ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

A survey on Mobile Agent Based Intrusion Detection System

Trushna Tushar Khose Patil[1], C.O.Banchhor[2]

Lecturer, Department of Computer Engineering, BSCOER, Pune, India1

Assistant Professor Department of Information Technology, SCOE, Pune, India2

ABSTRACT: The widespread proliferation of network connections has made current computer networks more vulnerable to intrusions than before. In network intrusions, there are multiple computing nodes that are attacked by intruders. The evidences of intrusions are to be gathered from all such attacked nodes. The detailed architecture and implementation of a prototype of DIDMA are described. In this paper, the performance metrics of different IDS are compared.

***Keywords*: MAD, VHL, HOST, MA, CONSOL**

I. INTRODUCTION

There are many kinds of cybercrimes related to computer networks which are connected to Internet. Intruders attack on multiple computing nodes in network intrusions. An Internet user having malicious intention may access, modify, or delete sensitive information present on other computers. Some of the computer services may be made unavailable to other users. Due to huge and complex infrastructure of computer network.

Intrusion detection system (IDS) is needed. Whenever the confidentiality, integrity, and availability of computer resources are under attack, it will help to detect and respond effectively. From all such attacked nodes the evidences of intrusions have to be gathered. An intruder may move between multiple nodes in the network. Due to this the origin of attack is concealed. We propose a new intrusion detection system (IDS) called Distributed Intrusion Detection using Mobile Agents (DIDMA).This system is helpful to detect such intrusion activities spread over the whole network.

There are some issues with the existing centralized ID models like,

1. If the new host is added, the load on the centralized controller increases significantly.
2. It makes the IDS non-scalable. Communication with the central component can overload parts of the network.
3. Some IDSs contain platform specific components.

We are implementing distributed IDS called Distributed Intrusion Detection using Mobile Agents (DIDMA). It overcomes issues which are present with centralized ID models. DIDMA uses software entity i.e. Object of Mobile Agent. These are software components and can migrate to all the hosts in the network. They can execute the tasks of detecting intrusions autonomously. Hosts detect the suspicious activity. Such Hosts who has detected suspicious activities are visited by Mobile agents. Mobile agent at visited host collects the amount of data which contains attack trace from that host. Then Mobile agent aggregates and correlates it with the data it has got from other Host that have detected the same type of suspicious activity. MA carries this resultant data to the next host which is to be visited same process is repeated at every visited host. It is decentralized data analysis carried out by mobile agents so IDS becomes more scalable. The failure of the controller module does not stop the currently ongoing ID tasks because even when MA disconnected from the controller module that created and dispatched the mobile agent, MA can function autonomously and execute assigned tasks so DIDMA more reliable. DIDMA is platform independent because it's all components are developed using JAVA platform.

II. RELEVANCE

Due to mobile agents in DIDMA it is possible to reduce network bandwidth usage. It increases scalability and flexibility. It can be able to operate in heterogeneous environments. DIDMA offers a new and good technique for decentralized data analysis. This is carried out by mobile agents at the site of audit data instead of sending the audit data to some central data analysis component. Small sized mobile agent code and attack trace data carried by mobile agents. Due to this there are fewer loads

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

than sending large amount of raw data sent over the network. Mobile agents offer unique features that can be used to improve the ways in which IDSs are designed, developed, and deployed in the network. Even when disconnected from the controller module, mobile agents can function autonomously and execute assigned tasks. So, the failure of the controller module does not stop the currently ongoing ID tasks. Thus it makes DIDMA more reliable. The components of DIDMA are developed using JAVA platform making it platform independent and operable on heterogeneous platforms. But our system is more reliable with compare to existing system because instead of moving MA code, our system makes a object of that code and migrate that code from one host to other host. It will take less amount of time for data migration.

