Advance Trends in Network Security with Honeypot and its Comparative Study with other Techniques

Aaditya Jain, Dr. Bala Buksh

M.tech (CS & E), Professor (CS & E)
R. N. Modi Engineering College, Kota, Rajasthan, India

Abstract— *Achieving network system security is one of the most popular and fastest Information Technologies in organizations. Tools for network security deal with the capture, recording and analysis of network events in order to discover evidential information about the source of security attacks. Advanced decoy based technology called Honeypot has a huge potential for the security community and can achieve several goals of other security technologies. This paper discusses about the honeypot technology with its classification based on various factors. Paper also throws light on some new types of honeypots with recently proposed models based on it. At last this paper provides comparative study with other network security tools.*

Keywords— *Honeypots, Ssh, Botnet, Polymorphic worm, Ids, and Firewall.*

I. INTRODUCTION

The number of people connecting to the internet is increasing very rapidly but the risks involved and malicious intrusions are also increasing day by day. Exploitation of computer networks is getting more common. Protection of information availability, its access and data integrity are the basic security characteristics of information sources. Any disruption of these properties would result in system intrusion and the related security risk. There are a numerous ways in which researchers and developers can work to protect the software that they write. Some are proactive, like code reviews and regression testing, while others are reactive, like new vulnerabilities are used to exploit browsers. One class of tool that can take on aspects of both in terms of network is honeypots.

Honeypot is a resource that is used in the area of security whose value is being attacked or compromised. Its primary purpose is not to be an ambush for black hat community but the focus lies on silent collection of as much information as possible about their pattern, used programs, purpose of attack. There are several more possibilities with honeypot like divert black hats from productive systems or catch black hats while conducting an attack.

II. HONEYPOT DEFINITION

Honeypot uses beware technology, is an elective means to save the network and search in order to design a tough system on a descriptive environment. Honeypot generates an alarm to the administrator of the system while attacker attacks the system [1]. Lance spitzner definition of such system [2].

"A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource".

By definition the value of honeypot is derive from the threats using them i.e. if black hats do not interact with honeypot then it has little value. A honeypot works by fooling attackers into believing that it is a legitimate system [3]. The attackers attack the system without knowing that they are being observed. When an attacker attempts to compromise a honeypot, its attack related information such as the IP address of the attacker, will be collected.

III. CLASSIFICATION OF HONEYPOT

A honeypot is a deception trap, designed to entice an attacker into attempting to compromise the information systems in an organisation. If deployed correctly, a honeypot can serve as an early-warning and advanced security surveillance tool, minimising the risks from attacks on IT systems and networks [3]. Honeypot comes in many shapes and sizes, making them difficult to get a grasp of [5].

For better understanding honeypot can be classified with the help of many factors:

A. Based on Interaction with intruders

The level of involvement does measure the degree an attacker can interact with the operating system.

1) Low Interaction Honeypot: Low interaction honeypots are used to detect the hackers and deceive them by emulating the operating system services and port services on the host operating system. The interaction with other hosts is limited so they have limited working capabilities and attacker can easily find finger prints but its simplicity and low risk are its advantages. Examples of low interaction honeypot are Honeyd, Spector, KFsensor and Dionaea.

2) Medium Interaction Honeypot: These honeypots lies between low interaction and high-interaction honeypots and do not provide OS access to attacker like low interaction honeypot, but chances to be probed are more than low interaction honeypot. These honeypots are more capable than low-interaction honeypot but involve high risk compare to it. Examples of medium interaction honeypot are Napenthes, Dioneae, and honeytrap.

