iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** VI **Month of publication:** June 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

FTP Based Honeypot Implementation Model for Network Security

Diksha Dewanagn[1], Amit Kumar Dewangan[2]
[1]M.Tech Scholar, Department of Computer Science and Engineering
[2]Assistant Professor, Department of Computer Science and Engineering

Abstract— In this paper, we have presented Network security using Honeypot technique that pretends to have one or more network vulnerabilities that a blackhat is looking for. But actually it does not have those vulnerabilities. It does so just to deceive the intruder, by stealthily monitoring the network. Honeypots are emerging technology and have got lot of attention of late. Our system uses the advantages of Honeypot for implementing an intrusion detection system. Honeyd is a powerful tool which can simulate even complex networks very easily. The system is basically a java based Intrusion detection system which can work along honeyd system. This is a basic pattern based IDS which uses snort rule base and some other rules to detect intrusion. It is a feature rich data analyzer which can detect intrusion. Function of the system is quite simple it reads the data logged by the honeypot system and looks for intrusion pattern of rule base into the packets. We have tested our data analyzer on many publicly available intrusion data. These data are in binary/pcap form. Accuracy of the system depends on the ruleset better the ruleset more will be the accuracy. We have used the ruleset of the snort and some other open source. If we include more such ruleset it will increase the accuracy of the system. The primary goal was to enhance the capability of the the honeypot. We have tested our tool on publicly available intrusion data which shows that only certain kind of attacks form the most of these attacks.
Keywords— File Transfer Protocol, Honeyd, Snort, Intrusion Detection System

I. INTRODUCTION

According to Lance Spitzner, founder of the Honeynet Project, a honeypot is a system designed to learn how “black-hats” probe for and exploit weaknesses in an IT system [1]. It can also be defined as “an information system resource whose value lies in unauthorized or illicit use of that resource” [2]. In other words, a honeypot is a decoy, put out on a network as bait to lure attackers. Honeypots are typically virtual machines, designed to emulate real machines, feigning or creating the appearance of running full services and applications, with open ports that might be found on a typical system or server on a network. A honeypot works by fooling attackers into believing it is a legitimate system; they attack the system without knowing that they are being observed covertly. When an attacker attempts to compromise a honeypot, attack-related information, such as the IP address of the attacker, will be collected. This activity done by the attacker provides valuable information and analysis on attacking techniques, allowing system administrators to “trace back” to the source of attack if required.

Honeypots can be used for production or research purposes. A production honeypot is used for risk mitigation. Most production honeypots are emulations of specific operating systems or services. They help to protect a network and systems against attacks generated by automated tools used to randomly look for and take over vulnerable systems. By running a production honeypot, the scanning process from these attack tools can be slowed right down, thereby wasting their time. Some production honeypots can even shut down attacks altogether by, for example, sending the attackers an acknowledgement packet with a window size of zero. This puts the attack into a “wait” status in which it could only send data when the window size increases. In this way, production honeypots are often used as reconnaissance or deterrence tools. Research honeypots are real operating systems and services that attackers can interact with, and therefore involve higher risk. They collect extensive information and intelligence on new attack techniques and methods, and hence provide a more accurate picture of the types of attacks being perpetrated. They also provide improved attack prevention, detection and reaction information, drawn from the log files and other information captured in the process. In general, honeypot research institutions such as universities and military departments will run research honeypots to gather intelligence on new attack methods. Some of the research results are published for the benefit of the whole community.[3]

At the core of all Honeypots are the IDS which are the monitoring system of network traffic. The Honeypot is not actually one specific type of software or hardware but is in fact a collection of different elements that make it up.

