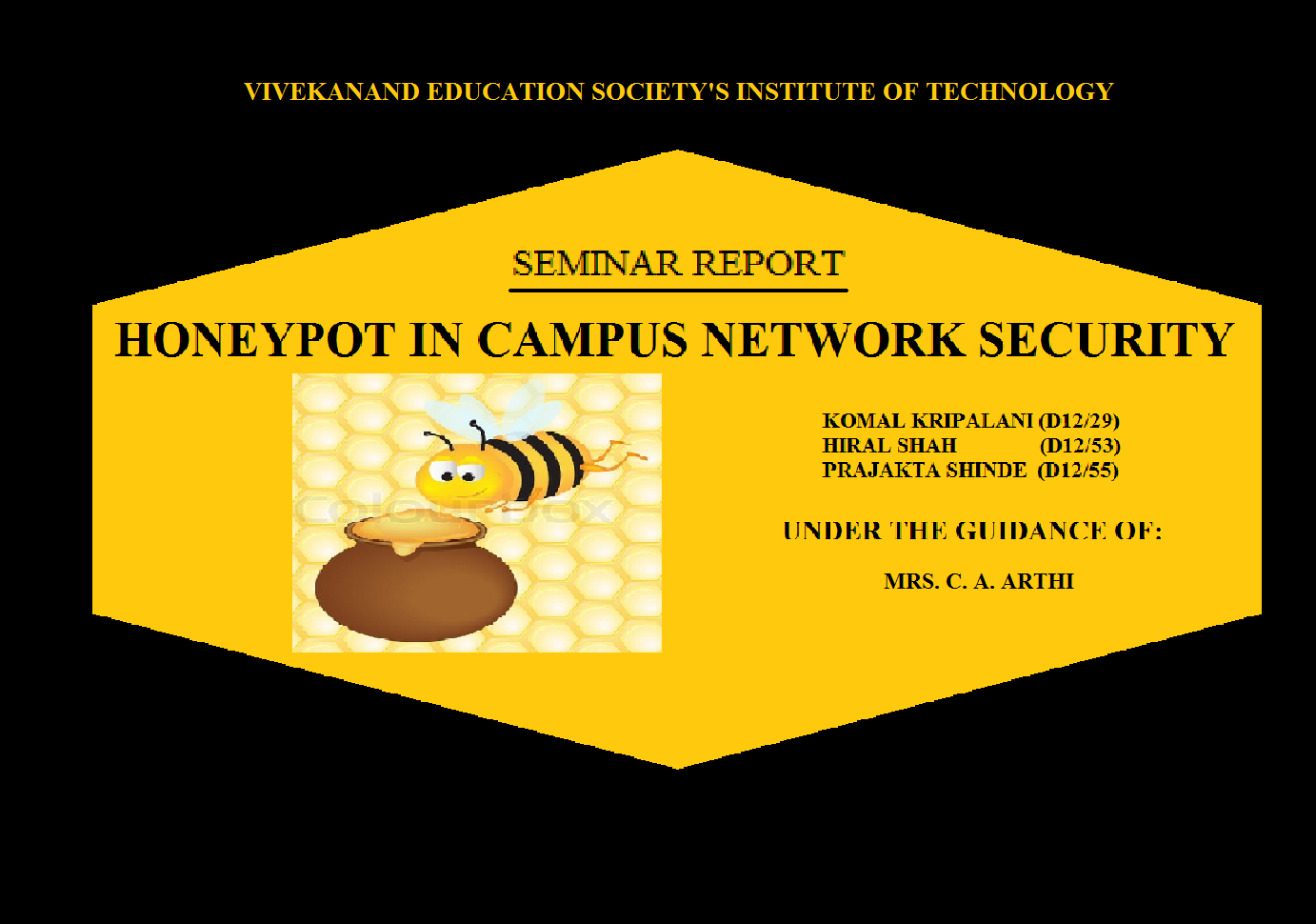
****

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TOPIC** | **PG NO** |
| **1.** | **Chapter 1** | **2** |
| a. | Introduction | 2 |
| b. | Problem Statement | 4 |
| c. | Problem Detail | 4 |
|  |  |  |
| **2.** | **Chapter 2** | **7** |
| a | Review of Literature | 7 |
| b. | Current Status | 14 |
| c. | Line of Research | 14 |
|  |  |  |
| **3.** | **Chapter 3** | **15** |
| a. | Methodology employed for the problem in hand | 15 |
| b. | Software tools required | 17 |
| c. | Hardware required | 17 |
|  |  |  |
| **4.** | **Chapter 4** | **18** |
| a. | Design(Architecture of the system) | 18 |
| b. | Implementation module | 24 |
|  |  |  |
| **5.** | **Chapter 5** | **28** |
| a. | Result analysis | 28 |
| b. | Limitation | 30 |
|  |  |  |
| **6.** | **Chapter 6** | **31** |
| a. | Conclusion | 31 |
| b. | Future scope | 31 |
|  |  |  |
| **7.** | **IEEE Paper submitted for an International Conference (Research on Applications of Honeypot in Campus Network Security)** | - |

**CHAPTER 1**

**INTRODUCTION**

Honey Pot Systems are decoy servers or systems setup to gather information regarding an attacker or intruder into your system. It is important to remember that Honey Pots do not replace other traditional Internet security systems; they are an additional level or system.

Honey Pots 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.

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

1. 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.
2. 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.

