# AN INTELLIGENT INTRUSION DETECTION FOR DETECTING UNAUTHORIZED MALWARE OVER THE NETWORK

Manjunath KG [#1], Dr. N. Jaisankar [*2],Shreedevi KG[#3]
[#] Research Scholar, VIT University ,Vellore And Assistant Professor
Siddaganga Institute of Technology, Tumkur , India
[1] sitmanju@gmail.com
[*] Professor ,School of Computer Science and Engineering
[2] njaisankar@vit.ac.in
[3]shreedevipramod@gmail.com

***Abstract*—**

**Monitoring Internet traffic is critical in order to acquire a good understanding of threats and in designing efficient security systems. It is the most important issue to monitor the traffic in internet and also in designing efficient security systems. Honeypot is one of the security tools for gathering intelligence of Internet attacks, traffic collected by honeypot is of high dimensionality that makes it difficult to characterize. In this paper, a multivariate analysis technique, for characterizing honeypot traffic and separating latent groups of activities is used. A multivariate analysis consists of collection of methods that can be used for detecting unauthorized malware over the internet. Data visualization, Data mining and statistical techniques are the multivariate analysis techniques for characterizing Honeypot. The internet has become a platform for all kinds of security-sensitive services and applications. In this modern era of computing, internet plays an important role and therefore, securing network hosts, learning attack methods, capturing of attack tools, and studying motives of computer criminals are important tasks for network administrators and security engineers. One important aspect of network attacks is malicious software (*malware*) that spreads autonomously over the network by exploiting known or unknown vulnerabilities. The various elements like web browsers, e-mail client and office are absolutely not secure with the development of new client application software vulnerabilities. This paper highlights the development strategy towards intrusion detection system based on honeypot. It is a trap set to detect, deflect towards any unauthorized/ anonymous malware distributed globally over the networks. We achieved designing a prototype with a unique network crawler which will keep track the illegal software but it has also potential to track the source URL from which the malicious events are taking place at the client side.**

**Keyword- Honeypot, VMware, diagnose, threadpool , FAIR**

## I. INTRODUCTION

There were several security techniques introduced to fight against various cyber attacks, in the past, firewalls, intrusion detection systems (IDSs) and intrusion prevention systems (IPSs), to name a few security techniques. all these techniques have contributed to the construction of a higher-level security architecture. But still there is a fatal weakness that they still suffer from unknown intrusions (i.e., 0-day attacks) which have not been identified as hazardous activities. Hence it is a major challenge in network security researches to identify such an attack. However new Operating systems have become secure, attackers are shifting their targets from Operating systems to applications, for example, even backdoors opened by some worm/virus also become their victims. Unlike Operating Systems, however, it is difficult to get precise information about these applications. Attackers are keep on attacking the applications until their exploit codes can work correctly. It is required to collect large amount of traffic data including unknown exploit codes. Honeypot is one of the most popular tools to decoy attackers into our network, and to capture lots of information about the activity of malicious attackers. Honeypots are electronic network resources (computers, routers, switches, etc.) deployed to be probed, attacked, and compromised. Although many honeypots have been proposed, there is a common problem that they can be detected by someone (mainly malicious attackers) easily. This is very important in the success or the failure of honeypots since if once an attacker notices that he/she is working on a honeypot , we can no longer observe his/her malicious activities. In fact, they showed that someone can identify the existence of honeypots within a short period only by sending their pseudo-packets and checking the reports of log analysis centers such as the SANS Internet Storm Center. In addition, if a honeypot is based on virtual machines such as VMware, it can be

characterized as honeypot by monitoring system information such as network devices, the process lists and so on.

## II. RELATED WORK

Handling Insider threat is very challenging especially when they get aggravated by money or revenge [1].With Insider attacks, organizations face financial loss as well as reputation and also loose intellectual property. The insider threat problem is more hard to pin down and puzzling than any other threat. Assessing the insider threat is the first step to identify and resolve the possible insider attack. Technical solutions are not adequate since insider threats are fundamentally a people issue. A combination of Technological, behavioral and organizational assessment is necessary in facilitating the forecast of insider threats and anticipates any insider attack. This improves the organizations security, survivability and resiliency in light of insider threats.