IDS
Mimic system (Windows Server)
Packet Analyser

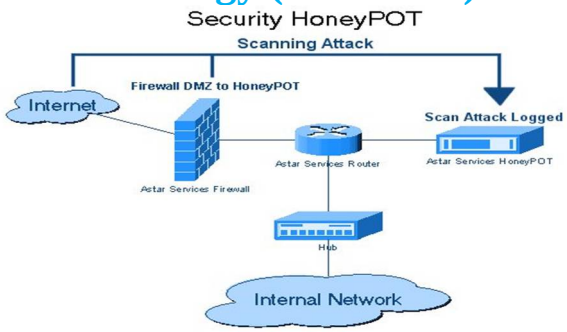


FIGURE 1.1: HONEYPOT NETWORK

Data Storage of analysed packets

A true Honeypot system will also have the ability to run as a virtual (Low Interaction) or real (High Interaction) system which can mean the difference between cost and protection.[4]

A. *What Is AN ID?*

At the core of this project is the setup and analysis of an IDS, so what is it?

An IDS is the core of the Honeypot, but an IDS system will not attract any traffic like a honeypot. While the Honeypot mimics a system, this one will not, and can in fact be implemented as a firewall with the correct equipment, which means that it is more of a pure monitoring system. An IDS will monitor the traffic as it occurs. There are two main types of IDS, the Network IDS (NIDS) and the Host IDs (HIDS). The NIDS would be the most common one, sitting at the head of the network reading all incoming traffic before it hits the firewall, however the NIDS can also be the firewall preventing certain types of traffic through regardless if it is classed as safe or not. This can then be tied in with other systems to act as a first stage alert system to turn on any preventative measures that can be used. [5]

The HIDS will monitor only one machine on a network or monitor traffic on a network going to and from one specific machine (like a server), usually this will be attached to a Server of some kind or other vital system, to ensure that all traffic is strict and above board. However you will never have a HIDS on its own, and will always be tied in with a NIDS. Combined these can form their own version of a Honeynet called a Sensor Server system, where the Server would be the NIDS and the Sensors are the HIDS that relay information back to the server with their ID number. While The Honeypot is how these types of systems are classed, there are very few actually Honeypots in commercial use due to their high risk of attack. Rather the Honeypot is used on more educational or experimental networks. The IDS would normally be desired do to their expandability or systems and implementation or external hardware such as the location and area based sensors, which can then tie into backup systems and emergency systems such as UPS‟s and emergency shutdowns or network isolation switches.

II. RELATED WORK

A. Different intrusion detection techniques

Now that we have examined the two basic types of IDS (HIDS and NIDS) and why they should be used together, we can investigate how they go about doing their job. For each of the two types, there are four basic techniques used to detect intruders: anomaly detection, misuse detection (signature detection), target monitoring and stealth probes.

1) *Anomaly Detection:* Designed to uncover abnormal patterns of behavior, the IDS establishes a baseline of normal usage patterns, and anything that widely deviates from it gets flagged as a possible intrusion. What is considered to be an anomaly can vary, but normally, we think as an anomaly any incident that occurs on frequency greater than or less than two standard deviations from the statistical norm. It identifies anomalies as deviations from “normal” behavior and automatically detects any deviation from it, flagging the latter as suspect. Thus these techniques identify new types of intrusion as deviations from normal usage. It is an extremely powerful and novel tool but a potential drawback is the high false alarm rate, i.e. previously unseen (yet legitimate) system behaviors may also be recognized as anomalies, and hence flagged as potential intrusions. If a user in the graphics department suddenly starts accessing accounting programs or compiling code, the system can properly alert its administrators.