A campus network, campus area network, corporate area network or CAN is a [computer network](http://en.wikipedia.org/wiki/Computer_network) made up of an interconnection of [local area networks](http://en.wikipedia.org/wiki/Local_area_network) (LANs) within a limited geographical area. The networking equipments ([switches](http://en.wikipedia.org/wiki/Network_switch), [routers](http://en.wikipedia.org/wiki/Router_(computing))) and transmission media ([optical fiber](http://en.wikipedia.org/wiki/Optical_fiber), copper plant, [Cat5 cabling](http://en.wikipedia.org/wiki/Category_5_cable) etc.) are almost entirely owned by the [campus](http://en.wikipedia.org/wiki/Campus) tenant / owner: an enterprise, university, government etc.

A “campus” network uses a mix of technologies, products, and applications, and serves a large user population. The campus network presents a challenging security picture because of the diversity of elements to protect:

• **Servers**, including departmental servers for user access and file sharing, central application servers such as finance and databases, and Web servers for either public Web or Intranet applications.

• **Operating systems**, typically multiple versions of multiple OS’s running on servers and clients.

• **Network devices**, including routers, Layer 4-7 load-balancing switches, Layer 3 core switches, Layer 2 distribution switches, and wireless LAN access points.

• **Security devices**, such as firewalls, VPN gateways, intrusion-detection and anti-virus servers, SSL accelerators, authentication servers, and content filtering servers.

With the rapid development of Campus Network in colleges and universities which have compulsorily been confronted with more and more invasions and attacking,

network and information security issues become especially prominent and acute. The current available security measures including firewall, intrusion detection, anti-virus software, authentication technology and data encryption are mainly passive protecting based on the known facts and attack mode, which are not capable of defending complex and volatile attacks effectively.

Honeypot is a new network security technology based on the inveiglement theory developed in recent years. A honeypot is a network inveiglement system under strict surveillance, which attracts attacks by genuine or virtual network and services so as to analyze the blackhat's activities during honeypot being attacked by hackers, delay and distract attacks in the meantime.

Using honeypot technology, the network administrators of Campus Network could expand the network topology space, delude the attackers, delay attacking and distract targets, deplete the attackers' resource, protect productive network. Meanwhile network and information security community can track, record and analyze the hacker's actions focused on the honeypots comprehensively to discover and get acquainted with the internal and external threats to Campus Network, the common attacking tools, methods and rules, so as to amend the network security architecture, to revised security management principles of all levels, to adjust the firewall configuration to enhance the holistic security of Campus Network.

**PROBLEM DEFINITION**

* To implement honeypot in a client server program
* To create logs of various login and attacks
* To send a mail to the corresponding user that their account is being attacked

**PROBLEM DETAIL**

The main aim of the honeypot is to lure the hackers or attacker so as to capture their activities. This information proves to be very useful since information can be used to study the vulnerabilities of the system or to study latest techniques used by attackers etc. For this the honeypot will contain enough information (not necessarily real) so that the attackers get tempted.  
Their value lies in the bad guys interacting with them. Conceptually almost all honeypots work they same. They are a resource that has no authorized activity, they do not have any production value.

Theoretically, a honeypot should see no traffic because it has no legitimate activity. This means any interaction with a honeypot is most likely unauthorized or malicious activity. Any connection attempts to a honeypot are most likely a probe, attack, or compromise. It is this very simplicity that give honeypots their tremendous advantages and disadvantages.  
We have two general categories, honeypots can be used for production purposes or research. When used for production purposes, honeypots are protecting an organization. This would include preventing, detecting, or helping organizations respond to an attack. When used for research purposes, honeypots are being used to collect information. This information has different value to different organizations. Some may want to be studying trends in attacker activity, while others are interested in early warning and prediction, or law enforcement. In general, low-interaction honeypots are often used for production purposes, while high-interaction honeypots are used for research purposes. However, either type of honeypot can be used for either purpose. When used for production purposes, honeypots can protect organizations in one of three ways; prevention, detection, and response. We will take a more in-depth look at how a honeypot can work in all three. 