Over the past couple of years, highly advanced and most determined threats are used by attackers. They are using information gathering tools [2], attacker tools and sophisticated methods. Attackers using social engineering to gain entry into networks and then changing the existing malware code . The goal of advanced persistent Threats is to gain access to Targeted information and to maintain a presence on the targeted system for long term control and data collection.

The insider threat to information security cannot be eliminated but it can be assessed and managed. The application of Technology [3] alone will not provide solutions. Security controls must work in variety of environments and designed, implemented and maintained with people's behaviour in mind .The designed solutions must be responsive and must maintain trust, secure relationships overtime. Hence a focus on human factors, more attention on the security and education is required to asses and manage the insider attack threats. The Flexible Automated Intelligent Response system (FAIR) architecture [4] is also one of the approach for handling the IDS based on Response to suspected incidents being handled by an automated Responder

## I. PROPOSED SYSTEM

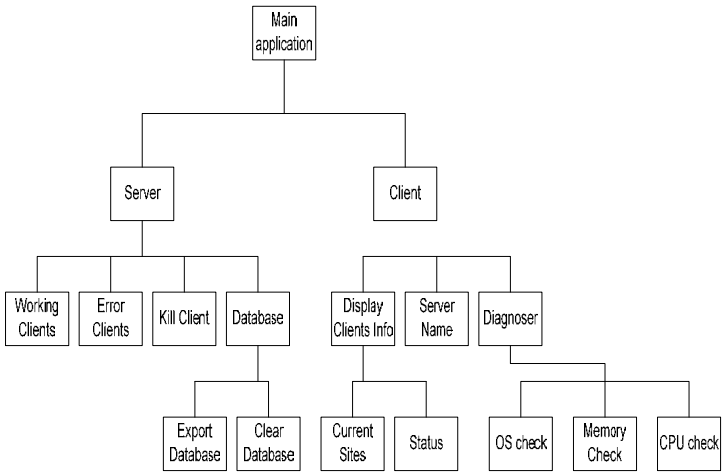


Fig 1.Structure chart proposed System

The proposed system consists of two sections Server Side and Client Side. The server acts as an administration layer which logs the event and provide control over the clients. The intention of the server is to monitor the events going on in the number of the client's PCs. To give a wide view, the server is designed with a prospective of no limits for the amount of clients it can support, and this is determined by the power of the platform it is run upon. The client basically acts as a honeypot. The dispatching module of the client searches a dictionary module for a word inside the Google query, or URL. It than travel to that URL using IE and wait for the IE to signal the page has been fully loaded. Once the page has been loaded, the dispatcher runs a self diagnosis according to what was configured before it was started. There will be a provision of reporting system which has to be sent to the remote server using Remote mechanism.

Client-side honeypot needs data source. We use crawler to get the URLs, and then this URLs will be analysed in the honeypot. However, the speed of the crawler and the honeypot doesn't match. The crawler is faster in behaviour compared to honeypot. The behaviour of the honeypot is restricted by the network bandwidth and the performance of the Operating System. The crawler must be designed based on known attack techniques to filter URLs and get more malicious URL, which can reduce the rate of opening non-malicious web-pages. The system is designed as a honeypot, when one of the computers serves as a server, which can reside not in line with the rest of the network. The system is based upon an external SQL server which will register the honeypot activity and help the security expert to tune his real network.

Structure chart (cs) in software engineering and organizational theory is a chart, which shows the breakdown of the configuration system to the lowest manageable levels. It is used to show the hierarchical arrangement of the modules in a structured program. .a structure chart depicts the size and complexity of the system, and number of readily identifiable functions and modules within each function and whether each identifiable function is a manageable entity or should be broken down into smaller components. Figure 1 shows the structure chart of the proposed system.

*a. Client registration*

After the user has configured it's client to its proper diagnostics and server, the client, through remoting, sends event that it has joined. The Server in his turn writes an entry in the remote SQL database.