III. DIDMA ARCHITECTURE

Overall System Architecture

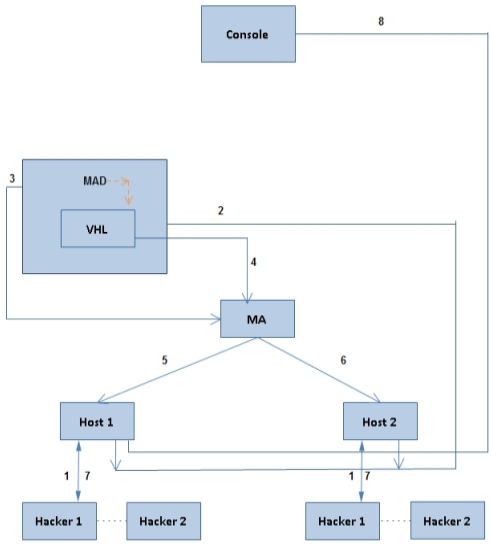

Figure 1. DIDMA Architecture

Our system consists of separate modules. Each module is complete in it and acting as a discreet processing unit which makes it easy to modify any block, provided the input and output data types remain compatible with the connecting blocks.

The modules are:

A. Mobile Agent (MA)
B. Host
C. MA Dispatcher (MAD)
D. IDS Console
E. VHL

We proposed intrusion detection system DIDMA. It is made up of different components like Host, Mobile Agents (MA), Mobile Agents Dispatcher (MAD), VHL and IDS console. MAD has subcomponent called VHL. ID event will get generates when suspicious activities are detected like failed login attempts, suspicious connections, port scanning, or modification of system sensitive files from suspicious users. The respective Host sends ID events related to such suspicious activities the MAD. Then MAD creates Mobile Agent to handle the task of detecting intrusions based on such activities. The VHL have lists that contain the IP addresses of hosts on which suspicious activities are detected. The address of the Host that generated the ID event is added to a list in the VHL. There are separate lists for each type of attack on VHL. The hosts listed in the VHL are visited by MA. The MA takes traces of an attack form visited Hosts. Then and aggregates and correlates the collected information with the data it has received from the other Hosts who have generated the same type of ID event. As guided by the VHL the resulting data is carried over to the next host to be visited. At every visited host same process is repeated. On the detection of any attack MA generates alerts. IDS console receives generated alerts from MA and displays the alerts to the security administrator using the IDS console. (Refer Figure No 1)

A. Mobile Agent (MA)

It is responsibility of Mobile agents to collect evidences of an attack from all the attacked hosts and further analysis of the gathered data. An MA takes the route which is specified in the VHL. MA contains less programming code because it is only for detecting a specific type of attack so size of each MA is small. We can easily extend the IDS by adding new MAs for detecting new attacks. We can modify existing MAs for better detection capability it results in a highly modular and extendable architecture. The MAs take data from all the hosts listed in the VHL then aggregate and correlate the data, and generate alerts. When movement from one host to another occurs the aggregation and correlation take place. Single centralized module does not perform the aggregation and correlation of data collected from the SAs so it results in highly decentralized data analysis architecture. The Functioning of the MAs after it is dispatch is not interrupted by the failure of the MAD

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

module. Due to this IDS can complete the currently running intrusion detection tasks even in case of failure of the MAD. In attacks like doorknob rattling, the data collected from Host have to be just aggregated to detect the attacks. But in chain or loop attack, the data have to be both aggregated and correlated. The alert generated by an MA is sent to alerting subsystem contained in the Console.

B. Host

Host monitors the generating ID events whenever a trace of an attack is detected. These events are sent to remote object in the MAD in the form of a message. Each ID event carries information about the probable type of attack. For example, if host identifies failed password guessing attempts as a suspicious activity, then generated ID event is to check for doorknob-rattling attack. Again, if a large number of connections generated from a host within a short period of time then it destined for a single target will trigger an ID event to indicate DoS activity on the host. The Host is responsible for pars the log files, checks the intrusion related data pattern in log files, separates data related to the attack from the rest of the data, and format the data as required by the MA.