1. **Prevention** : Honeypots can help prevent attacks in several ways. The first is against automated attacks, such as worms or auto-rooters. These attacks are based on tools that randomly scan entire networks looking for vulnerable systems. If vulnerable systems are found, these automated tools will then attack and take over the system (with worms self-replicating, copying themselves to the victim). One way that honeypots can help defend against such attacks is slowing their scanning down, potentially even stopping them. Called sticky honeypots, these solutions monitor unused IP space. When probed by such scanning activity, these honeypots interact with and slow the attacker down. They do this using a variety of TCP tricks, such as a Windows size of zero, putting the attacker into a holding pattern. This is excellent for slowing down or preventing the spread of a worm that has penetrated the internal organization. Honeypots can also be used to protect the organization from human attackers. The concept is deception or deterrence. The idea is to confuse an attacker, to make him waste his time and resources interacting with honeypots. Meanwhile, the organization being attacked would detect the attacker's activity and have the time to respond and stop the attacker.
2. **Detection** : The second way honeypots can help protect an organization is through detection. Detection is critical, its purpose is to identify a failure or breakdown in prevention. Regardless of how secure an organization is, there will always be failures, if for no other reasons then humans are involved in the process. By detecting an attacker, you can quickly react to them, stopping or mitigating the damage they do. Traditionally, detection has proven extremely difficult to do. Technologies such as IDS sensors and systems logs have proved ineffective for several reasons. They generate far too much data, large percentage of false positives (i.e. alerts that were generated when the sensor recognized the configured signature of an "attack", but in reality was just valid traffic), inability to detect new attacks, and the inability to work in encrypted or IPv6 environments. Honeypots excel at detection, addressing many of these problems of traditional detection. Since honeypots have no production activity, all connections to and from the honeypot are suspect by nature. By definition, anytime a connection is made to the honeypot, this is most likely an unauthorized probe, scan, or attack. Anytime the honeypot initiates a connection, this most likely means the system was successfully compromised. This helps reduce both false positives and false negatives greatly simplifying the detection process by capturing small data sets of high value, it also captures unknown attacks such as new exploits or polymorphic shellcode, and works in encrypted and IPv6 environments. In general, low-interaction honeypots make the best solutions for detection. They are easier to deploy and maintain then high-interaction honeypots and have reduced risk.
3. **Response :** The third and final way a honeypot can help protect an organization is in reponse. Once an organization has detected a failure, how do they respond? This can often be one of the greatest challenges an organization faces. There is often little information on who the attacker is, how they got in, or how much damage they have done. In these situations detailed information on the attacker's activity are critical. There are two problems compounding incidence response. First, often the very systems compromised cannot be taken offline to analyze. Production systems, such as an organization's mail server, are so critical that even though its been hacked, security professionals may not be able to take the system down and do a proper forensic analysis. Instead, they are limited to analyze the live system while still providing production services. This cripples the ability to analyze what happened, how much damage the attacker has done, and even if the attacker has broken into other systems. The other problem is even if the system is pulled offline, there is so much data pollution it can be very difficult to determine what the bad guy did. By data pollution, I mean there has been so much activity (user's logging in, mail accounts read, files written to databases, etc) it can be difficult to determine what is normal day-to-day activity, and what is the attacker. Honeypots can help address both problems. Honeypots make an excellent incident resonse tool, as they can quickly and easily be taken offline for a full forensic analysis, without impacting day-to-day business operations. Also, the only activity a honeypot captures is unauthorized or malicious activity. This makes hacked honeypots much easier to analyze then hacked production systems, as any data you retrieve from a honeypot is most likely related to the attacker. The value honeypots provide here is quickly giving organizations the in-depth information they need to rapidly and effectively respond to an incident. In general, high-interaction honeypots make the best solution for response. To respond to an intruder, you need in-depth knowledge on what they did, how they broke in, and the tools they used. For that type of data you most likely need the capabilities of a high-interaction honeypot.

**CHAPTER 2**

**REVIEW OF LITERATURE**

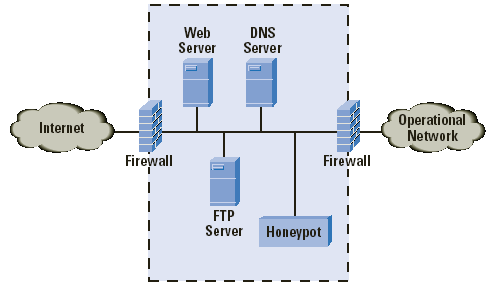
**Honeypot:**

A honey pot is a computer system on the Internet that is expressly set up to attract and "trap" people who attempt to penetrate other people's computer systems. (This includes the [hacker](http://searchsecurity.techtarget.com/definition/hacker), [cracker](http://searchsecurity.techtarget.com/definition/cracker), and [script kiddy](http://searchmidmarketsecurity.techtarget.com/definition/script-kiddy).) To set up a honey pot, it is recommended that you:

* Install the operating system without [patch](http://searchenterprisedesktop.techtarget.com/definition/patch)es installed and using typical defaults and options
* Make sure that there is no data on the system that cannot safely be destroyed
* Add the application that is designed to record the activities of the invader

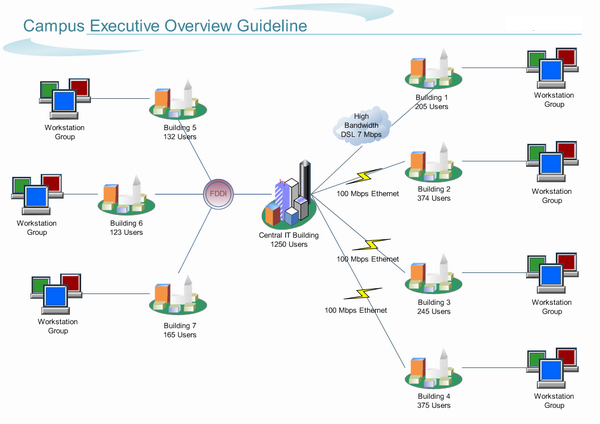
Maintaining a honey pot is said to require a considerable amount of attention and may offer as its highest value nothing more than a learning experience (that is, you may not catch any hackers).

We can also define honeypots as security resource who's value lies in being probed, attacked, or compromised. Unlike firewalls or IDS sensors, honeypots are something you want the bad guys to interact with.



**Campus network security:**

With the proliferation of computers around the world then coupled that with virus’s spreading from sharing floppy disks in the early 1980’s on network computers, hackers have been trying ever since to bring down computer systems. Whether it was meant as a joke, game or just nuisance, mischievous folks have been finding ways to hack computer systems and most times we are unaware of malicious code on our computer. The rush to be “webvergence” (which is the converging of voice, data and video on the Web) has introduced many different avenues for spyware and key-loggers to attack computer systems leaving them wide open to identify theft and internet predators.  Every day the local news station top stories are about the latest computer security threats from virtually unknowns due to the anonymous nature of the internet. Cyber-crime is on the increase targeting unsuspecting people via phishing scams, spyware, identify theft and internet predators.  We have seen attempts by hackers to compromise computer systems information as a direct result of focused attacks.



**Types of network attack**

There are four primary types of attacks, they are:

* Reconnaissance
* Access
* Denial of Service
* Worms, Viruses, and Trojan Horses

1. **Reconnaissance**

Reconnaissance attack is a kind of information gathering on network system and services. This enables the attacker to discover vulnerabilities or weaknesses on the network. It could be likened to a thief surveying through a car parking lot for vulnerable – unlocked - cars to break into and steal.

