

Survey on Current HoneyNet Research

Goaletsa Rammidi
Member of UNB HoneyNet Project
Faculty of Computer Science
University of New Brunswick, Fredericton, Canada

Abstract

Network and Information security continue to be one of the areas that require more attention and improvement in the Information Technology world. Hackers are no longer just hacking for fun or to show-off their programming skills, but they are doing it for profit making reasons. At the same time businesses and many other activities in our everyday lives are becoming more and more dependent on Information Technology. HoneyNets provide another option not to replace but to compliment the existing security tools used, like firewall and IDS. A honeyNet provides an environment with no production value that hackers can interact with at different levels and the hacker activities monitored and captured while ensuring that the risk of the attacker using the honeyNet to hurt non honeyNet systems is mitigated. Also the attacker need not detect that they are being monitored. This survey looks into the current research that has been done on honeyNets so far and presents a summary of the most important aspects to know about honeyNets.

1. Introduction

HoneyNets provide an information gathering approach to security; they are used to gather information about threats in the network. A honeyNet is an interactive type of honeypot which provides real systems and application for attackers to attack and thus capture real information on a real attack unlike low-interaction honeypots which use emulated systems and applications. A honeyNet is made up of a network of honeypots. However note that some authors can also talk about low-interaction honeyNets, as in [4]

HoneyNets, hence all the other types of honeypots, are based on the idea of deception [7], in that the hackers are tricked into thinking that they are interacting with the real production systems, hence they can do all the damage thinking they are not being watched and at the end of the day, that collected information is used for the good of improving the detection, defense and mitigation techniques against those monitored attacks. ““We define computer security deception as being those actions taken to deliberately mislead computer attackers and to cause them to take specific actions. The application of honeyNets as part of a deception plan for network security is supported by this definition. Our general deception goal is to mislead an attacker into a predictable course of action that can be exploited (Dewar, 1989). “[7]”

According to [8], honeyNets are not only important for attracting attackers so as to capture

much information on the black hat community activities, but a honeynet project in the university, or any learning institute, can help students apply the knowledge they learnt in classroom about such things as the security policy, vulnerabilities and also develop a memorable hands on experience on different security aspects involved in setting up and maintaining a honeynet.

Honeypots can be used to add another security layer to the network as the firewalls and network intrusion detection systems (NIDS) systems used usually have some limitations .Some of the limitations are that the network firewall placed at the networks interface to the Internet cannot protect hosts within the corporation from threats that originate within the network, and some attacks can manage to bypass the firewall while the NIDS usually suffer from large numbers of false positives and false negatives [14]. The increasing use of encryption also reduces the amount of useful information that can be collected with a NIDS and also a NIDS may fail to detect new attacks with unknown signatures in its database [4]. Therefore deploying a honeynet to capture attacker actions can help get the information missed by the firewall and NIDS, also this information can be used to refine the NIDS rules to reduce the number of false positives. It can also help to deploy some stronger security measures to protect from internal threats [3, pg35] and capture decrypted data from encrypted sessions.

The collected information can also be useful to fighting criminals by the Government, it can be collected and analyzed to identify criminals, what they are doing and what are their ultimate goals [3, pg 34]. A honeynet can be set up to lure these criminals and convict them, and also if their actions can be predicted from the honeynet data then they can be stopped or the risks of their actions mitigated.

Disadvantages of honeynets; they need many different real systems and applications which may be costly, difficult to maintain and complex to deploy. Deploying virtual honeynets, discussed later in the paper, tries to reduce some of these problems. Also since honeynets usually use high interaction honeypots which are commercial-off-the-shelf products which are not hardened, a hacker who knows a particular operating system well and its vulnerabilities can compromise the honeypot and gain full access to the honeypot, or even entire honeynet. If data control measures employed are not strong enough the hacker can then use the honeynet to attack other non-honeynet systems. Analysing honeynet data takes more time and requires some skills.