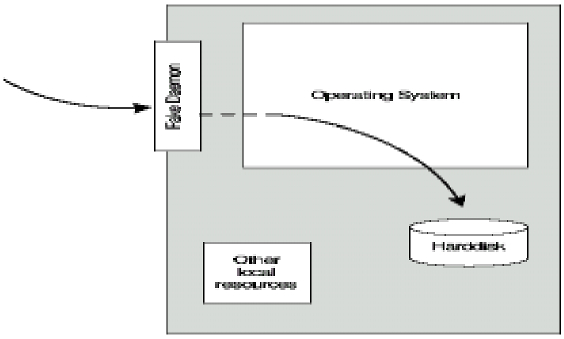

Fig. 1 Low interaction honeypot

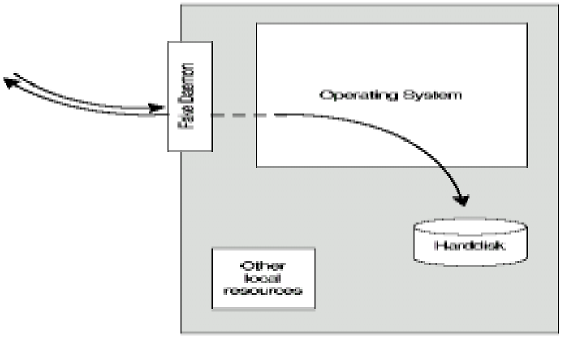

Fig. 2 Medium interaction honeypot

3) High Interaction Honeypot: These are the most sophisticated honeypots [5]. These are difficult to design and implement because they involve real operating systems and applications, like a real FTP server will be built if the aim is to collect information about attacks on a particular FTP server or service. By allowing the attackers to interact with real systems more data can be captured from attacker's activities but associated risk is very high because it uses real os. Example of High interaction honeypot is Honeywall.

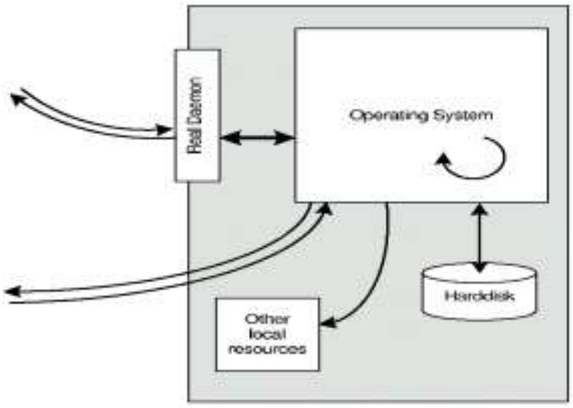

Fig. 3 High interaction honeypot

TABLE I

COMPARISON BETWEEN LOW INTERACTION AND HIGH INTERACTION HONEYPOTS

Low Interaction Honeypot	High Interaction Honeypot
It does not provide operating system access.	It provide operating system access with no restrictions.
Easy design and low maintenance time.	Complex design with very high maintenance time.
It can capture limited information.	It can capture more information.
Minimum risk, as provided only services.	More risk, as provided real system for attackers.

B. Based on Purpose

Honeypot can be classified according to their purpose. They add value to security and reduce the organization's overall risk [5].

1) Production Honeypot: Production Honeypots are systems that help mitigate risk in the organization or environment. They provide specific value to securing systems and networks by preventing, detecting and responding attacker's activity. It is placed along with production servers in the production network.

The idea behind production honeypots is to emulate real production systems so that attackers spend time and resource attacking on them, also learns the way they exploit vulnerabilities in production environment [6]. These types of honeypots are easy to use and have limitation of capturing limited information and mostly used in companies and corporations.

2) Research Honeypot: Research honeypots are basically used to attain information about the new ways of attacks, viruses, worms which are not detected by IDS. Research Honeypots give a platform to study cyber threats and fill the lack of information on the enemy. It is used for research purpose.

These honeypots are difficult to maintain and complex in architecture, it provide brief information about the blackhats and their attacking policies.

Mostly in educational entities, military or government organizations, these kinds of honeypots are used to gather information about motives and new tactics about the black hat community.

TABLE II

COMPARISON BETWEEN PRODUCTION AND RESEARCH HONEYPOTS

Production Honeypot	Research Honeypot
Primarily used in organizations for protecting their internal IT infrastructure.	Primary goal is to provide real live sight of how attack happen.
Implementation and deployment is relatively easy.	Comparatively complex in nature.
It can capture limited information.	It can capture more information as possible as.
Comparatively minimum risk.	More risk because of its high interaction with intruders.

C. *Based on Physical Presence in the Network*

By Xuxian Jiang et al. honeypot can be classified as hardware based and software emulation honeypot [7].

1) Hardware Based Honeypot: Hardware based honeypots are servers, switches or routers that have been partially disabled and made attractive with commonly known misconfigurations. These honeypots placed inside the network and look real to outsiders.

2) Software Emulation Honeypot: Software emulation honeypots are elaborate deception programs that mimic real Linux or other servers and can run on machines as low power as a 233-MHz PC. Intruders deal with it and do not occupy the control of real system. If in any case intruder find out that it deal with mimic system, then the box on which it's running should be highly secure so that he could not do any harmful activity.

TABLE III

COMPARISON BETWEEN HARDWARE BASED AND SOFTWARE EMULATION HONEYPOTS

Hardware Based	Software Emulation
These honeypots are placed on the internal network of organization so look real to outsiders.	They mimic real servers.
It used Servers, Switches, and routers to meet their requirements.	It can run on machines as low power as a 233-MHz PC to meet their requirements.
If intruder control this system then it will use to launch new attack on other system.	If intruder know that he is trapped then he could not do anything only leave from it.
High risk.	Comparatively low risk.