Reconnaissance attacks can consist of:

* Internet information lookup
* Ping sweeps
* Port scans
* Packet sniffers

Network intruders can use Internet tools, such as the nslookup and whoisutilities, to easily determine the IP address space assigned to a given organisation or network. After finding out the IP address, the intruder can then ping the publicly available IP addresses to identify the addresses that are active.

There are automate ping sweep tool which an attacker can use, such as[fping](http://en.wikipedia.org/wiki/Fping) or gping, these tools methodically pings all network addresses in a given range or subnet. This is like to going through a section of a telephone directory and calling each number to know who answers.

When the attacker discovers active IP addresses, the intruder or attacker uses a port scanner (Nmap or Superscan -softwares designed to search a network host for open ports) to determine which network services or ports are active on the active IP addresses. The port scanner queries the ports to determine the application or operating system (OS) type and version, running on the targeted host. Based on the information gathered, the intruder can determine if a possible vulnerability or weakness that can be exploited exists.

Packet sniffing or Network snooping are common terms for eavesdropping. The information gathered by eavesdropping can be used to pose other attacks to the network.

A common method for eavesdropping on communications on a network is to capture TCP/IP or other protocol packets and decode the contents using a protocol analyser or similar tools such as wireshark. After packets are captured, they can be examined for vulnerable information.

# Network Access Attacks.

Technology is forever evolving, so is hacking! It might come as a surprise to many that, as one wakes up in the morning and prepares for work, gets to the office and spends nine to twelve hour working; the same way a professional hacker spends all day modifying hacking techniques and looking for networks to exploit!Firstly, for an attacker to gain access to a system network, the intruder has to find out the vulnerabilities or weaknesses in the network authentication, FTP and web services. Finding and exploiting these vulnerabilities will enable the attacker to gain access to web account and other confidential or sensitive information.

**Types of access attacks**

**1. Password attack** : A Network attacker uses packet sniffer tools to obtain user accounts and passwords information. Normally we log in and out of a system using authentication passwords to shared resources in a router or server, an attacker also repeatedly attempts to log in to a shared resource or to gain unauthorised access to an organisation’s network; this can also be referred to as dictionary or brute force attacks. To carry out this type of attacks, the intruder can use tools like the [**L0phtCrack**](http://en.wikipedia.org/wiki/L0phtCrack)**or Cain.**These software or programs repeatedly attempt to log in as a user using words derived from a dictionary. Most dictionary attacks often succeed because network users often choose simple and short passwords, single words that are easy to predict.

**2. Trust Exploitation**: The goal of a trust exploitation attacker is to compromise a trusted host, using it to stage attacks on other hosts in a network. If a host in a network of a company is protected by a firewall (inside host), but is accessible to a trusted host outside the firewall (outside host), the inside host can be attacked through the trusted outside host.

**3. Port Redirection**: A port redirection attack is another type of attack based on trust exploitation. The attacker uses a compromised host to gain access through a firewall that would otherwise be blocked.Look at it this way; the host on the outside can get to the host on the public services segment, but not the host on the inside. If an intruder is able to compromise the host on the public services segment, the attacker could install software to redirect traffic from the outside host directly to the inside host.

**4. Man-in-the middle attack**: A man-in-the-middle (MITM) attack is implemented by intruders that manage to position themselves between two legitimate hosts. The attacker may allow the normal communication between hosts to occur, but manipulates the conversation between the two.

There are many ways that an attacker gets position between two hosts. A very good example is called the transparent proxy. The attacker prey on their victims by sending a phising email or by defacing a legitimate website.

When the victim loads the URL of a defaced webpage, the attackers URL is added to the front of it.

1. **Denial of Service (DoS) Attacks**

DoS attack prevents authorized users from using services by consuming system resources. Most times DoS attack is regarded as trivial but in a sense it is a consequentially threat. This type of attack poses potentially damage to networks. Not only are they easy to execute, but its among the most difficult to eliminate. DoS attacks deserve special attention from network security administrators.There are different types of DoS attacks. The following are some examples of common DoS threats:

**Ping of Death**

A ping of death attack gained prominence in the late 1990s.  Then were the older operating systems, which were not as secured as the recent ones. Ping of death type of attack took advantage of vulnerabilities or loop holes in older operating systems, what it does was to modified the IP portion of a ping packet header to indicate that there is more data in the packet than there actually was. A ping is normally 64 or 84 bytes, while a ping of death could be up to 65,536 bytes. Sending a ping of this size may crash an older target computer. Most networks are no longer susceptible to this type of attack.

**SYN Flood**

A SYN flood attack exploits the TCP three-way handshake. It involves sending multiple SYN requests (1,000+) to a targeted server. The server replies with the usual SYN-ACK response, but the malicious host never responds with the final ACK to complete the handshake. This ties up the server until it eventually runs out of resources and cannot respond to a valid host request.

Other types of DoS attacks include:

**i**.  **E-mail bombs** - Programs send bulk e-mails to individuals, lists, or domains, monopolizing e-mail services.

**ii**.  **Malicious applets** - These attacks are Java, JavaScript, or ActiveX programs that cause destruction or tie up computer resources.

### Solution:

### DoS and DDoS attacks can be controlled by the implementation of special anti-spoof and anti-DoS [Access Control Lists](http://www.orbit-computer-solutions.com/Access-Control-Lists--ACL-.php). ISPs can also implement traffic rate, limiting the amount of unnecessary traffic that crosses network segments.common example is to limit the amount of ICMP traffic that is allowed into a network, because this traffic is used only for problem-solving purposes.