2) *Misuse detection (Signature detection):* Here each instance in a data set is labelled as “normal” or “intrusive” and a learning algorithm is trained over the labelled data. These techniques are able to automatically retrain intrusion detection models on different input data that include new types of attacks; as long as they have been labelled appropriately. Unlike signature-based IDS, models of misuse are created automatically and can be more sophisticated and precise than manually created signatures. They have high degree of accuracy in detecting known attacks and their variants. Their disadvantage is that they cannot detect unknown intrusions and they rely on signatures extracted by human experts. This method uses specifically known patterns of unauthorized behavior to predict and detect subsequent similar attempts. These specific patterns are called signatures. For hostbased intrusion detection, one example of a signature is "three failed logins." For network intrusion detection, a signature can be as simple as a specific pattern that matches a portion of a network packet. For instance, packet content signatures and/or header content signatures can indicate unauthorized actions, such as improper FTP initiation. The occurrence of a signature might not signify an actual attempted unauthorized access. Depending on the robustness and seriousness of a signature that is triggered, some alarm, response, or notification should be sent to the proper authorities.
3) *Target Monitoring:* these systems do not actively search for anomalies or misuse, but instead look for the modification of specified files. This is more of a corrective control, designed to uncover an unauthorized action after it occurs in order to reverse it. One way to check for the covert editing of files is by computing a cryptographic hash beforehand and comparing this to new hashes of the file at regular intervals. This type of system is the easiest to implement, because it does not require constant monitoring by the administrator. Integrity checksum hashes can be computed at whatever intervals you wish, and on either all files or just the mission/system critical files.
4) *Stealth Probes:* this technique attempts to detect any attackers that choose to carry out their mission over prolonged periods of time. Attackers, for example, will check for system vulnerabilities and open ports over a two-month period, and wait another two months to actually launch the attacks. Stealth probes collect a wide-variety of data throughout the system, checking for any methodical attacks over a long period of time. They take a wide-area sampling and attempt to discover any correlating attacks. In effect, this method combines anomaly detection and misuse detection in an attempt to uncover suspicious activity [8].

III. PROPOSED METHODOLOGY

A. System Components & Design

There are two major components in this system

1) *Data Accumulator*
   - *a)* Main purpose of this part is to attract intruders.
   - *b)* Honeyd sets up a virtual network to attract the intruders.
   - *c)* Data Accumulator then collects anything thrown at honeypot.
2) *Data Analyzer*
   - *a)* This part of the system looks for suspicious traffic from intruders.
   - *b)* If the data from intruders contains suspicious pattern the system raises alarm and saves the data separately.

Honeypot Systems are decoy servers or systems setup to gather information regarding an attacker or intruder into your system. It is important to remember that Honeypots do not replace other traditional Internet security systems; they are an additional level or system. Honeypots can be setup inside, outside or in the DMZ of a firewall design or even in all of the locations although they are most often deployed inside of a firewall for control purposes. In a sense, they are variants of standard Intruder Detection Systems (IDS) but with more of a focus on information gathering and deception.

Honeypot to be easier prey for intruders than true production systems but with minor system modifications so that their activity can be logged of traced. The general thought is that once an intruder breaks into a system, they will come back for subsequent visits. During these subsequent visits, additional information can be gathered and additional attempts at file, security and system access on the Honey can be monitored and saved. An example of a Honeypot systems installed in a traditional Internet security design:

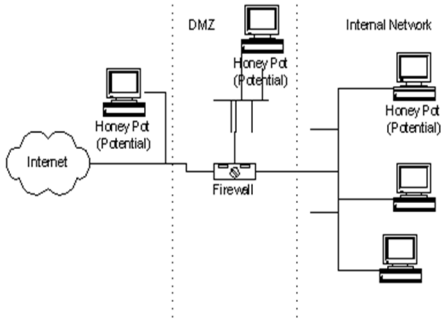


FIGURE 1.2 HONEYPOT SYSTEMS

Generally, there are two popular reasons or goals behind setting up a Honeypot:

Learn how intruders probe and attempt to gain access to your systems. The general idea is that since a record of the intruder's activities is kept, you can gain insight into attack methodologies to better protect your real production systems.

Gather forensic information required to aid in the apprehension or prosecution of intruders. This is the sort of information often needed to provide law enforcement officials with the details needed to prosecute.