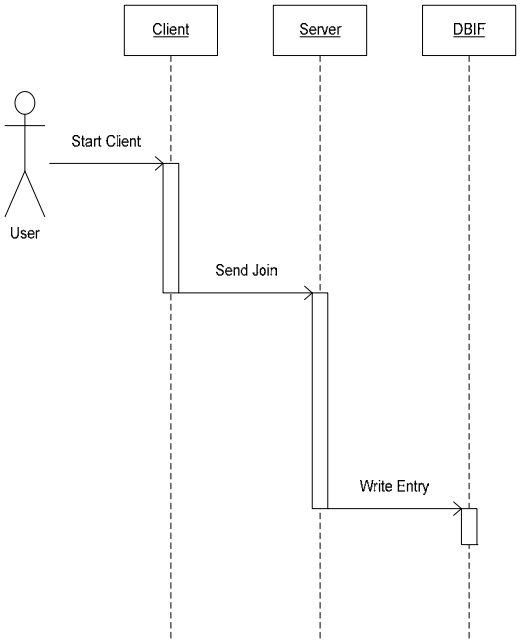


Fig5. Sequence Diagrams-I of the system

The client is triggered by finish of loading a page and waiting for "enough" time, or timeout has passed for IE reply. The client than chooses a URL randomly and browses there, checks its status and send to the server. The server registers the report in the database.

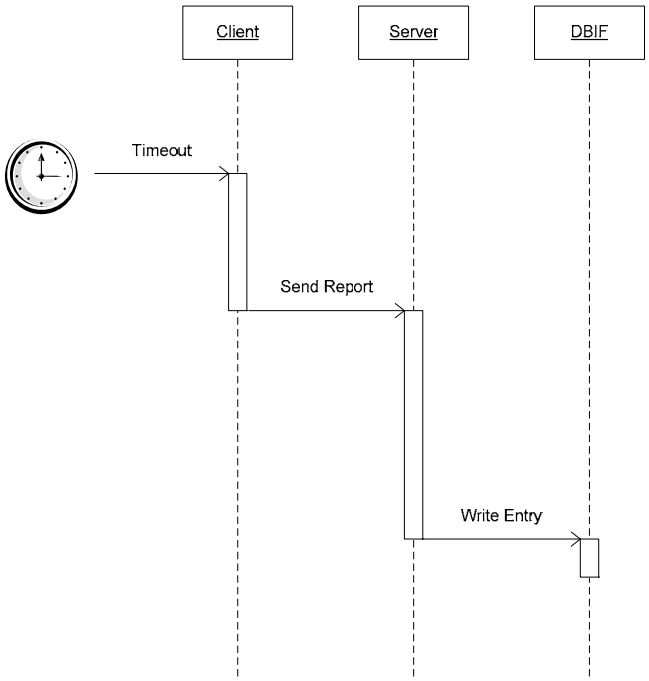


Fig6.Sequence Diagram of Client Reporting System

The dispatcher listens to the GO event which is driven by the GUI. Once it is set, the dispatcher starts an IE_Interface task which will start a new IE instance and will browse the web. Once a page has been completely loaded, the IE_Interface signals the dispatcher, and the dispatcher uses the diagnosers in his container to check its status. After it is done constructing a status string it sends it to the server thru the Server_Remote_Interface.

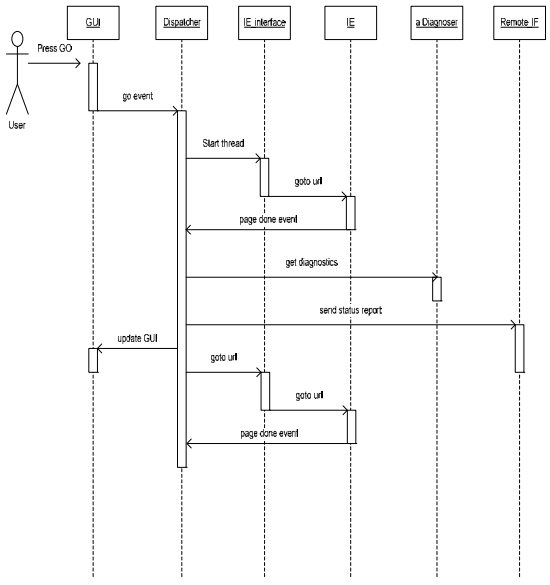


Fig 7. Sequence Diagram of Client Work

The server work is initiated by events generated by the client. When the dispatcher is notified on such event, it checks the string that is carried with the event and parse to see if its status is BAD or GOOD. According to the status it than generates a query to the SQL server and send it thru the database interface (DBIF). The last phase is getting a complete log from the database and displaying it in the GUI.