# Malicious Code Attacks

Worm, virus, and Trojan horse attacks constitute a potential threat to end-user workstations.

**Worms**

A worm executes code and installs copies of itself in the memory of the infected computer, which can, in turn, infect other hosts on the network. The structure of a worm attack is as follows:

* **Creating loopholes**- A worm installs itself by exploiting known vulnerabilities in systems, such as naive end users who open unverified attachments in e-mails.
* **Parasitic ability**- After gaining access to a host, a worm copies itself to that host and then selects new targets.
* **Payload**-Once a host is infected with a worm, the attacker has access to the host, often as an authorised user. Attackers could use a local exploit to escalate their privilege level to administrator.

**Solution:**

**1.** Contain the spread of the worm in and within the network. Sort out parts of the network that are not infected.

**2.** Start patching all systems and, if possible, scanning for vulnerable systems.

**3.** Scan and locate each infected workstations inside the network. Disconnect, remove, or block infected machines from the network.

4   Clean and patch each infected system. Some worms may require complete core system reinstallations to clean the system.

**Viruses**

A virus is malicious software that is attached to another program file so that they can spread from one machine to another. For your machine to be infected, you must have or had run an infected program or software.

Viruses are potential threats to machines and the entire network, they don’t only constitute a strain or nuisance; but are like a time bomb that could destroy all files or contents in your hard drive.

A virus normally requires a delivery mechanism-a vector-such as a zip file or some other executable file attached to an e-mail, to carry the virus code from one system to another. The key element that distinguishes a computer worm from a computer virus is that human interaction is required to facilitate the spread of a virus.

**CURRENT STATUS**

“Seminar” was an opportunity to study fundamentals working of honeypots and the current status of the project is:

* We have learnt how to set up LAN
* Successful implementation of sending mail via java
* Redirecting the server to a decoy server

**FITTING OF THE PROBLEM STATEMENT IN THE LINE OF RESEARCH WORK:**

All the three goals defined in problem statement have been implemented successfully.

Honeypots can be used to gather intelligence on new attack methods. 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.

**CHAPTER 3**

**METHODOLOGY EMPLOYED:**

**Revisiting "What is 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. Honeypots can also analyse the ways in which attackers try to compromise an information system.

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.

**Password Attack:**

An attacker tries to crack the passwords stored in a network account database or a password-protected file. There are three major types of password attacks: a dictionary attack, a bruteforce attack, and a hybrid attack. A dictionary attack uses a word list file, which is a list of potential passwords. A bruteforce attack is when the attacker tries every possible combination of characters. Especially for attackers inside Campus Network is more potentially dangerous to network

security.

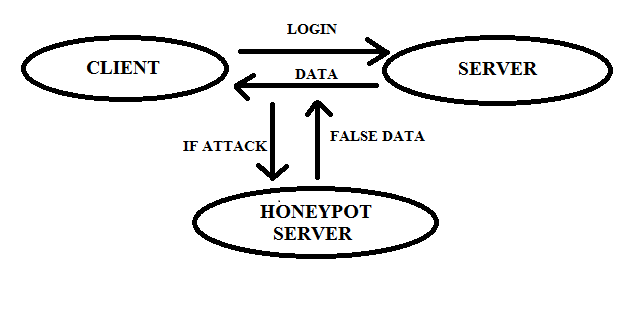
**Prerequisites:**

1. Three/two computers with windows 7 installed

2. A connection between them using LAN switch.

Assume there is a Server, a Client and a Honeypot Server.

1. Client sends request to access a folder present on server which is password protected.
2. When Client sends username and password to server, server saves this activity in logs and authenticates whether the user is allowed to access the file or not.
3. If authentication is successful, Client gets access to the requested file.
4. But if authentication fails continuously for 4 times, an email is sent to the user alerting him about the attack and request is redirected to honeypot server.
5. Gives report of no of attacks of particular user.
6. Gives report of no of attempts for a username and password.



**SOFTWARE TOOLS REQUIRED:**

1. Windows 7 OS
2. Java jdk

**HARDWARE TOOLS REQUIRED:**

1. Three computers with Windows 7 installed.
2. LAN connection cables(CAT 5)
3. LAN switch

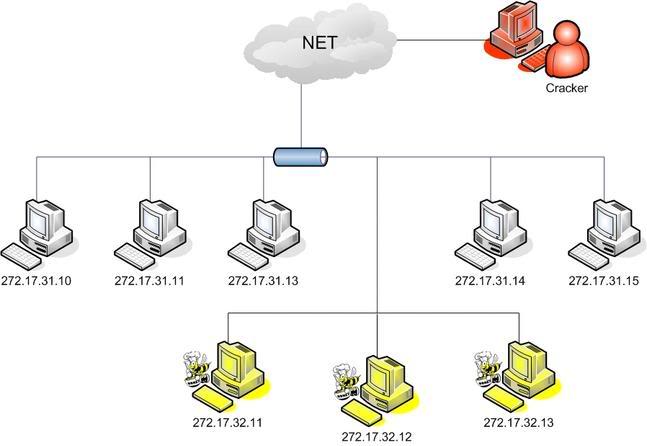
**CHAPTER 4**

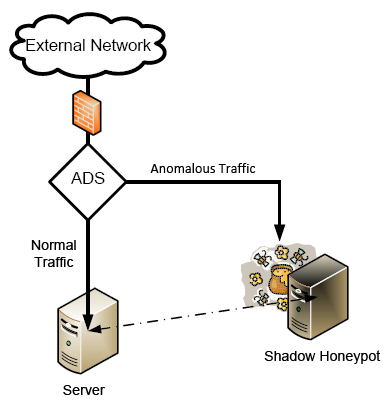
**DESIGN(ARCHITECTURE)**

**Working Of Honeypots:**

Conceptually, honeypots are very simple. They are a resource that has no production value, it has no authorized activity. Whenever there is any interaction with a honeypot, this is most likely malicious activity. For example, if you have someone on your internal network scanning for vulnerable desktops, and the attacker scans your internal honeypot, your honeypot will easily detect and log this unauthorized activity as no one should be interacting with it.