2. Classification of Honeypots

2.1 Low-Interaction and High-Interaction.

Honeypots can be classified into two major groups as low-interaction honeypot and high-interaction honeypots [7], [10]. The difference between these classes is the extent to which the attacker is allowed to interact with the system, a low-interaction honeypot uses emulated systems and applications for the attacker and the system usually uses some scripts to

respond to the hacker's activities .e.g. the Honeyd honeypot which responds to Nmap and xprobe scan traffic with the help of the ARPD daemon and the Nepenthes which passively emulates vulnerabilities and then download the malware trying to exploit the displayed vulnerabilities[10].The major advantage of low-interaction honeypots is that they are simple and easier to deploy, and the information gathered is mostly statistical data on scan and worm attacks , and studying of new and ongoing attack patterns. [4] Emphasizes that “low-interaction honeypots can also be combined into a network, forming a low-interaction honeynet” and “attacker is not able to fully compromise the system, since he just interacts with a simulation, hence low risk.”

A high interaction honeypot on the other hand is the one that uses real systems and applications to interact with the hackers. A network of this high interaction honeypots is what we call a honeynet. These are just systems running real operating systems like Windows, Linux, routers etc that are placed on the network as honeypots and then exposed to the hackers to be attacked, and a lot of valuable information can be collected from the attackers' high interaction with the honeynet. For the accuracy and integrity of the data collected, the honeynet systems should not be intentionally compromised (or configured) in any way that will advertise them to or make them more vulnerable to attackers.[1].Some problems with high-interaction honeypots is that they need high maintenance and close monitoring, also analysing the attacker's actions and trying to find the motive of their attacks can take long times. These are also high risk compared to low-interaction honeypots, and they have poor scalability to include many machines [4]

2.2 Virtual Honeynet and Physical Honeynet

A physical honeynet is one in which the honeypots are running in separate physical machines other than being run a several virtual hosts in a single physical machine. The data control and data capture are also implemented in separate physical equipment than to be combined with the honeypots in one machine.

Virtual honeynet is a technology that virtually implements many different operating systems in one hardware computer, and hence instead of having a honeynet of different physically separate honeypots, all the honeypots will be virtually housed in one machine and still appear to the attacker like its different separate machines.

The main advantage of using a virtual honeynet over a classic honeynet is the cost of equipment; a virtual honeynet needs about one machine compared to buying many physical machines and their connecting equipment. It is also easy to manage since all the honeypots are configured in one machine. [10][3, pg 45]

Disadvantages are that the types of services that can be monitored by a virtual honeynet are limited by the hardware and virtualization software, for example”UML is only able to support Linux based operating systems [5], VMWare only supports operating systems that run on the xX6 processor architecture.”[7] and “it's difficult to run Cisco IOS on an Intel chip.” [10].

Also some virtual honeynet architectures are very less secure as it is possible for attackers to compromise the virtualization software and take over the entire honeynet. Virtual honeynets can be divided into self-contained and hybrid virtual honeynets [10], [7].

2.2.1 Self-Contained Virtual Honeynet

This is where the whole honeynet, which includes honeypots and the honeywall and any other data control and data capture tools, is contained in only one physical computer [10],[7]. It is advantageous because it is portable to carry the whole honeynet around from place to place e.g. if the computer used is a laptop, it is also possible to use the same honeynet to capture in different networks, at different times, easily e.g. one can just connect and operate the honeynet in LAN1, and later just disconnect and plug the honeynet into LAN2 and collect data easily. Also using only one hardware device cuts much on the cost of buying many separate computers.

The main drawback of this however is that since all the services are run in one machine if that machine fails or is compromised then the whole honeynet is gone, and if the attacker detects they are interacting with a self-contained virtual honeynet they can then attack other parts like the firewall, the IDS since they are all in the same system. Also there is going to be a great need of memory and CPU power in that machine, hence the cost of the machine and may be upgrading its CPU and memory may be too high .The virtualization software and hardware used can limit the type of the services offered.

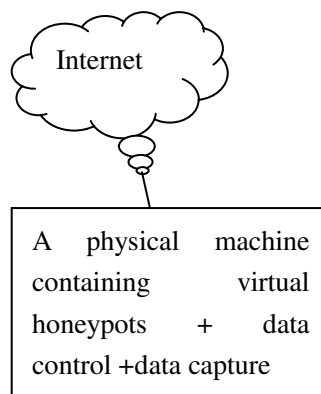


Figure1: Overview of a self-contained virtual honeynet

2.2.2 Hybrid Virtual Honeynet

In this approach the data control and data capture are implemented in physically separate machines and the honeypots are then run virtually on the same computer [10] [7]. This approach is more robust than the self contained (its not very easy for the hacker to compromise both the honeynets and data control and capture easily)and very flexible as there are no limitations on the type of technologies to use for data control and data capture.