IV. ADVANCE HONEYPOTS PROPOSED IN RECENT YEARS

Honeypots play a great role in the area of network security. Honeypots have evolved in diverse directions to cope with various new security threats against not only security defenders but also novice users in the Internet today. To cope with the recent changes in the network security new types of honeypots are introduces, they act against the new vulnerable activities.

Portokalidis et al. proposed a honeypot called "Argos" [8]. It automates monitoring, detecting, and generating signatures of new unknown malware for intrusion detections. It is designed to slow down dissemination of new, and thus unknown, malware, such as worms, viruses, and bug exploits. When Argos detects vulnerable data, it also dynamically inserts assembly codes, called "shellcode", into the process to extract detailed information about the process so that the process is slowed down or trapped in an infinite loop to minimize its harm.

Alosefer and Rana proposed "Honeyware" [9]. It is low interaction client-side honeypot for detecting malicious web servers. Alosefer tested Honeyware against 94 URL's he collected in advance in which 84 malicious and 10 benign. Honeyware detected 83 of the malicious URL's. Since Honeyware is a low interaction honeypot, the data collected by it must be processed by an external processing engine, which takes time.

Adachi and Oyama proposed "BitSaucer" [10]. It is a hybrid honeypot i.e. provide the facility of both low interaction honeypot to achieve less resource requirements and high interaction honeypot to emulate full responses.

Zhuge et al. proposed a new honeypot, called "HoneyBow" [11], to automatically detect and capture malware, such as viruses and worms, without requiring human security experts manually investigating output data from honeypots. HoneyBow detects the modifications of files by comparing their initial MD5 hash after it intentionally lets malware modify its files. When any modification is detected, the process that made the modification is captured as malware and its component MmFetcher restores the initial copy of the files. Another component, MmWatcher, monitors system calls that perform file creation and modification, which triggers intrusion detection. Finally, MmHunter monitors code being executed like a debugger to detect malware's suspicious activities.

Anagnostakis et al. proposed "Shadow Honeypots" [12]. They are real production network applications but contain honeypot codes embedded in it. They are focused on the trade-off problem like false positive and false negative in high interaction honeypot. All incoming requests to a server running the shadow honeypot will be executed just as if they were executed by a production server. If the shadow honeypot determines a request to be innocent, it forwards the request to the production server.

LaBrea is another kind of honeypot [13], designed to slow down or stop attacks by acting as a sticky honeypot to detect and trap worms and other malicious codes. It can run on both Windows and UNIX.

Vinu V. Das proposed a solution to mitigate denial of service attacks by hiding production servers behind an access gateway, called “Active Server (AS)” [14]. Each AS authenticates its clients and once a client is authenticated, a path is opened between the client and a server. If an AS does not authenticate a client, it behaves as a honeypot, trapping the client there. If a client has access to multiple ASes, the client can be authenticated by any AS. Honeypots trap attackers, which prevented, reduced, and delayed the impacts from the DoS attacks.

Niels Provos proposed a low interaction open source honeypot called “Honeyd” [15]. It is a powerful honeypot, and can be run on both UNIX like and Windows platforms. It can monitor unused IPs, simulate operating systems at the TCP/IP stack level, simulate thousands of virtual hosts at the same time, and monitor all UDP and TCP based ports.

Nazario proposed a new type of honeypot called “PhoneyC” [16]. It extends existing honeypots in two directions. The first is to make honeypots active, which means client-side honeypots. The second is the dynamic web content parser to interpret binary dynamic contents, especially client-side scripts, such as JavaScript, VB Script, and even Active-X controls. Integrating the two extensions to web applications, active client-side honeypots become web “clawers” that visit a large number of web servers to automatically detect malicious web servers. As a result, PhoneyC was able to detect many malicious script/control activities during experiments.

Rowe et al. proposed the idea of “Fake Honeypot” [17]. The goal of fake honeypot is to repel attackers from a production network by intentionally exposing themselves to attackers. It look like a real honeypot, but they are not performing any real feature typical honeypots perform. A mathematical model was introduced to maximize the effect of the fake honeypots, using some parameters, such as the probability of a system being a honeypot, the benefit expected by an attacker from compromising a production host, and the cost for compromising a host.

Honey Mole is a tool for the deployment of honeypot farms, or distributed honeypots, and transport network traffic to a central honeypot point where data collection and analysis can be undertaken [18].

V. MODELS BASED ON HONEYPOT SYSTEM

Honeypot can also be used in the various area of information technology to enhance the security policies, some advanced models are proposed.