Honey pots work on the idea that all traffic to a honey pot should be deemed suspicious. Honeypots are generally based on a real server, real operating system, and with data that appears to be real. One of the main differences is the location of the machine in relation to the actual servers. The decoy machine are usually placed somewhere in the DMZ. This ensures that the internal network is not exposed to the hacker. Honey pots work by monitoring and\or sometimes controlling the intruder during their use of the honey pot. This can done whether the attack came from the outside or the inside of the network, depending on the location of the decoy system. Honey pots are generally designed to audit the activity of an intruder, save log files, and record such events as the processes started, compiles, file adds, deletes, changes, and even key strokes. By collecting such data the honey pots work to improve a corporation’s overall security system. If enough data is collected it may be used to prosecute in serious situations. In cases where you do not wish to prosecute the data collected can be used to measure the skill level of hackers, their intent, and in some cases, even their identity. All in all the honey pot helps a company prepare for attacks and respond to those attacks by learning from information gathering





**Value Of Honeypot:**

Honeypots are unique, they don't solve a specific problem. Instead, they are a highly flexible tool with many different applications to security. It all depends on what you want to achieve. Some honeypots can be used to help prevent attacks, others can be used to detect attacks, while other honeypots can be used for information gathering and research.

**Types**

In general, there are two different types, low-interaction and high-interaction. Level of interaction measures how much activity, or interaction, an attacker can have with a honeypot. Low interaction honeypots limit the level of interaction by emulating services. The interaction an attacker has with the honeypot is limited by how advance the emulation of the service is. An example of a low interaction honeypot is [Honeyd](http://www.honeyd.org/). In contrast, high interaction honeypots do not emulate services, instead they provide real applications for attackers to interact with. An example of a high interaction honeypot is [Honeynets](http://www.honeynet.org/papers/honeynet/). Neither is better then the other. Low interaction is simpler to deploy and has less risk (as the attacker can do less), but you can not learn as much. With a high level of interaction you can learn a great deal, as the attacker has a real operating system and applications to interact with. However, this comes at a cost, as the more interaction you provide, the more complex and greater risk you have.

Honeypots can be classified based on their deployment and based on their level of involvement.

Based on deployment, honeypots may be classified as:

1. **Production honeypots** are easy to use, capture only limited information, and are used primarily by companies or corporations; Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, which are easier to deploy. They give less information about the attacks or attackers than research honeypots do.
2. **Research honeypots** are run to gather information about the motives and tactics of the [Blackhat](http://en.wikipedia.org/wiki/Hacker_(computer_security)) community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats organizations face and to learn how to better protect against those threats.[[1]](http://en.wikipedia.org/wiki/Honeypot_(computing)#cite_note-1) Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations.

Based on design criteria, honeypots can be classified as

1. **pure honeypots**
2. **high-interaction honeypots**
3. **low-interaction honeypots**
4. **Pure honeypots** are full-fledged production systems. The activities of the attacker are monitored using a casual tap that has been installed on the honeypot's link to the network. No other software needs to be installed. Even though a pure honeypot is useful, stealthiness of the defense mechanisms can be ensured by a more controlled mechanism.
5. **High-interaction honeypots** imitate the activities of the real systems that host a variety of services and, therefore, an attacker may be allowed a lot of services to waste his time. According to recent researches in high interaction honeypot technology, by employing [virtual machines](http://en.wikipedia.org/wiki/Virtual_machine), multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. In general, high interaction honeypots provide more security by being difficult to detect, but they are highly expensive to maintain. If virtual machines are not available, one honeypot must be maintained for each physical computer, which can be exorbitantly expensive. Example: [Honeynet](http://en.wikipedia.org/wiki/Honeynet_Project).
6. **Low-interaction honeypots** simulate only the services frequently requested by attackers. Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the security of the virtual systems. Example: [Honeyd](http://en.wikipedia.org/wiki/Honeyd). Xypor

**E-mail trap**

An e-mail address that is not used for any other purpose than to receive spam can also be considered a spam honeypot. Compared with the term [spamtrap](http://en.wikipedia.org/wiki/Spamtrap), the term "honeypot" might better be reserved for systems and techniques used to detect or counter attacks and probes. Spam arrives at its destination "legitimately"—exactly as non-spam e-mail would arrive.

An amalgam of these techniques is [Project Honey Pot](http://en.wikipedia.org/wiki/Project_Honey_Pot). The distributed, open-source Project uses honeypot pages installed on websites around the world. These honeypot pages hand out uniquely tagged spamtrap e-mail addresses. and [Spammers](http://en.wikipedia.org/wiki/Spammers) can then be tracked as they gather and subsequently send to these spamtrap e-mail addresses.