The disadvantage is that it is not as portable as the self-contained one as there are more than one machines used and it may be more costly to buy several equipment.

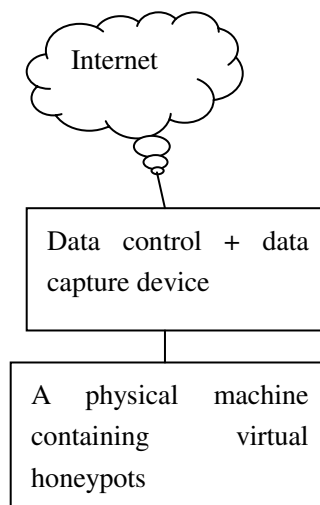


Figure2: Overview of a hybrid virtual honeynet

2.2.3 Some Related Study on Virtual Honeynets

A study from [7] compares the classic Gen II honeynet, the self contained honeynet and the hybrid virtual in terms of portability, easy of setting up at a new location, flexibility to deploy different systems, cost, security and how easy to detect it. It shows that classic honeynets and hybrid virtual honeynets are more secure than the self-contained because they separate the data control and data capture modules (honeypot) from the honeypots so compromising the honeypots does not make it easy to compromise these modules and attack other outside system. The security of a self contained virtual honeynet depends on how difficult it is to compromise the virtualization software and use the honeynet to attack other systems.

Also self contained are the most portable and easy to setup in a new location since the whole honeynet is in one machine and it will be easy to install images of the same operating system in different locations with less hardware problems. The Gen II are the most flexible as one can deploy any honeypot system they want without being restricted by the virtualization software or hardware, and the hybrid virtual are moderately flexible because even if one is restricted on the honeypots systems deployed, one still has the flexibility on the data control and data capture technologies. It also takes a lot effort for the attacker to detect that he is interacting with the honeynet in the classic Gen II architecture.

[7] Introduces the use of Security Enhanced Linux (SELinux) as a means of improving the security concerns of virtual honeynets which are the possibility of the attacker compromising the virtualization software and gaining full control of the honeynet, or the attacker compromising some of the honeypots inside and still taking control of the honeynet.

2.3 Classification of Honeynets: Gen i, Gen ii and Gen iii

GenI are the first honeynet architecture used by the honeynet project. They mainly use IPTables and Snort IDS for data control and data capture. GenII tries to improve the data control and data capture modules of GenI. In addition to the GenI IPTables firewall layer and Snort IDS layer, GenII adds another data capture layer where data is captured in the honeypot hosts using Sebek and Syslogd [3, pg 120] to capture more information and address the problem of encryption. In terms of data control, GenII improves GenI by adding another layer that uses Intrusion Prevention System, Snort-Inline operating in packet drop mode and packet replace mode to try and mitigate the risks that may be caused by those outbound packets allowed to pass by the IPTables.

GenIII offers a way of analysing data from different sources without the person having to manually go through different data sources and try to determine the relationships themselves. According to [10] “This release is considered a GenIII technology, as it has radical new improvements. It contains the core GenII Data Control and Data Capture functionality, but also now has remote GUI administration, Data Analysis integration, support for the Sebek 3.x branch, robust OS base, automated updating, and much more. We wanted a solution that any security professional could easily use and maintain.”

3. Main Functions of a Honeynet

3.1 Data Control

This is the most important function and must always be given high priority when implementing a honeynet. [3, pg38] this is providing an environment for the attacker to continue with the attack activities, with as much freedom as possible, and without noticing that he is interacting with a honeynet.[10],[3,pg38] This also has to prevent the attacker to use the honeynet to attack other non-honeynet machines either by mistake or intentionally, even when the honeynet itself is compromised.