The common line of thought in setting up Honeypot systems is that it is acceptable to use lies or deception when dealing with intruders. What this means to you when setting up a Honeypot is that certain goals have to be considered. Those goals are:

The Honeypot system should appear as generic as possible. If you are deploying a Microsoft NT based system, it should appear to the potential intruder that the system has not been modified or they may disconnect before much information is collected.

You need to be careful in what traffic you allow the intruder to send back out to the Internet for you don't want to become a launch point for attacks against other entities on the Internet. (One of the reasons for installing a Honeypot inside of the firewall!)

You will want to make your Honey Pot an interesting site by placing "Dummy" information or make it appear as though the intruder has found an "Intranet" server, etc. Expect to spend some time making your Honey Pot appear legitimate so that intruders will spend enough time investigating and perusing the system so that you are able to gather as much forensic information as possible.

Some caveats exist that should be considered when implementing a Honey pot system. Some of the more important are:

The first caveat is the consideration that if the information gathered from a Honey Pot system is used for prosecution purposes, it may or may not be deemed admissible in court. While information regarding this issue is difficult to come by, having been hired as an expert witness for forensic data recovery purposes, I have serious reservations regarding whether or not all courts will accept this as evidence or if non-technical juries are able to understand the legitimacy of it as evidence.

The second main caveat for consideration is whether hacking organizations will rally against an organization that has set "traps" and make them a public target for other hackers. Examples of this sort of activity can be found easily on any of the popular hacker's sites or their publications.

IV. SIMULATION OF PROPOSED METHODOLOGY

A. Data Accumulator

1) It is basically a virtual network set up with the help of honeyd.

2) Setting up honeyd needs other softwares as well.

3) Software tools required are

a) Nmap

b) Libdnet

c) Libdevent

d) Arpd

e) Hmap

B. Data Accumulators functionality:

1) We have changed the basic behaviour of honeyd little bit, now it logs the packet in tcp dump format.

2) It logs in honeyd format as well.
3) It also creates a virtual network in which various fake http servers run.
4) It attracts intruder to attack the fake servers.

C. *Data Analyser functionality:*

1) This part of the system is capable of analysing data logged by honeypot.
2) We have developed a libpcap based application which scans data packets.
3) This application looks for some predefined pattern to find out the suspicious http packets.
4) It also separates the data packets which it finds suspicious.

Following steps are to be performed for Installation of system:

STEP 1: Installation of required libraries

For example installing libdnet-1.10 is as simple as:

```
# cd /playground/Honeyd
# gzip -d libdnet-1.10.tar.gz
# tar -xvf libdnet-1.10.tar
# cd libdnet-1.10
# ./configure
  # make
  # make install
```

(Same steps will be taken for libevent1.1 and libpcap)

STEP 2: Verification by regression test

```
#make verify
```

STEP 3: Configure dynamic linker

```
#ldconfig /usr/local/lib
```

STEP 4: Install ARPD

STEP 5: Install Honeyd

STEP 6: Creating Honeyd configuration file

it is this file that gives a virtual Honeypot a particular behaviour.

```
create default
set default default tcp action block
set default default udp action block
set template uptime 1234567
add template tcp port 135 open
add template tcp port 139 open
add template tcp port 445 open
bind 132.140.1.123 template
```

This will create a virtual system running MS windows XP SP2 on IP address 132.140.1.123.(Place it on /usr/local/share/honeyd/honeyd.conf)

STEP 7: Setting up firewall rules to accept packets for the IP addresses

```
$iptables -A INPUT -d 132.140.1.123 -j ACCEPT
$iptables -A INPUT -m state --state RELATED, ESTABLISHED -j ACCEPT
```