**Database honeypot**

Databases often get attacked by intruders using [SQL Injection](http://en.wikipedia.org/wiki/SQL_Injection). Because such activities are not recognized by basic firewalls, companies often use database firewalls. Some of the available [SQL database](http://en.wikipedia.org/wiki/SQL_database) firewalls provide/support honeypot architectures to let the intruder run against a trap database while the web application still runs as usual

**Advantages of honeypot**

Honeypots have several powerful advantages. They include:

* Small data sets: Honeypots collect small amount of data, but almost all of this data is real attacks or unauthorized activity. Instead of dealing with 5,000 alerts and 10GB of logs every day, you may only get 30 alerts with your honeypots and 1MB of logs every day. Since honeypots collect only malicious activity, it makes it much easier to analyze and react to the information they collect.
* Reduced false positives: With most detection technologies (such as IDS sensors) a large percentage of your alerts are false warnings, making it very difficult to figure out what is a real attack. With honepyots, almost everything you detect or capture is an attack or unauthorized activity, vastly reducing false positives.
* False negatives: Unlike most technologies, its very easy for honeypots to detect and records attacks or behavior never seen before in the wild.
* Cost effective: Honeypots only interact with malicious activity, you do not need high performance resources. Most honeypots can easily run on an old Pentium computer with 128 MB of Ram.
* Simplicity: Honeypots are very simple, there are no advance algorithims to develop, nor any rulebases to maintain.

**APPLICATIONS OF HONEYPOTS IN CAMPUS NETWORK SECURITY**

In recent years, a serious of network security research and experiments have been implemented by different teams based on honeypot technology in diverse Campus Network circumstance. The research results show that honeypot technology can improve the security of Campus Network effectively. The four typical application scenarios are analyzed as follows.

**1. Analyzing Rogue Softwares**

**2. Expanding the network address space**

**3. Attracting attacks and gathering information**

**4. Catching the worms**

**IMPLEMENTATION MODULE**

**Program to send mail via java**

final String username1 = "honeypot.server2013@gmail.com";

final String password1 = "honeypot123";

Properties props = new Properties();

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.starttls.enable", "true");

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.port", "587");

Session session = Session.getInstance(props, new javax.mail.Authenticator() {

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(username1, password1);}});

try {

Message message = new MimeMessage(session);

message.setFrom(new InternetAddress("honeypot.server2013@gmail.com"));

message.setRecipients(Message.RecipientType.TO,InternetAddress.parse(emailid));

message.setSubject("SERVER ATTACK");

String msg="DEAR "+username+",\nYOUR ACCOUNT HAS BEEN HACKED BY

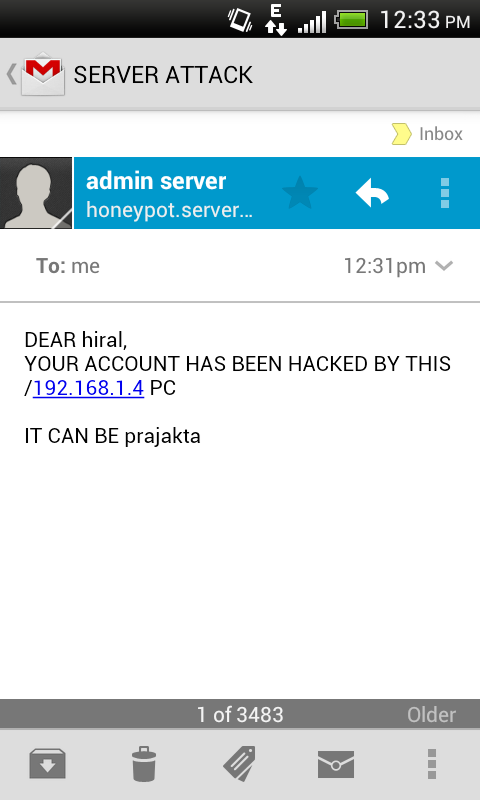
THIS "+ipadd+" PC";

if(!username2.equals(""))

msg=msg+"\n\n IT CAN BE "+username2;

message.setText(msg);

Transport.send(message);}catch(exception e){}

****

**Program to authenticate client and to determine password attack**

String login = "SELECT \* FROM login WHERE username='"+username+"'";

stmt.execute(login);

String datetime=sdf.format(d);

ResultSet rs = stmt.getResultSet();

while((rs!=null) && (rs.next()))

{

if(rs.getString("password").equals(password))

{

auth="AUTHENTICATED";

dos.writeUTF(auth);

Statement st = con.createStatement();

String log="INSERT INTO log VALUES

('"+username+"','"+password+"','"+datetime+"','Yes')";

st.execute(log);

Statement st1 = con.createStatement();

String lastip=" UPDATE lastip SET ipadd= '"+s1.getInetAddress()+"'

WHERE username='"+username+"'";

st1.execute(lastip);

count=0;

File marks = new File("s.txt");

BufferedInputStream bis = new BufferedInputStream(new

FileInputStream(marks));

byte[] mybytearray = new byte[(int) marks.length()];

bis.read(mybytearray, 0, mybytearray.length);

s1out.write(mybytearray, 0, mybytearray.length);

s1out.flush();

}

else

{

auth="PASSWORD WRONG";

dos.writeUTF(auth);

Statement st = con.createStatement();

st.executeUpdate("INSERT INTO log VALUES

('"+username+"','"+password+"','"+datetime+"','No')");

if(count>4)

count=0;

count++;

if(count==4)

{

String emailid=rs.getString("emailid");

InetAddress ipadd=s1.getInetAddress();

Statement st2 = con.createStatement();

st2.executeUpdate("INSERT INTO attack VALUES

('"+username+"','"+datetime+"','"+ipadd+"')");

Statement st3 = con.createStatement();

String user = "SELECT username FROM lastip WHERE

ipadd='"+ipadd+"'";

st3.execute(user);

ResultSet rs1 = stmt.getResultSet();

username2=rs1.getString(0);

if(username.equals(username2))

username="";

mail(username,emailid,ipadd,username2);