A. Sophisticated Hybrid Honeypot Model in IDS Security [19]

The proposed model uses a sophisticated hybrid Honeypot with an autonomous feature as an IDS detection mechanism with the aim to minimize failure in detection process with the combination of security tools like Snort IDS, Sebek and Dionaea. The idea is integrated as client-server architecture, consisting of centralized main server and multiple client workstations. Client stations capture malicious code and sent it to the server for analysis and based on analysis server decides whether issue security warning or not. This model aim is to provide early warning against suspicious activities.

*1) **Server Architecture:*** Central server perform multiple functionality at the same time like receive all incoming data, normalize them and at last store it in the database for further use. So server architecture consist three main components, Sebek server: it receives and filters data sources and represents connection to incoming data storing process. Dionaea server: accepts patterns of malicious code that sends the dionaea client part. Verification process: uses hybrid open source system for intrusion detection, receives the amount of data from clients and integrates diversified data formats. Web server interface displays all information about captured attack.

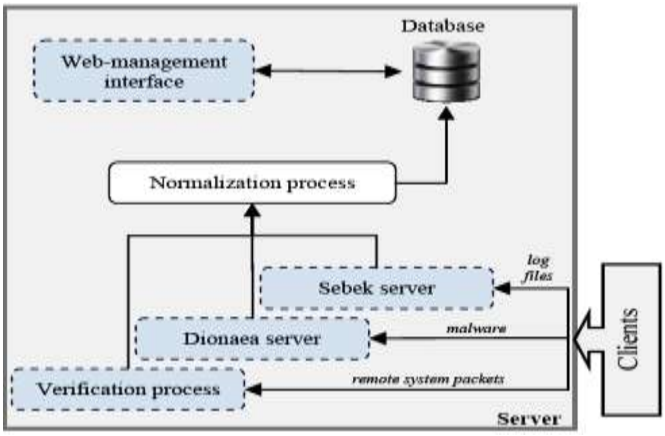

Fig. 4 Server architecture [19]

*2) **Client Architecture:*** Clients aim is to gather information about blackhats activities during attack. Obtained data is send to server for enhancing system security. It also has three components to perform required functionality. Sebek client: records attacker behaviour during interaction with the honeypots in log files. Dionaea client: attracts attackers and captures the patterns of malware by simulating basic system services and vulnerabilities. Snort: monitors and filters packets during detecting intrusions, identifies patterns of separate attacks, information and warning messages.

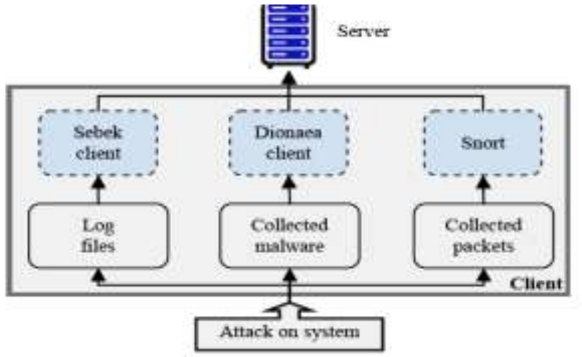

Fig 5 Client architecture [19]

B. Honeypot Based Signature Generation Model & other Techniques against Polymorphic Worm Attacks

1) Proposed model in paper [1] introduce two high interaction honeypots i.e. Honeytrap1 and Honeytrap2 for the purpose of research about attacker activities without inform them. Both of these have multiple levels of physical honeypots and have three layers of software i.e. system software, application software and sebeck client.

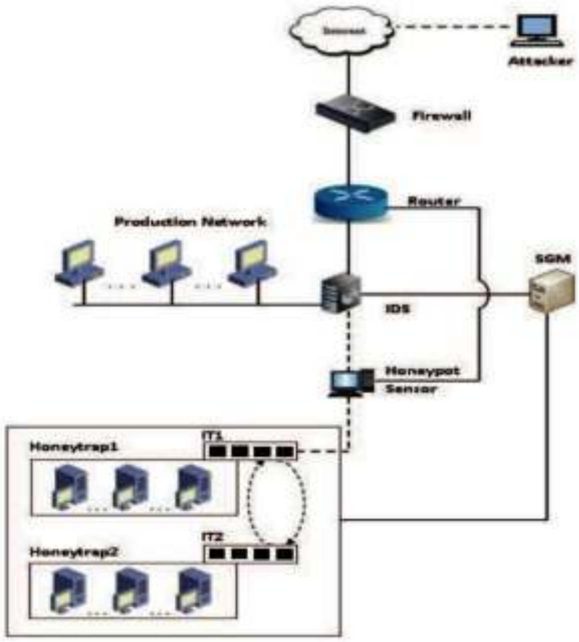

Fig. 6 Architecture of proposed model [1]

Figure shows a separate link between router and honeypot that allow attacker to come in the system but the fact is he could not be able to know IP address and other information.

