Implementation of Honeypot

Mr. Saurabh Alva, Mr. Rahul Madhyan, Mr. Anoop Madan

***Abstract*— In the IT world, information is considered to be the most valuable asset for any organization in today's times. The ability of an organization to be able secure this asset is a critical factor and the art of securing this asset is known as information security. In today's competitive IT business, network administrators must be competent enough to protect the network and the information on the One of them is Honeypot. Honeypot reduces the overhead of the network administrator which is- 'always be on the network and always monitor it'. Honeypot is a setup to imitate a real network. The idea is to create a fake environment in order to deny the fake user accessibility to the system. This Paper proposes the methodology to identify and trap the misbehaving user.**

***Index Terms*— network security, fake user detection**

I. INTRODUCTION

The ability to secure data on the file server is of main concern these days. Organizations tend to have some kind of sensitive data, which critical inorder to develop market competitive products. This critical information is the subject of interest of many such attackers.

There are different kinds of attacks that an intruder tries to employ on the file server of the targeted organization and fetch the data of their interest.

kinds of attacks employed could be -

•1) The intruder pretends to be the genuine user and registers himself on the fileserver thereby enabling him to collect all the data on file server.

•2) The intruder tries to interpret the data packets that are transmitted from genuine user to client server which in turn may lead data leakage.

• 3)The attacker tries to forge the login details of a client that is already registered.

II. LITERATURE SURVEY

Anonymous Networks.

When dealing with anonymous networks the aspect of security is that much more constrained since the identity of the attacker is not clearly attainable. The employment of advanced algorithms of methods of detection may produce a hefty strain on the financial feasibility of such methods , thereby undermining their importance of said technique to quite a large extent.

Using the Honeypot architecture the ability to safely detect intrusions in anonymous networks is not only promised but can be feasibly achieved as well.

The advent of anonymous architectures in the new era of technology advancement is undeniable and its involvement in data sharing is unavoidable. Rather than eliminating such constraints the Honeypot

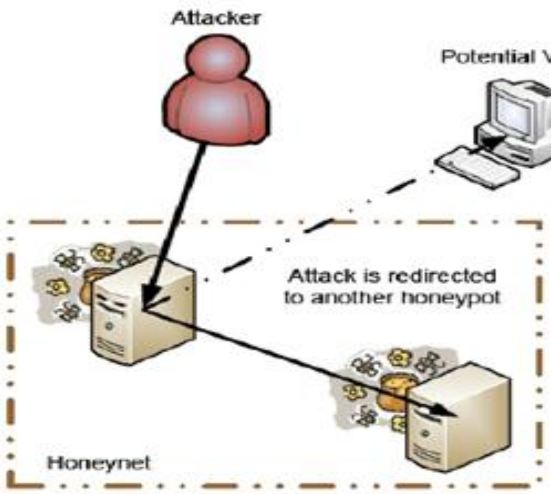
architecture adapts to said constraints and finds an optimal solutions.

III. SYSTEM ARCHITECTURE .

The basic architecture of the system is systematically depicted in the figure. The various components that go into making a successful 'Intrusion Detection System' are as follows :-

• Attacker :- This refers to the compromised user/attacker that the Honeypot system is built around. The attacker could be anyone that

intends to obtain information for any purpose that may include plagiarism , system disorientation , information disclosure , etc .



Attacker is fooled by redirecting the call to honeypot system

• The potential Victim :- This refers to the primary target of the attacker . The victim could refer to a single person or a multiscale organization . The vulnerability of the victim depends on size of the organization as well as parameters such as brand value, data dependability etc. The effect of the attack could be widespread or could be localized on a single domain in the organization. The attacks could disrupt the normal flow of operation in the network and may also lead to data inadequacy .

•The Honeypot :- This refers to the actual 'Intrusion Detection' mechanism that is active/present to successfully detect unauthorized access of the systems data. The various methods employed would be explained the leading sections.