}

}

}

if(auth.equals("NOT AUTHENTICATED"))

{

dos.writeUTF(auth);

Statement st = con.createStatement();

st.executeUpdate("INSERT INTO log VALUES

('"+username+"','"+password+"','"+datetime+"','No')");

auth="NOT AUTHENTICATED";

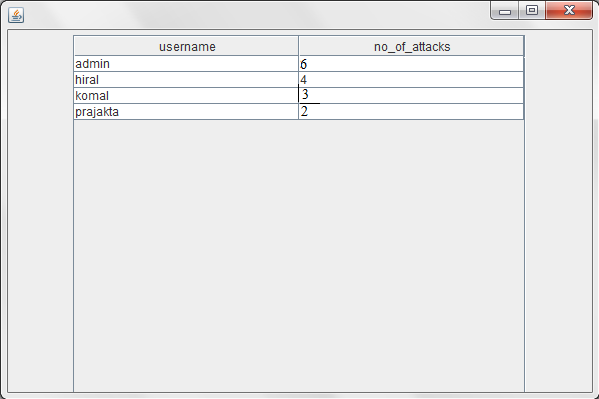
count=0;

}

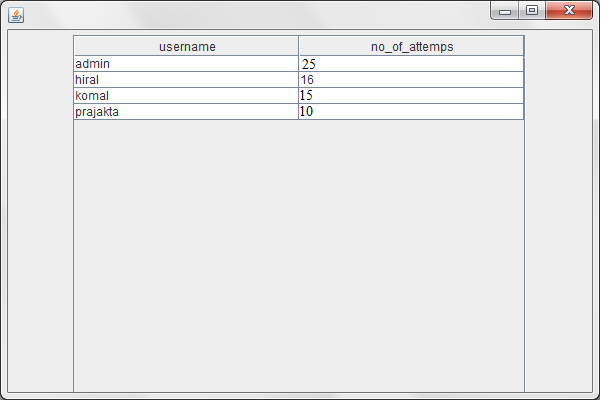
**CHAPTER 5**

**RESULT ANALYSIS**

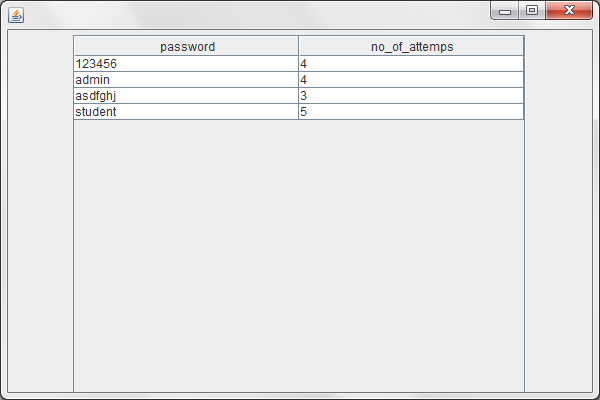
**Total no of attacks for each user**



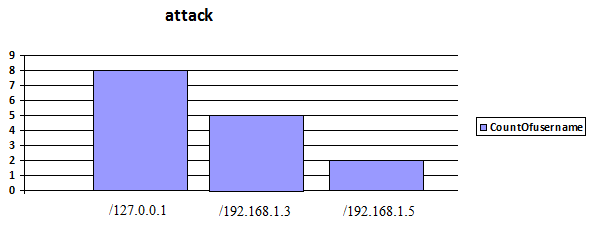
**Total no of false attempts for each user**



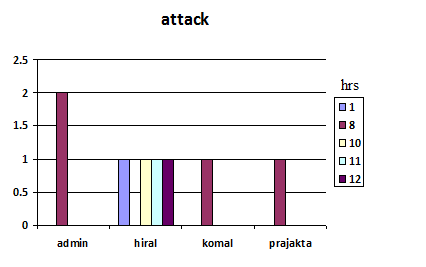
**Total no of false attempts for a particular password**



**No of attack for an ip address**



**Time for maximum attack**



**LIMITATIONS**

Honeypots also have their disadvantages. This is why they do not replace any existing technologies. Instead they work with and compliment your existing infrastructure.

* Honeypots only track and capture activity that directly interacts with them, they cannot detect attacks against other systems in the network.
* Honeypots are designed to be exploited, and there is always a risk of them being taken over by attackers, using them as a stepping-stone to gain entry to other systems within the network.

**CHAPTER 6**

**CONCLUSION**

Honeypots have their advantages and disadvantages. They are clearly a useful tool for luring and trapping attackers, capturing information and generating alerts when someone is interacting with them. The activities of attackers provides valuable information for analysing their attacking techniques and methods. As honeypots only capture and archive data and requests coming in to them, they do not add extra burden to existing network bandwidth.

**FUTURE SCOPE:**

* Government projects: Currently honeypots are mainly used by organizations, to detect intruders within the organization as well as against external threats and to protect the organization. In future, honeypots will play a major role in the government projects, especially by the military, to gain information about the enemy, and those trying to get the government secrets.
* Ease of use: In future honeypots will most probably appear in prepackaged solutions, which will be easier to administer and maintain. People will be able to install and develop honeypots at home and without difficulty.
* Closer integration: Currently honeypots are used along with other technologies such as firewall, tripwire, IDS etc. As technologies are developing, in future honeypots will be used in closer integration with them. For example honeypots are being developed for WI-FI or wireless computers. However the development is still under research.
* Specific purpose: Already certain features such as honeytokens are under development to target honeypots only for a specific purpose. Eg: catching only those attempting credit card fraud etc.