STEP 8: Start arpd

```
#arpd –i wlan0 “132.140.1.123”
```

V. CONCLUSION

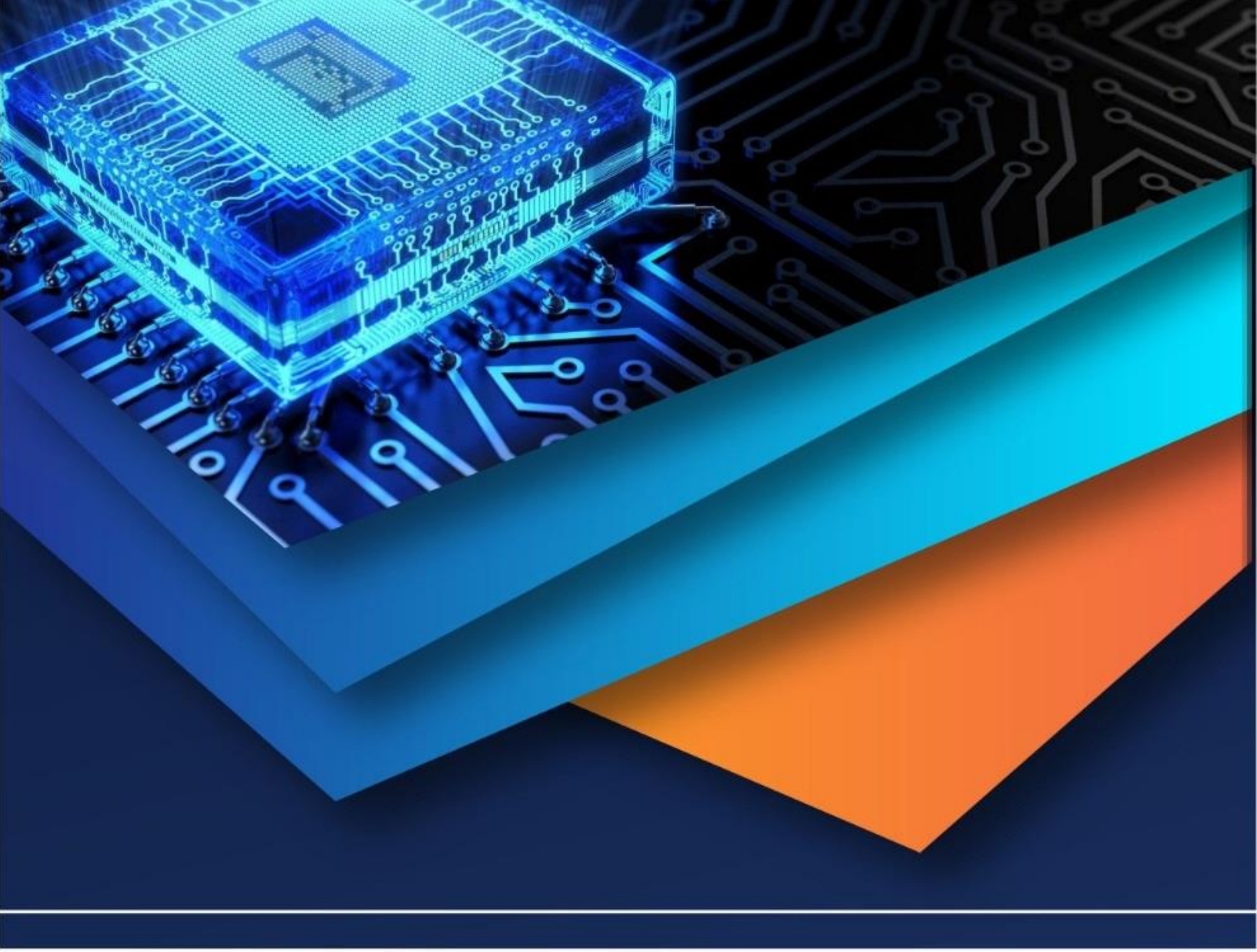
FTP protocol is useful for sending large files. The proposed system is useful to detect the intruder pattern using FTP (File Transfer Protocol). We have simulated the result using honeyd tool; the accuracy of system is 80% which is better than existing approaches. System performance can be improved by increase the dataset of FTP based database.