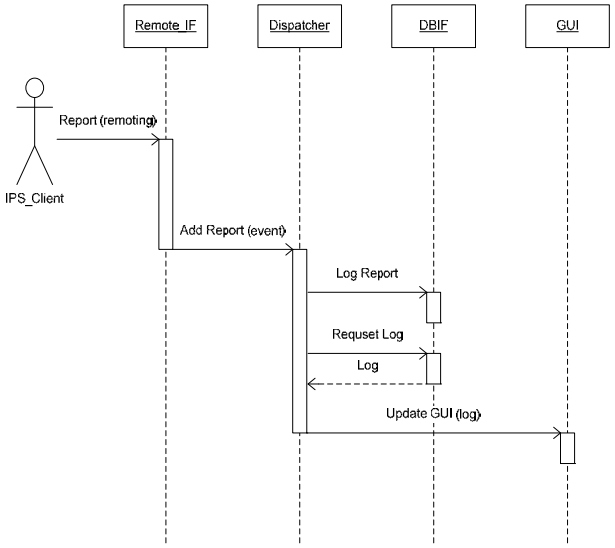


Fig 8. Sequence Diagram of Server Work

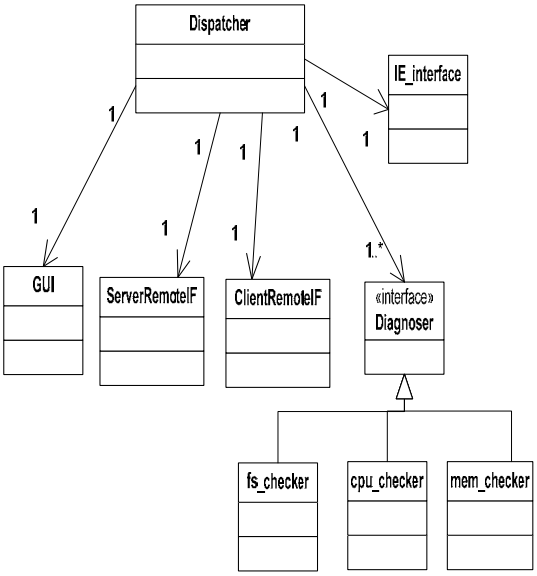


Fig 9. Client-Relationship Diagram

Figure shows the Client-Relation Diagram. The Dispatcher class gets services from all the other classes, and communicates with another thread of the IE Interface. It holds a container of classes that implement the Diagnoser interface, and uses them to diagnose the system that runs the application. The GUI is another thread which is part of the application, and all the inter-thread communication is event driven, which is part of the application, and all the inter-thread communication is event-driven, which is faster and lighter than poling on semaphores.

## II. RESULTS AND DISCUSSION

Microsoft Visual studio is used for the implementation of the work.

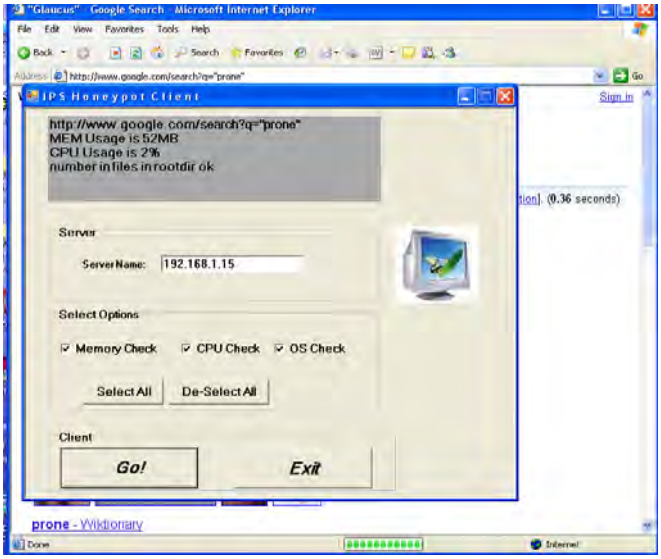


Figure 5.1: shows client application accepting server URL for checking memory usage, cpu usage and number files in the root directory.