C. MA Dispatcher (MAD)

According to the ID event generated by host MA Dispatcher (MAD) decides which MA has to be dispatched. MA originates from MAD. The MAD is nothing but a program that initiates an object request broker server. Host communicates with the MAD using the proxy of the objects created at the MAD for sending event messages. Then these objects are responsible for creating an MA and sending it to the victim host(s). The MAD contains Victim Host List (VHL). These are use to maintain separate lists to store the IP addresses of all the hosts that are subjected to same types of attacks. E.g. all the hosts subjected to doorknob rattling attack are maintained as a separate list in the VHL. The VHL provides the guidance for the movement of an MA within the network.

The IP address of that host is added to the respective list in the VHL When the MAD receives an ID event message from a Host.

D. IDS Console

MAs send alerts generated to IDS onsole. IDS console is used by the security administrator so that he can alerts generated from the IDS.

F. VHL: (Victim Host list)

VHL Module is intended to retrieve and store the IP addresses of the infected host's in the network. It receives IP addresses from MAD i.e. Mobile Agent Dispatcher.

Further these IP addresses of infected host's are used by Mobile Agent in order to visit the infected hosts.

IV.Attacks

A. Doorknob-Rattling Attack

In this attack a very few common username and password combinations tried by the intruder on several computers which results in failed login attempts. Unless from all the hosts, the data related to login failures are collected and aggregated to check for doorknob-rattling from any remote destination, this attack can go undetected. After detecting a predefined number of failed connection attempts from legal or illegal users, host generates an ID event which is the indication of probable doorknob-rattling attack. Generated event is then sent to the MAD. MA is dispatched by MAD for detecting this attack. The IP addresses of all the hosts that detect this attack are added to a list in the VHL. For detecting doorknob-rattling, all the hosts listed in the VHL are visited to gather traces of doorknob-rattling data. MA aggregates this data to detect the attack. It gives the list of victim hosts which any remote host has tried to connect with the list of usernames used to connect to those hosts.

B. Chain/Loop Attack

In this attack intruder tries to hide his point of origin by moving across several hosts. This results the chain of connection which is passed through many hosts. In a loop attack the chain of connections makes loop, here it is harder to track the source of the connection. Intruders use chain attacks to hide their identity. An SA notifies the MAD by generating an ID event of every suspicious successful login attempt. If more than a predefined number of failed login attempts are made before a successful attempt then login attempt can be deemed suspicious. Then MAD dispatches an MA to all the hosts that have generated this ID event to detect any loop/chain attack.

C. DOS Attacks (Denial of service attack)

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person

Not all service outages, even those that result from malicious activity, are necessarily denial-of-service attacks. Other types of attack may include a denial of service as a component, but the denial of service may be part of a larger attack.

Illegitimate use of resources may also result in denial of service. For example, an intruder may use your anonymous ftp area as a place to store illegal copies of commercial software, consuming disk space and generating network traffic.

D. Modification attack

An unauthorized party not only gain access to but tamper with an asset. This is an attack on integrity. Example is change value in data file, alerting a program, modifying the content of message being transmitted in network.

E. Replay attack

Users capture the sequence of events or some of data units and request them. For instance user A wants to transfer some amount to user C's account. User C could captures user A's message when it is being transferred and send a second copy of the same. The bank would have no idea that it is an unauthorized message. Therefore user C would get the benefit through replay copy through replay attack.

F. Worm attack

Worm is similar to the concept of virus but different in implementation. A virus modifies a program means it attacks itself to the program under attack. A worm however does not modify a program. Instead, it replicates itself again and again. The replication grows so much that ultimately the computer or network, on which the worm resides, becomes very slow, and come to a halt. Thus basic purpose of worm attack is different from virus. A worm dose not performs any destructive action and instead only consumes system resources to bring it down.

An e-mail virus has some characteristics of worm, because it propagates itself from system to system. However we can still classify it as virus because it requires human to move forward. A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines.