Honeytrap1 create an outbound connection if it receives any attack and then transfer traffic to Honeytrap2 with the help of internal translator1 (IT1).Working of Honeytrap2 is same as Honeytrap1 and uses IT2 for transfer traffic to Honeytrap1 and try to make a connection to outside systems. This system meets with our requirements as storing enough activities of malicious traffic and reduces the possibilities of denial of services attack. This model have multilayer data storing capability so helps in analysing vulnerable activities correctly for generating signature. Firewall checks header information and filter out any malicious activity.

2) Keibach and Crowcroft proposed a technique called "Honeycomb" [20]. It combines honeypot with automated signature generation scheme. Honeycomb generates signature consisting of a single longest common substring to match all worm's instances. The generation of single contiguous substring to match all instances of polymorphic worms is the big problem because it generates multiple alarms for the same attack.

3) Hyang-Ah Kim and Karp Proposed "Autograph" [20]**.** It is a distributed, automated worm signature generation scheme to detect polymorphic worms. Autograph takes input from across DMZ traffic. Payloads partition is done into different content block and using COPP algorithm. The content blocks are analysed and autograph selects most frequently occurring byte sequence across the flows in suspicious flow pool. Prevalence histogram is generated for each content block which acts as worm signature. Polymorphic worms may change their payloads in each injection. Autograph fails to address this problem.

4) Mohssen proposed a new technique called "Double Honeynet" [21]. It includes two honeypots, one for inbound traffic and other for outbound traffic. Both are high interactive honeypot, hence can collect sufficient amount of worm instances. For signature generation different methods are used like protocol classifier, clustering based on destination port, substring extraction algorithm, an efficient algorithm that converts worm substrings into binary representations and using these binary representations for pattern matching.

C. Honeypot Based Advanced SSH Model for Linux & UNIX like Operating System [22]

SSH is an encrypted remote connection mechanism, commonly used in Linux and UNIX based operating system. It provides a secure data communication over an insecure network.

The motivation for developing such model is attackers are searching the network for servers that can be used for their malicious activity. One of the most prominent target is servers on which the administrator has set up a remote access service i.e. SSH. When an attacker finds such a server that runs the particular service, he will try to compromise it, and if login attempt is successful then the attacker gains remote access to the server.

SSH Honeypot Working: SSH honeypot operation includes web trap of attackers who target SSH services, focus on SSH brute-force and dictionary attacks and analyse data.

First we deploy a SSH honeypot using a virtual private server (VPS). It was connected to internet using a static IP address and software for web trap. It is a medium interaction honeypot allows interaction with attacker and binds to SSH default TCP port22 and log each connection attempt with server.

To monitor attacker activity, the following tools are used, an open SSH server to collect attempted passwords, syslogging to remotely log important system events, logins and password changes, Sebek tool used to collect secretly all keystrokes on incoming SSH connections. Figures define the block diagram and working principal of that model.

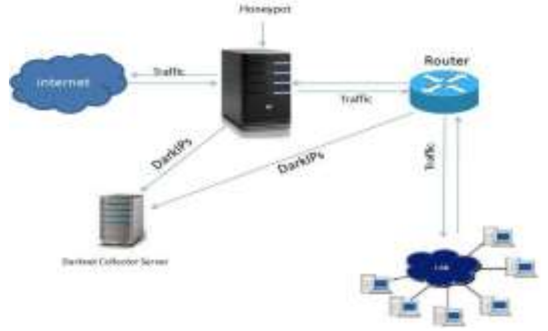

Fig. 7 Block diagram [22]

A Darknet is a private network and the connections are made only between trusted friends. SSH honeypot had one original root account and five non-privileged user accounts with password for each user and encourage attackers to enter the non-privileged user accounts instead of the root account. Most of results are surprisingly very low percentage of successful attacks on this system even with common passwords.