VI. ACKNOWLEDGMENT

I would like to thank my research guide Mr. Amit Kumar Dewangan for his valuable support during research work. I would like to thank all respected faculty members of Department of Computer Science and Engineering, Dr. C V Raman Institute of Science and Technology..

REFERENCES

[1] Awad Johnny ,Derdemezis Andreas , "Honeynets And Honeypots Implementation of High Interaction Honeynet Testbed for Educational and Research Purposes" ,MsITT Thesis 2005.
[2] Bhumika, Vivek Sharma, " Design & Implementation of Honeyd to Simulate Virtual Honeypots" , IOSR Journal of Computer Engineering (IOSRJCE),Vol. 3,PP. 28-34, Issue 1, 2012.
[3] Elmehdi Bendriss,Boubker Regragui, "Honeypot Based Intrusion Management System: From a Passive Architecture to an IPS System ", Journal of Theoretical and Applied Information Technology , Vol. 47 ,No.2 ,PP. 792-797, 2013.
[4] Ashish Girdhar, Sanmeet Kaur , "Comparative Study of Different Honeypots System", International Journal of Engineering Research and Development, Vol. 2, PP. 23-27, 2012.
[5] Abdulrazaq Almutairi , David Parish, Raphael Phan, "Survey of High Interaction Honeypot Tools: Merits and Shortcomings",ISBN: 978-1-902560-26-7 ,2012 .
[6] Balaji Darapareddy, Vijayadeep Gummadi, "An Advanced Honeypot System for Efficient Capture and Analysis of Network Attack Traffic", International Journal of Engineering Trends and Technology, Vol. 3,Issue 5, PP. 616-621,2012 .
[7] Juan M. Estevez-Tapiador , Pedro Garcia-Teodoro, Jesus E. Dıaz-Verdejo, "Measuring normality in HTTP traffic for anomaly-based intrusion detection" ,Elsevier Computer Networks, Vol. 45, PP. 175–193, 2004.
[8] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang and S. Zhou, "Specification-based Anomaly Detection : A New Approach for Detecting Network Intrusions",ACM 1-58113-612-9/02/0011, PP. 18-22 , 2002.
[9] Talasila Vamsidhar, Reddyboina Ashok and RayalaVenkat, "Intrusion Detection System for Web Application With Attack Classification" , Journal of Global Research in Computer Science ,Vol. 3, No. 12, PP. 44-50, 2012.
[10] Luisse Margarette A. Macasaet, "A Network Intrusion Testbed through Honeypots " ,CMSC 190 Special Problem, Institute Of Computer Science , 2012.
[11] S R Tandan, Niharika Vaishnav, "A Bird's Eye View of Anti-Phishing Techniques for Classification of Phishing E-Mails" International Journal for Research in Applied Science & Engineering, Technology (IJRASET) (2015)
[12] Gaurav Kumar Rai, Rohit Miri, , S R Tandan "Enhanced Security Technique in WAP & WEP Based Wireless (Wi-Fi) Network for the Protection against Unauthorized Users international journal for advance research in engineering and technology (ijaet), (2014)
[13] Niharika Vaishnav, S R Tandan "Development of Anti Phishing Model for Classification of Phishing E-Mail"International Journal of Advanced Research in Computer and Communication Engineering, (2015).
[14] G Singhal, S R Tandan, R Miri "IAA (Internet access account) based security modal for detection and prevention of cyber crime" International Journal of Engineering Research and Technology, (2013)
[15] Shikha Gupta, S R Tandan, R Miri "Modeling of Election Algorithm for Coordinator Selection Using Neuro Fuzzy Approach in Distributed Computing Environment" International Journal of Engineering Research and Technology, (IJERT), (2013)
[16] K Lahre, S R Tandan, R Miri "Implementation of Improved Distributed Wireless Channel Allocation Algorithm for Mobile Computing" Wireless Communication-CIIT-IJ, (2011)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY