicast transfer. The MOM Router at the external network now is acting as a Subscriber + MOM Router + Publisher (for the external segment). So the whole message is traversed across all the participating Intrusion Detection Engines. IDS4 monitors the network segment where Web-Server and Mail-Server resides. The density of Intrusion Detection Systems in the network may look impractical. This is only a pictorial representation and efforts are put here so as to avoid confusions while defining roles for individual IDS roles. So again, in a real network scenario, there won’t be as many IDS as below;

MOM Logical Diagram:



Distributed IDS Logical Diagram (Corresponding to the above diagram)



Looking at the above sketch, we can divide the whole network into 3 segments. One that is closer to the external network. One is the internal network and the last one being the DMZ where web servers, SMTP servers reside. The external network is being monitored by IDS5 and IDS3. Internal is taken care by IDS1 and IDS2. The DMZ is taken care by IDS4.

Consider all of them are Signature based IDS systems and they have a total of  $N$  signatures and average time taken to go through one signature is say  $T$  seconds. If I enable all the signatures on IDS5, then if a packet has to be analyzed for all the signatures, it would take  $N * T$  seconds. So during this time all the other data packets will be kept in a queue to be analyzed. If the queue buffer is say  $Q$  then when a packet which is  $Q+1$  comes, that packet will not be analyzed at all. This is more applicable in that segment because your traffic is exiting to and entering from Internet at that point. So how can I handle it better? If I have a mechanism where I load share between IDS5 and IDS3 then it is possible (25 signatures each). This load sharing is now possible using traditional IDS devices, the only problem is that a Network Administrator has to tune it accordingly and deploy that set of signatures to BOTH of the sensors. If you need to have different set of priorities for signatures or different set of signatures itself, the administrator has to do it all by himself. This is a hectic job apart from forensic analysis. Now think about if the sensors can talk to each other and update them based on the previously configured policies for day and night, wouldn't that be cool?

Another scenario would be if a SMTP attack is detected by IDS4, then it is obvious that it is missed by

both IDS5 and IDS3. In such cases, IDS4 can inform this to either IDS5 or IDS3 so that such attacks could be trimmed at the very entry itself. Also blocking that particular host sending the attack would be easier because both the IDS5 and the firewall reside on the same segment. The same applies to the internal network attacks also.

Now the obvious questions! Who will be Publisher, What will be the mode of communications? Selecting publisher can run through a selection process as in routing protocols. However, manual selection of IDS could also be an option. In today's networks, it is rated that "Internal" attacks/attempts are more predominant than the "External". So it has to be a careful and well thought process to select the Publisher. Then again, multiple publisher scenarios also are possible. When it comes to mode of communication, no other choice, just encrypt them. Most of the Modern IDS solutions already do support IPSEC level of encryption. More efficiency can be brought by introducing "Intrusion Forecasting" into the IDS itself by models like Markov Model. The challenge is to make a compromise between Costs to the Organization and worth of information secured, by owning such equipment!

## Summary

Integrated communication between sensors could yield advantages like;

- Grouping of sensors based on Geographical or Departmental basis.
- Signature updates/Policy updates can be carried over in a more automotive and efficient manner.
- Dynamic Signature tuning based on the network dynamics (A group of sensors can have intelligence to identify the interesting sniffing traffic over a period of time and disable the unwanted signatures.
- Load sharing is possible to reduce the database size on individual sensors.
- OS fingerprinting can be integrated and can have Master-Slave relationship for mapped information so that the traffic can be trimmed off at the first stage of detection itself.
- Communication of different type of attacks can be handled better. Say Sensor A is the edge sensor (first) and Sensor B being the internal sensor. Now if Sensor B sees a syn flood and blocked it, Sensor B can ask Sensor A to block that traffic at the perimeter itself. This way blocking can be made effective like ACL rules.
- A traffic which misses IDS/IPS mechanism (due to lack of signature) can still be found by a Host based IDS. Such information can be communicated to the nearest Network sensor for automatic creation of custom signatures.
- Recover from attacks that are meant for IDS device. If an IDS device is compromised, other devices in the group can have the intelligence to bring up all the signatures and protect the network efficiently in no time.

Its time Security Solutions provide an integrated solution instead of point patches which spoils all the concept of inventing the computers itself. Well, we would all like watching "Hackers" or "Perfect Storm" but not have one on our networks!!!

## Author Biography

Rajesh T Sivanandan holds a Bachelors Degree in Electrical and Electronics Engineering. He is currently working with HCL Technologies – NPD in Chennai, India. He is having 4 years of experience in Networking and held certifications like MCSE, CCNA, CCNP, CSS-1, CCIE (Theory).