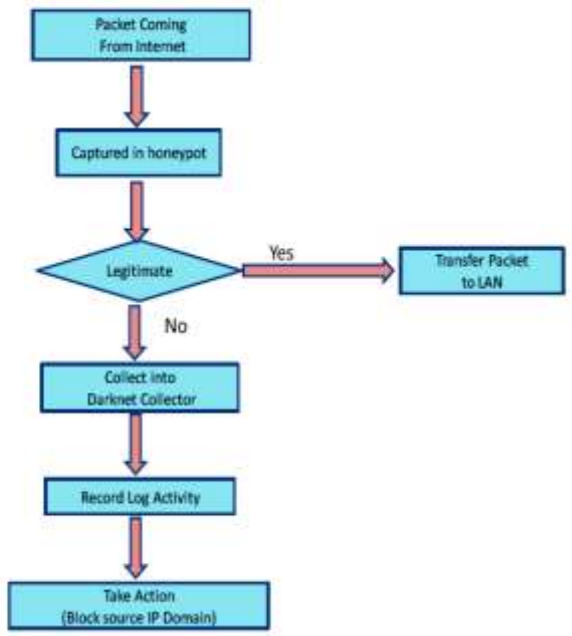
Packet Coming From Internet
Captured in honeypot
Legitimate
Yes
Transfer Packet to LAN
No
Collect into Darknet Collector
Record Log Activity
Take Action (Block source IP Domain)

Fig. 8 Working principal with flow chart [22]

D. *Honeypot against Advance Botnet Attack*

Botnet is one of the major internet threats now a day, mainly focus on compromising and controlling victim computers. Each compromised computer is installed with a malicious program called a “bot”, which is used to communicate with other bots in the botnet. Bot master or client bots as in figure maintain complete control of their botnets, and can conduct distributed denial-of-service (DDoS) attacks, email spamming, key logging, abusing online advertisements, spreading new malware, etc [23].

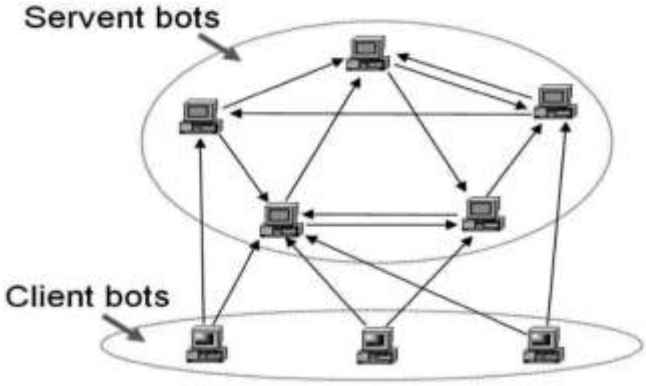

Fig. 9 Architecture of botnet [24]

Many network security defence system use honeypot to expose botnet membership and its behaviour. Honeypot can also be used to mitigate the malicious effects of botnets. Security professionals using honeypots have liability constraints such that their honeypots cannot be configured in a way that would allow them to send out real malicious attacks or too many malicious attacks [24].

The value of honeypot is directly related to the amount of time the attacker spends on it for malicious activities, so deploy honeypot in the network in such a way that it lures the attacker. When attacker gain control on honeypot system it behaves like a bot, but actually gather information about bot masters strategy. As security defenders build more honeypot based detection and defence systems, subsequently botnet operators have found counter strategies to avoid honeypot traps in their botnets so called as honeypot aware botnets [23]. Attackers can observe the data control and data containment activities of honeypot when it work as servant bot in the botnets and reliably determine that it is not a compromised bot.

A successful strategy of using a honeypot or multiple honeypots i.e. honeynets relies on the fact that they are undetected from external attackers and carefully control the propagation of data that is compromised or malicious data, which was generated by an attack [24].

VI. OTHER NETWORK SECURITY TOOLS AND THEIR COMPARATIVE STUDY

Honeypot actually cannot prevent cyber attacks against the network but helps in identifying and detecting them when used with other defense oriented tools such as Firewall and Intrusion detection system (IDS).

A. Firewall

Firewall defines a single chock point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network and provides protection from various kinds of IP spoofing and routing attacks. Single choke point simplifies security management because security capabilities are consolidated on a single system or set of system.

The firewall itself is immune to penetration. This implies that use of trusted system with secure operating system. Basically number of firewall can be deployed in the proper positions of the managed network for integrated, cooperative, and in depth network security protection [25]. It is notice that it does not protect against internal threads or against the transfer of virus infected programs or files.

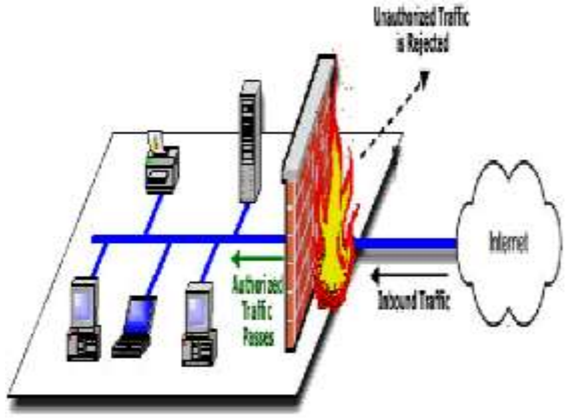

Fig. 10 Firewall deployment

B. Intrusion Detection System

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system [27]. Over the year, intrusion detection has been used by individuals and companies in a number of ways including erecting ways and fences around valuable resources with sentry boxes to watch the activities surrounding the premises of the resource.