It is important to implement different data control layers, such as counting outbound connections, intrusion prevention gateways, or bandwidth restrictions. It will avoid single point of failure especially for new attacks. This module also ensures that the honeynet blocks all outbound connections to the real production systems or to the rest of the Internet in case of software and hardware failures within the honeynet e.g. when the hard drive is full [3, pg 38]. Some examples of data control approaches include counting the number of outbound connections from within the honeynet, e.g. to a number say 5, then if any honeypot reaches that limit it is assumed to be compromised and an alert is logged and any more connections to that honeynet are blocked by the firewall. [1], intrusion prevention gateways, and bandwidth restrictions [10]. Data control should be configurable by the administrator at any time, include remote access for cases in which a problem arise when he is not physically in the honeynet area. There should always be automated alerting when a honeynet is compromised. [3,pg 39]

The GenII honeynets honeywall uses both the counting of outbound connections at the firewall, and then a Snort Inline is used as an active Intrusion Prevention System to drop and/or disable any malicious packets that have been allowed through the firewall. In case the attacker succeeds to compromise any honeypot in the honeynet, this honeywall further uses Swatch as a automated log monitoring tool and will therefore send an automated email alert to notify the administrators. IPTables to control the type of traffic allowed into and out of the honeynet, and also GRSecurity is a linux based tool that controls who can do what in the host, auditing privileges etc. in a honeypot etc .[13]

GenI data control uses only counting outbound connections, while Gen II counts outbound connections and then adds another data control layer which is an Intrusion Prevention System (IPS) [12].

3.2 Data Capture

This is the module that captures data on all attackers' activities in the honeynet. It tries to capture as much data and details as possible from the attackers actions by monitoring and logging all the hackers activities while ensuring that the hacker does not detect that his activities are monitored and recorded. All activities must be captured including activities on the network, against the honeynet hosts and also those activities that originate within the honeynet.

One of the major challenges faced by data capture is the use of encrypted data e.g. in ssh sessions, by the black-hat community. This means that encrypted data has to be captured in the unencrypted form, which can either be done by trying to break the decryption key and then decrypting the data. This has proven to be a very difficult approach [3] .Another approach is to capture the data post decryption and this is approach used by the kernel based Sebek tool used for data capture by the honeynet project.

Using different layers to capture this data can be beneficial since if one system missed some details, then it's possible that we can capture them with the other ones. Also if the attacker is able to detect that they are monitored and bring down the capturing tool, e.g. disabling or bypassing Snort somewhere, then we capture those details and others with the remaining tools.

Another problem in data capturing is that, if detected, attackers can delete or modify the captured data, e.g. modifying the snort logs to mislead on how the actual attack was conducted or change the details on actions that generated alerts e.g. signature id etc. hence it is advisable to store the data in a separate secure location. [10].Some examples of data capture tools used by the honeynet project are described in detail below.

3.2.1 Using Sebek capturing tool

Sebek is a data capturing tool that is used by the honeynet project GenII and GenIII

architectures. It resides in the kernel space and is able to capture some data on user activities as they access the system. It is able to copy most or the entire intruder's activities from the honeypot(s) and transfer it over the honeynet network to the honeywall anonymously without the intruder detecting this. The Sebek tool does key stroke logging for encrypted sessions, and it is able to use this to strokes to retrieve passwords that intruders use for remote logins, retrieve keys to decrypt the encrypted data and to recover files copied over session control protocol (SCP) etc[8], [2]

When encryption is not used it is easier to get the intruder's keystroke actions and user output details using a tool like Ethreal which can sniff packets in the wire and then do some stream reassembly to determine the TCP sessions of the intruder. Using Sebek at operating system kernel space enable administrators to capture any intruder's activities regardless of any binaries the intruders are using for decryption. Also because the user space is separate from the kernel space, this Sebek can be hidden from all users including the one with root privileges. [2] "Sebek version 2 records keystrokes and all sys_read data. An intruder, even using a sniffer, cannot detect Sebek traffic." [8]. There is some research on how this Sebek can be detected, disabled and bypassed by attackers on [8] along with how the Sebek maintainers tried to improve it as the vulnerabilities were pointed out by this study.

3.2.2 An intrusion detection system

Using snort open-source Intrusion Detection System (IDS) to monitor and log all traffic in the honeynet. [13] Snort will sit between the honeynet and the external network and examine and report all the traffic as per snort configurations, and also generate alerts. A GenI honeynet example deployed by the honeynet project used a layered data capture by using Snort IDS and TCPdump as their data capture tools [3, pg 53] and the two tools dumped data in separate memory partitions. Snort was set to a full logging mode so that it captures all the IP packets and also TCPdump captures all the network traffic in binary format. So whenever the snort failed like when "the partition that Snort was logging to became full and traffic capture was aborted" [3, pg 53], the data missed by Snort was fully recorded by TCPdump. Another type of IDS like Bro, which most is anomaly detection based can be used to complement Snort to capture more details and also for results correlation.