Mr. Saurabh Alva, Department of Computer Science and Engineering, The National Institute of Engineering, Mysore,Karnataka, India

Mr. Rahul Madhyan, Department of Computer Science and Engineering, The National Institute of Engineering, Mysore,Karnataka, India

Mr. Anoop Madan, Department of Computer Science and Engineering, The National Institute of Engineering, Mysore,Karnataka, India

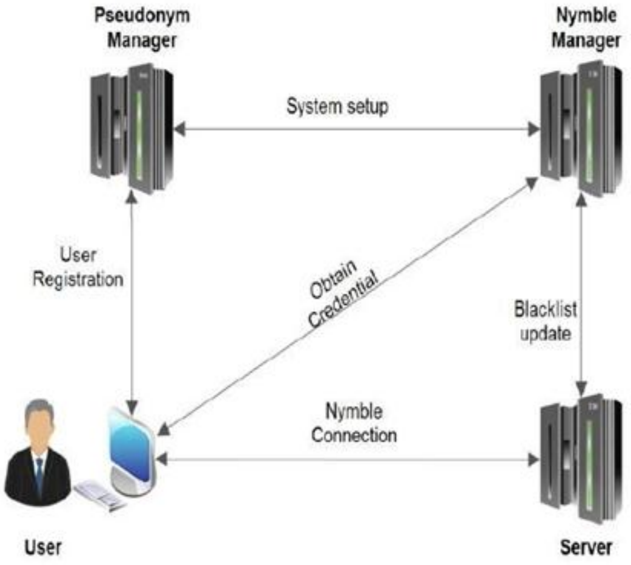
IV. Design Methodology

The quantitative design of the system may differ from architecture to architecture . This paper proposes a concrete architecture that enables users/clients to easily adapt to the configuration of the system. Various stages involved are as follows.

•STAGE 1 :- Normal Operation.

The normal working of the system as depicted in the leading figure. The various activities that are performed in the normal confinements of the system are shown. The following activities are performed in the 'normal operation phase .

□ The user registers himself on to the system by entering his details on to the pseudonym manager, which is nothing but a system that contains the details of all the registered clients that are allowed access to the system. Note that, at this stage the system does not have the intelligence nor the functionality to be able to differentiate between a genuine and a compromised user .

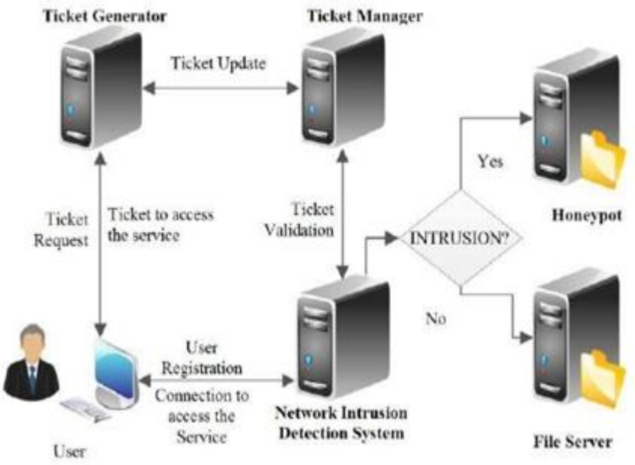


□ 'The system setup' call is done between the pseudonym manager and a 'Nymble manager' .The nymble manager refers to the entity that handles the node detection in the system. This can be seen as the architecture responsible for detecting intrusions in the given system. The nodes here refer to each clients and its attributes are the details of the client supplied by the pseudonym manager. The nymble manager performs matching of the parameters that have been described as 'critical ' before establishing the server and decides whether the given user is genuine or not.

□Server :- The system server that hosts the architecture. Its primary function is to allow connection between the client and the system .

□User:- Refers to the entity that is currently operating the system. The given entity may be genuine or the attacker may have taken the role of a user to enable entry in to the systems architecture.

•STAGE 2:- DETAILED DESIGN



The detailed design of the architecture is represented in the already
mentioned diagram. The diagram represents a table of activities that are crucial to the design phase.