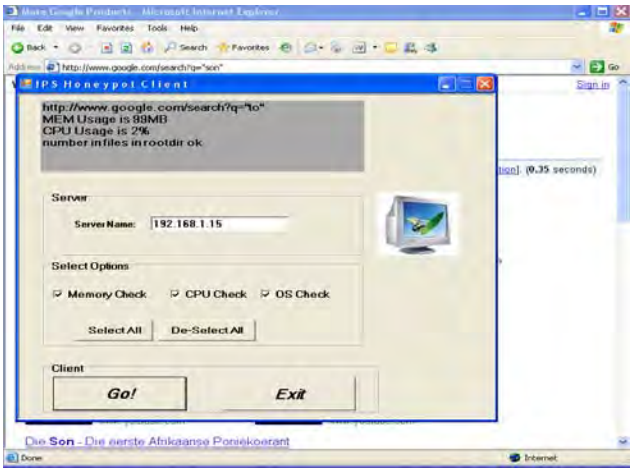


Figure 5.2 : shows client applications memory usage, cpu usage and number files in the root directory.

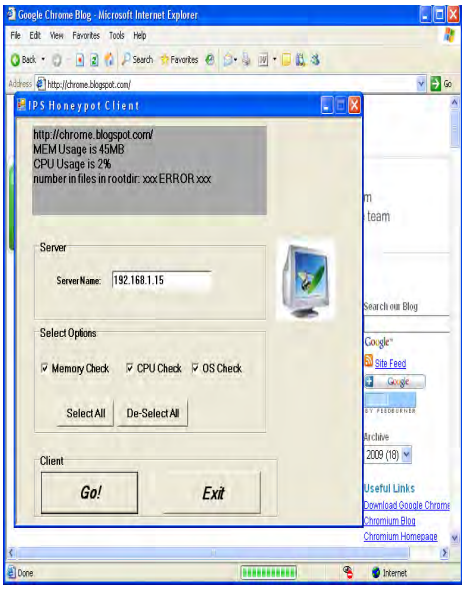


Figure 5.3 : shows hacker fails to access server application

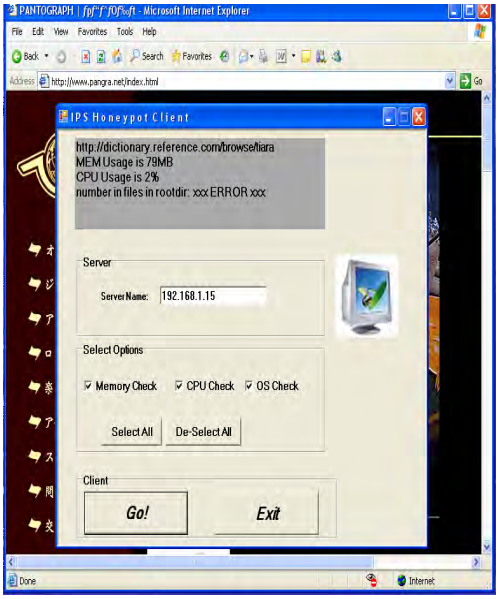


Figure 5.4: shows hacker fails to access server application

## III. CONCLUSION AND FUTURE WORK

Congregation information regarding the topics covered in this system are hard to come by, and was mostly achieved by trial and error. It is wiser to modify the protocol between the clients and the server in a way that will enable the server to know where the client is heading, and than if it succeeded. In this way, if the client found a malicious page that caused him to crash, the server will also know about it. This change requires a change in the server architecture (work with threadpool and negotiate session with each client). Using IPS system is very effective and easy o deploy. A model in which IPS is run in idle time should be considered in any computer farm, since the computing power is handy. Having a main server which will give the client dictionaries and starting point might be a good idea, since it will avoid checking same places twice, and give a more flexible approach to the security expert managing the system. If the system (client) were to be distributed on a livecd, it would be safer to deploy in working networks, since it guards their actual data and OS. Another option is using virtual machines like VMWARE Images that can be freely handed in ordered to be used for this target.

- Following are the future works, which includes changing the protocol between the clients and the server, so a client may tell where he intends to go, and then report if he got there. Manage a thread pool and assign each client a thread which will collect better data of the client behaviour.
- URL management – in the current configuration, each client decides for itself where it should go. If we modify the system to work more like a search engine, each client will report to server of the page it found and what it stripped off of him. The server will maintain a database (also in an SQL server), and will tell the client where should it go. In this way, we solve the problem of two clients going to the same page, and have better utilization of our honeynet resources.
- In the pursuit of safer networks, it is possible to combine the data collected from such a project with a DNS server, and give each site a grade that tells how safe it is, or how safe it domain is, new versions of Internet Explorer can have a gauge that can be set to the amount of security the user wishes to have, and in this way we can block whole segments of malicious networks.
- It is possible to add a new diagnoser to the system that checks the validity of the cookie requests made by the site. Site that planet cookies may be marked as unsafe (according to user specifications), and sites that request cookies they didn't plant, are also of malicious type. For example, a site that requests a cookie of some bank, or the Google cookie, should also be considered dangerous.

## REFERENCES

[1] Kuheli Roy Sarkar ,“Assessing insider threats to information security using technical, behavioural and organisational measures” , ELSEVIER,15 2010,pp.112-133
[2] Colin Tankard, Digital Pathways ,“ Persistent threats and how to monitor and deter them”, Network Security, August 2011
[3] Carl Colwill, “Human factors in information security: The insider threat e Who can you trust these days?”, ELSEVIER , i n f o r m a t i o n s e c u r i t y t e c h n i c a l r e p o r t , 1 4 ( 2 0 0 9 ), pp. 1 8 6-1 9 6.
[4] M. Papadaki, S.M. Furnell “Informing the decision process in an automated intrusion response system”, ELSEVIER , i n f o r m a t i o n s e c u r i t y t e c h n i c a l r e p o r t , 1 0 ( 2 0 05 ), pp. 1 50-1 61.

[5] J. Bethencourt, J. Franklin, M. Vernon, "Mapping Internet Sensors with Probe Response Attacks", In Proceedings of the 14th USENIX Security Symposium, pp.193-208, 2006.
[6] Y. Shinoda, K. Ikai, M. Itoh, "Vulnerabilities of Passive Internet Threat Monitors", In Proceedings of the 14th USENIX Security Symposium, pp.209-224, 2006.
[7] T. Holz, F. Raynal, "Detecting Honeypots and other suspicious environments", In Proceedings of the 6th IEEE Workshop on Information Assurance and Security, pp.29-36, 2005.
[8] N. Provos, "A Virtual Honeypot Framework", 13th USENIX Security Symposium, 2004.
[9] Leita, C., Mermoud, K., Dacier, M., "Scriptgen: an automated script generation tool for honeyd", In Proceedings of the 21st Annual Computer Security Applications Conference, 2005.
[10] Leita, C., Dacier, M., Massicotte, F., "Automatic Handling of Protocol Dependencies and Reaction to 0-dayAttacks with ScriptGen Based Honeypots", 9th International Symposium, Recent Advances in Intrusion Detection, LNCS 4219, pp. 185-205, 2006.
[11] M. Bailey, et. al., "The Internet Motion Sensor: A Distributed Blackhole Monitoring System", 12th Annual Network and Distributed System Security Symposium, 2005.
[12] Walker Terrence. Information Security Group, Royal Holloway, UK. Practical management of malicious insider threat – An enterprise CSIRT perspective. Information Security Technical Report November 2008;13(4):225e34.
[13] Jones A, Colwill C. Dealing with the malicious insider. In: 9th Australian information and Warfare security Conference; December 2008.
[14] Raywood D. Companies being hit by moles who are employed by gangs to steal data. Secure Computing Magazine; 2008. 2/10/08. Royds J. Virtual battlefield. CIR Magazine; August 2009.
[15] Toth T, Kruegel C. Evaluating the impact of automated intrusion response mechanisms. In: Proceedings of the 18th annual computer security applications conference (ACSAC). 9e13 December 2002