IDS are easier to deploy as it does not affect existing systems or infrastructure but it is not a solution to all security concerns because it meet with the problem of false positive and false negative [6].

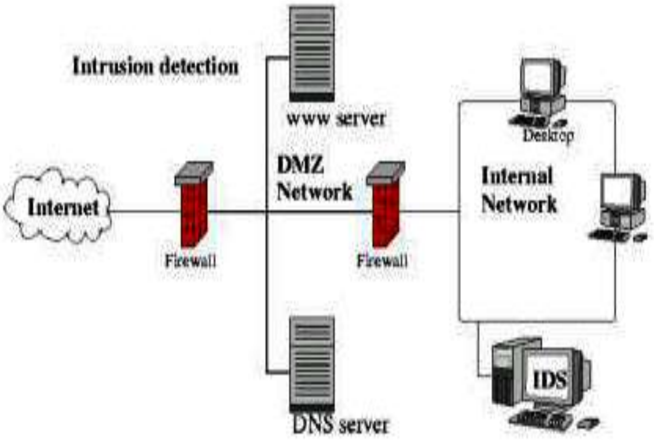

Fig. 11 IDS deployment [26]

TABLE IV

COMPARISON BETWEEN FIREWALL AND HONEYPOT

Firewall	Honeypot
It is design to keep intruders out of the network.	It is design to lure intruders to attack on the system.
Only authorized traffic will be allowed to pass.	It allows all traffic to interact with the honeypot system.
Placed at network's traffic entering points.	Placed inside the network as mimic the original production servers
Logs of incoming and outgoing traffic are maintained, so contains more entries.	Maintain the logs of interacted traffic only, so collect fewer entries.
It cannot protect from internal threats and from attacks that bypass the firewall.	It can protect from internal threats, information gathering is our prime aim.
According to purpose various firewalls are used i.e. packet filter, application level gate-ways and circuit level gateways.	According to purpose two types of honeypots are used i.e. production honeypot, research honeypot.

TABLE V

COMPARISON BETWEEN IDS AND HONEYPOT

IDS	Honeypot
A system silently monitors the network's traffic and gives alerts to tell about the kind of intruders based upon the database of existing intruders.	It is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information system.
IDS require signatures for detecting malicious activities.	Honeypot does not require any signature for detection.
IDS is fail to detect attacks if they are unknown at the time of its deployment.	Honeypots can detect vulnerabilities that are not yet understood or known.
Easy to deploy as it does not affect existing infrastructure.	Deployment complexity is based on type and purpose for which it developed.
It is suffer from the problem of false alerts like false positive and false negative.	It collects information about strategy used and generates alert when intruder try to compromise it, so overcome false alert problem.
According to monitoring scope in terms of area covered, it has two main types Network based IDS, Host based IDS	According to interaction with intruders it can be divided as low, medium and high interaction honeypots.

VII. CONCLUSION

Over recent years area of network security achieves the biggest progress because nobody wants that his system will be attacked by intruders. Honeypot technology is useful and extremely important part of an overall network security strategy if security professionals and researchers are to know their enemies, and insure that network security keeps pace with the rapid changes in network attacks. No other mechanism is comparable in the efficiency of a honeypot if gathering information is a primary goal. This paper describe new ways with honeypot to enhance network security policies but we also have to consider the fact that if attacker know about such system or bypass from it than the whole mechanism is meaningless, so develop a honeypot in such a way that attacker will definitely believe that it is a original production server. Strong control mechanism is required because if attacker is successful in controlling the honeypot system than it will not used by attacker for further attacking purpose.

REFERENCES

[1] Snehil Vidwarshi, Atul Tyagi, Rishi Kumar, "A Discussion about Honeypots and Different Models Based on Honeypot", 28th IRF International Conference, ISBN: 978-93-85465-37-6, June 2015.

[2] L. Spitzner, "Honeypot: Catching the Insider Threat", 19th Annual Computer Security Applications Conference, 2003.

[3] Niharika and Ranjeet Kaur, "Honeypot for Network Surveillance", International Journal of Research in Engineering & Technology, ISSN (E): 2321-8843, ISSN (P): 2347-4599 Vol. 2, Issue 5, May 2014.

[4] http://www.honeynet.org.