•*TICKET GENERATOR* :- The ticket generator is the entity of the nymble manager that supports the authentication process. Each user is given a ***TcktNO*** upon successful registration on the system. The TcktNO is passed on to the user's email address that is entered as a mandatory field while registering via the Pseudonym manager . Upon login into the system by the new user the TcktNO is compared with the one already present in the *Ticket Manager*. Only upon successful matching of the keys only is the user allowed to enter into the system. In this regard both the 'Ticket Generator ' as well as the 'Ticket Manager' can be both thought of as a concatenated system .

•*Network Intrusion Detection System(NIDS)* :- This architecture performs the actual detection of intrusions in the system. The NIDS
performs the intrusion detection in the following manner :-

□First the NIDS architecture acquires the 'critical ' parameter from the ticket manager system. The critical parameter could be a unique key that was agreed upon by the client and the service provider before the client registered on the system. Let the unique/critical parameter be labeled as ***KeyCrit.*** Upon receiving the KeyCrit, the NIDS asks the user to authenticate himself every time a certain action is performed. This authentication involves matching the KeyCrit value with the value supplied by the user.

□The next step is a decision making process. As depicted in the figure the NIDS makes the decision whether or not a intrusion has taken place by the matching of parameters supplied by the user. Upon intrusion detection the NIDS transfers the currently active user into the 'Honeypot' system. If an intrusion has not taken place , then the user is given undivided access to the file server.

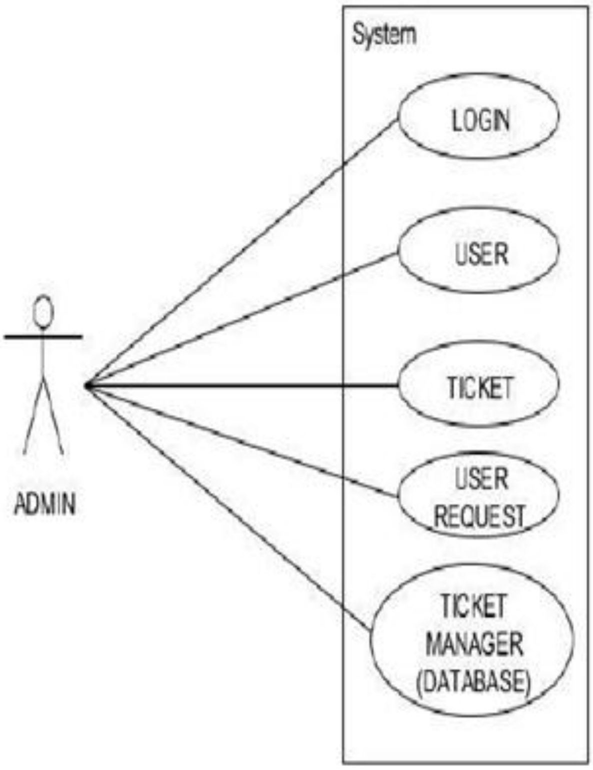
•*Honeypot* :- This refers to the fake environment that is meant to contain the compromised user/attacker. The environment includes data that does not have any relevant significance to the data currently being scrutinized . Implementing the Honeypot can be done by just establishing an additional database that can hold at least 3/4th of the normal capacity of the primary database.

•File Server :- This refers to the database that has the actual data that is present in the organization. Upon successful authentication by the NIDS , a genuine user is directed to the Fileserver where he/she is granted unfiltered access to the data in the organization .

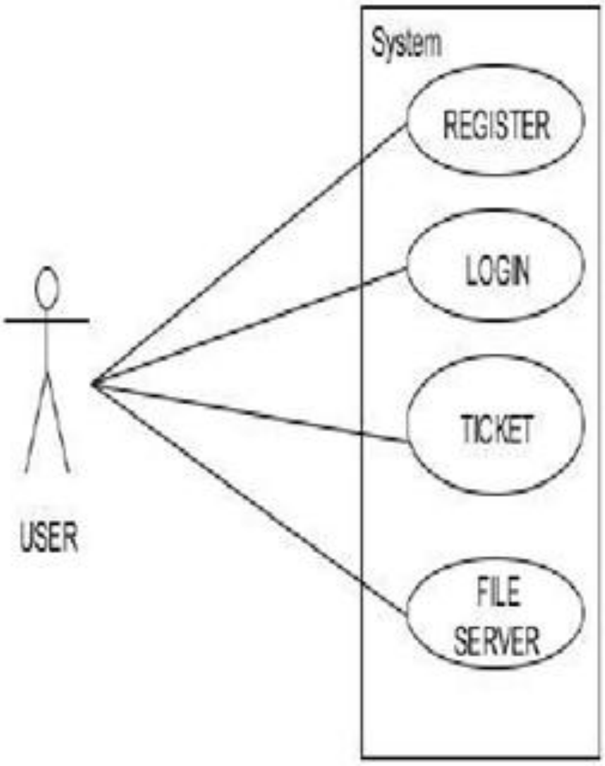
•STAGE 3 :- USAGE SCENARIOS

The detailed usage of the system could involve a multiple number of scenarios, each unique in their functionality and constraints. The design paper should include such scenarios that effectively describe the normal operation of the 'Honeypot Structure'.
The following briefly explains different system scenarios :-

USE CASE SCENARIO :- A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the case and will often be accompanied by other types of diagrams as well.

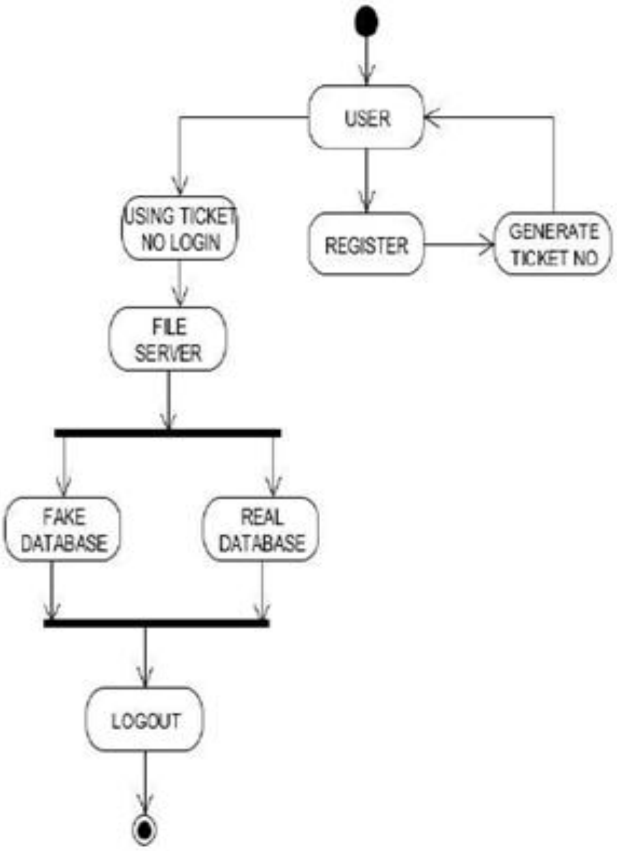


Use case- Accessing stored data in data box
Primary actor- Admin, user
Goal in context-
to retrieve the data files stored in the database through the personal
user account
Preconditions-
This system is equipped with a MySQL database and to recognise user functions
Trigger- When user interacts with the system.
Scenario Specific Roles
Admin:-
•Observes the admin panel
• Account activation only if the user's details are authentic
•Update or delete the files
•View user functions and approve the requests
•Account revoke
•Manage the creation of member groups
User
•login to the system and if a new user then register
•Wait for authentication
•use private key for verification confirmation. •joining a given member group.
•access the member files in the database
• delete account
Exceptions-
•Password is incorrect.
•Password may not reach the user in latency time due to congestion. •There may be no result present in the database for users' search.
Priority- Essential, must be implemented
When available- Second increment
Frequency of use- many times



Channel to actor- view main program interface

Activity Diagrams :- Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language. activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.



As depicted in the diagram, the activity diagram represents the best abstraction of all the activities performed by the participating entities
in the 'Honeypot Structure' . The various activities involved are :-

•The user registers himself onto the server

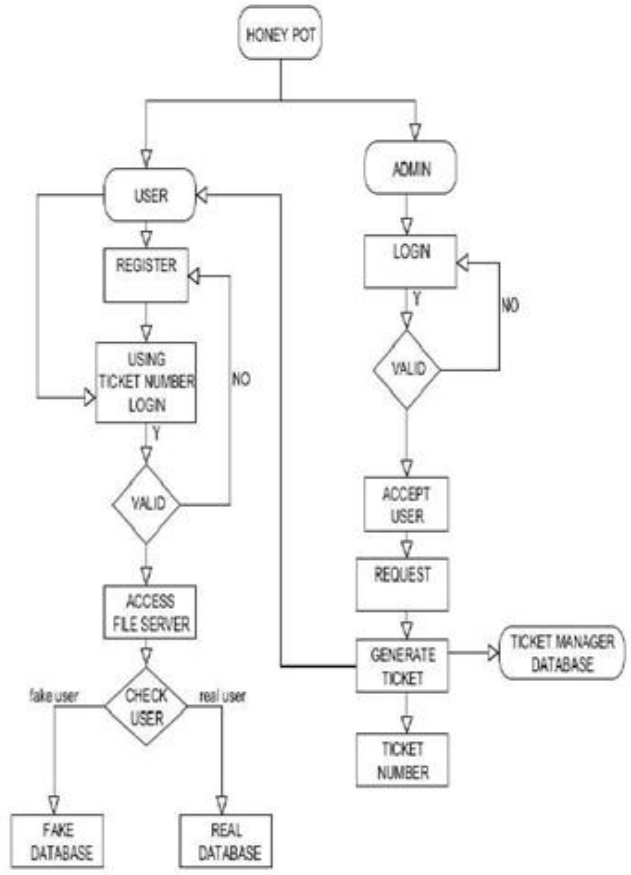
•The server then addresses the Ticket Generator in-order to obtain a
ticket for the given session

•The user is redirected onto the login screen along with a prompt to enter the ticket obtained as a result of registration.

•Depending on the authenticity of the entered ticket value , the user
is either directed to the "Real database" or a "Fake database "

□Flow Chart Of Operations:-

The system can be accurately described using a flow chart which pictorially describes the entire scenario of operations that can be undertaken when an entity accesses the system. A flowchart is a type
of diagram that represents an algorithm ,workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem .Flowcharts are used in analyzing, designing, documenting or managing a process or
program in various fields



All the architectures described about in STAGE 2 of system design are present in the flow chart of operations, and rightly so , since the working of the system can only be understood if the operations of all the architectures present in the system are listed.

V. CONCLUSION .

This paper represents ideas that help a given organization in better securing their critical data in view of data oriented attacks . The paper highlights the features of an alternate 'Honeypot' system that performs authentication of various 'critical ' parameters before enabling access to the give file server.

REFERENCES

[1] Identify Misbehaving users, International Journal of Innovative
[2] Science and Modern Engineering (IJISME),ISSN: 2319-6386,
[3] Volume-2, Issue-6, May 2014
[4] Anonymous P2P:
[5] http://wired4geeks.wordpress.com/2011/01/07/anonymous-p2p/
[6] Jiang Zhen, Zhenxiang Liu, "New Honeypot System and its Application in Security of Employment Net Work" IEEE
[7] Symposium on Robotics and Application, 2012
[8] Amandeep Singh, Satwinder Singh, Saab Singh M.Tech CE & Punjabi University Punjab, India"Review of Implementing a Working Honeypot System", ijarcsse,Volume 3, Issue 6, June 2013