Network worm programs use network connections to spray from system to system. For replication network worm uses some network facilities as e-mail, remote execution and remote login. As with virus, network worm are difficult to counter.

Example of macro worm: An electronic "Christmas card" passes around several IBM network. It instructs the recipient to the message and run it as a program. The program drew Christmas tree and prints" Merry Christmas". It then checked the recipient's list of mail and address book to create a new list of e-mail address. It then sends copies of itself to all these addresses. The worm quickly overwhelmed the IBM network and forced the networks and system to be shut down.

My doom is example for 1000 times per minute and reportedly flooded the Internet with million infected messages in 36 hours.

Ideal solution to the threat of virus is prevention and it is impossible to achieve. Prevention can reduce the number of successful virus attacks.

G. File Property Attack

An attack made by attacker in which the extension of the file get changed to .exe from its original extension say .doc, which results into failure of file open or access related operation.

V. MA ALGORITHM

```
For (Agent)
  Begin
      (Extract a target address)
     If (exist)

          //Check whether connect, waits the agent during
          some system time     stamp.
       Try to connect socket to the address

   If (Success)
```

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

```
    Begin
    Call go agent
    Exist;
 End

 Else
 Begin
   Do not call the agent

End
End
```

VI. CONCLUSION

We are implementing a distributed intrusion detection system using mobile agents called DIDMA. It overcomes some of the disadvantages of the centralized distributed intrusion detection systems. DIDMA uses Hosts as monitors and mobile agents for collecting data, aggregation and correlation, and to give response to any attack. Use of mobile agents in DIDMA makes application advantageous such as it reduces network bandwidth usage, it increases scalability and flexibility. It can be able to operate in heterogeneous environments. DIDMA offers a new and good technique for decentralized data analysis which is carried out by mobile agents at the site of audit data instead of sending the audit data to some central data analysis component.

VII. REFERENCES

[1] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. David Dittrich, "The stacheldraht distributed denial of Levitt, "Network intrusion detection," IEEE Network, 8(3): service attack tool," University of Washington, December 26-41, May/June 1994. 1999.

[2] S. Snapp et al, "DIDS (Distributed Intrusion Detection [15] Wayne Jansen, and Tom Karygiannis, "Mobile Agent System) - Motivation, Architecture, and An Early Security," National Institute of Standards and Technology, Prototype," Proc. of the 14th National Computer Security Gaithersburg, MD, 1999.Conference, pp. 167-176, Washington D.C., October 1991.

[3] W. Jansen, P. Mell, T. Karygiannis, and D. Marks, [16] P. Kannadiga, M. Zulkernine, and S. Ahamed, "Applying Mobile agents to Intrusion Detection and "Towards an Intrusion Detection System for Pervasive Response," NIST Interim report (IR) – 6416, National Computing Environments," to appear, Proc. of the Institute of Standards and Technology, USA, October 1999. International Conference on Information Technology (ITCC), IEEE CS Press, Las Vegas, Nevada, USA, April 2005.

[4] M. Roesch, "Snort – lightweight intrusion detection for networks," Proceedings of the USENIX 13th System Administrations Conference, LISA'99, Seattle,Washington, USA, November 1999.

[5] Ko, D. Frincke, T. Goan, L. T. Heberlein, K. Levitt, B. Mukherjee, and C. Wee, "Analysis of an Algorithm for Distributed Recognition and Accountability," Proceedings of the first ACM Conference on Computer and Communication Security. Fairfax, VA, Nov. 1993.

[6] Mobile Agent Algorithms versus Message Passing Algorithms J. Chalopin1, E. Godard2, Y. M´tivier1 and R. Ossamy1 e 1 LaBRI UMR 5800 ENSEIRB - Universit´ Bordeaux 1 e 351 Cours de la Lib´ration e 33405 - Talence France {chalopin,metivier,ossamy}@labri.fr 2 LIF UMR 6166 Universit´ de Provence e 39 rue Joliot-Curie 13453 Marseille France.