[5] Navneet Kambow and Lavleen Kaur Passi, "Honeypots: The Need of Network Security", International Journal of Computer Science and Information Technologies, ISSN: 0975-9646, Vol. 5, 2014.

[6] Snehal B Rase and Pranjali Deshmukh, "Summarization of Honeypot: A Evolutionary Technology for Securing Data over Network" International Journal of Science and Research, ISSN: 2319-7064, 2013.

[7] Xuxian Jiang, Dongyan Xu, Yi-Min Wang, "A VM Based Honeyfarm and Reverse Honeyfarm architecture for Network Attack Capture and Detection", 2006.

[8] Georgios Portokalidis, Asia Slowinska, and Herbert Bos, "Argos: an Emulator for Fingerprinting Zero-Day Attacks", ACM SIGOPS Operating Systems Review, Vol. 40, No. 4, pp. 15-27, October 2006.

[9] Yaser Alosefer and Omer Rana, "Honeyware - Web-based Low Interaction Client Honeypot", Proceedings of the International Conference on Software Testing, Verification, and Validation Workshops, pp. 410-417, April 2010.

[10] Yu Adachi and Yoshihiro Oyama, "Malware Analysis System using Process-Level Virtualization", Proceedings of IEEE Symposium on Computers and Communications, pp. 550-556, July 2009.

[11] Jianwei Zhuge, Thorsten Holz, Xinhui Han, and Wei Zou, "Collecting Autonomous Spreading Malware using High-Interaction Honeypots," Proceedings of the International Conference on Information and Communications Security, pp. 438-451, December 2007.

[12] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis "Detecting Targeted Attacks Using Shadow Honeypots", Proceedings of the Conference on USENIX Security Symposium, pp. 9-23, August 2005.

[13] http://labrea.sourceforge.net/labrea-info.html.

[14] Vinu V. Das, "Honeypot Scheme for Distributed Denial-of-Service", Proceedings of the International Conference on Advanced Computer Control, pp. 497-501, January 2009.

[15] Niels Provos, "A virtual honeypot framework", in Proceedings of the 13th conference on USENIX Security Symposium, Vol. 13, SSYM'04, Berkeley, CA, USA, 2004.

[16] Jose Nazario, "PhoneyC: A Virtual Client Honeypot", Proceedings of USENIX Workshop on Large-Scale and Emergent Threats, pp. 1-8, April 2009.

[17] Neil C. Rowe, E. John Custy, Binh T. Duong, "Defending Cyberspace with Fake Honeypots", Journal of Computers, Vol. 2, No. 2, pp. 25-36, April 2007.

[18] http://www.honeynet.org.pt/index.php/HoneyMole.

[19] Swapnali Sunder Sadamate, "Review Paper on Honeypot Mechanism-the Autonomous Hybrid Solution for Enhancing", Internationlal Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 227712X, Vol. 4, Issue 1, January 2014.

[20] Sounak Paul and Bimal Kumar Mishra, "Honeypot Based Signature for Defence Against Polymorphic Worm Attack in Networks", IEEE International Advance Computing Conference (IACC), 2013.

[21] Bimal Kumar Mishra and Dinesh Kumar Saini, "SEIRS epidemics model with delay for transmission of malicious objects in computer network", Applied Mathematics and Computation, Elsevier, 188, 2007.

[22] Shaik Bhanu, Girish Khilari, Varun Kumar, "Analysis of SSH Attacks of Darknet Using Honeypots", International Journal of Engineering Development and Research, ISSN: 2321-9939, Vol. 3, Issue 1, 2014.

[23] Bacher, P., Holz, T., Kotter, M. and Wicherski, G., "Know your enemy: Tracking botnets", 2008, available at http://www.honeynet.org/papers/bots/.

[24] Rajab Challoo, Raghavendra Kotapalli, "Detection of Botnets Using Honeypots and P2P Botnets", International Journal of Computer Science and Security (IJCSS), Vol. 5, Issue 5, 2011.

[25] S. Ioannidis et al., "Implementing a Distributed Firewall", in proceedings of the ACM Computer and Communication Security (CCS), PP. 190-199, 2000.

[26] Tejvir Kaur, Vimmi Malhotra, Dr. Dheerendra Singh, "Comparision of Network Security Tools Firewall, Intrusion Detection System & Honeypot", International Journal of Enhanced Research in Science, Technology & Engineering, ISSN: 2319-7463, Vol. 3, Issue 2, February 2014.

[27] Ram Kumar Singh & Prof. T. Ramanujam, "Intrusion Detection System Using advanced Honeypots", International Journal of Computer Science and Information Security, Vol 2, No. 1, 2009.