3.2.3 Firewall logging

The firewall monitors all traffic entering and leaving the honeynet, so logs of this traffic can provide very useful data for honeynet e.g. an attempted backdoor access to the honeynet. The firewall will even log the failed or refused connections which mean that some detail means by other tools here can be captured by these logs. GenI, GenII and GenIII use the IPtables firewall and connections are logged using the Syslogd daemon [3, pg123]. Outbound firewall logs are very important since a honeypot must have no production value, any traffic originating from inside the honeynet is suspicious and it calls for serious and detailed analysis.

3.3 Data Analysis

This tries to get some useful information out of the large captured data.

3.4 Data Collection

This is collecting all the data captured by different honeynets in a distributed environment to a central location, needs a secure storage.

4. Data Control Methods for Honeynets

4.1 Counting outbound connections

Personally I think using this approach alone, as in [1], is very risky because if an attacker has already detected that they are interacting with a honeynet, then they can infect other honeypots first before making outbound connections, and if maybe there are 5 honeypots then this attacker will have a chance to make 20 outbound connections before the firewall can block all of them. Even if all honeypot systems are different this can always work for common attacks like launching DoS attacks.

Also a sophisticated attacker can learn that they are interacting with a honeynet and then ensure that whatever the attack they want to launch from the honeynet is achieved within the allowed outbound limit, as in Gen I [12]. Therefore different data control layers are needed, emphasized by the Gen II architecture.

Limiting outbound connections helps stop the attacker from scanning or launching attacks to a large number of hosts from the honeynet, it can also help prevent the honeynet from being used to launch Denial of Service attacks (DoS). This number can also be a signature to the attacker where they will try to launch several connections from the honeynet, and if they are blocked after a certain number they will know they are likely to be interacting with a honeypot. [10]

4.1.1 Using IPTables

GEN II honeynet architecture is able to achieve this by using IPTables which are a built-in linux firewall. There is a limit that is set for the number of daily of IP protocols including TCP, UDP and ICMP outbound connections that each honeypot can make. Once the attacker reaches that limit, no more connections are allowed from that honeypot and the IPTables reset with a time limit policy, if maybe the policy is 24 TCP connections per 24 hrs then the IPTables will use that policy to allow only 24 connections in the next 24 hours. The attacker will not be allowed to start again until the time limit policy is reached. [10]. GenI and GenIII also use the IPTables firewall.

“We use IPTables for this, which is configured and implemented by the rc.firewall script

which comes with the Honeynet CDROM” [10]

EXAMPLE

Note, the variable OTHER is any IP protocol that is NOT TCP, UDP, or ICMP (such as IPsec, IPv6 tunneling, Network Voice Protocol, etc).

###Set the connection outbound limits for different protocols.

```
SCALE="day"
```

```
TCPRATE="15"
```

```
UDPRATE="20"
```

```
ICMPRATE="50"
```

```
OTHEREATE="15"
```

The above example shows the default outbound connection limits per protocol in the Honeywall CDROM.

4.2 Dropping or Disabling Malicious Packets

4.2.1 Using snort-inline

Gen II honeynet uses snort-inline with help of IPTables to drop or disable those packets that match known attack signatures. Dropping the malicious packet will totally stop the intended attack from happening while disabling it will try to reduce or mitigate the damage caused by that packet to the target outside machine. IPTables acts as a routing process that forwards packets to and from snort-inline to analyse. Outbound packets that IPTables allow are placed in a user space queue using ip_queueing module and snort-inline accesses this queue to inspect the packets, either dropping or disabling them depending on the rules and returns them to IPTables again for further routing. IP tables will always count every outbound connection made regardless of whether snort-inline drops or allows the packet through. [10]

Disadvantage is it will only drop or disable known attack signatures, and will not detect unknown attacks.

EXAMPLE , taken from [12]

```
alert tcp $HONEYNET any -> any 53  
msg:"DNS EXPLOIT named";flags: A+;  
content:"\CD80 E8D7 FFFFFFFF/bin/sh";  
replace:"\0000 E8D7 FFFFFFFF/ben/sh";)
```

Figure 3. Snort-Inline signature used to modify and disable a known DNS attack using the replace option. Highlighted in bold is the command used to modify and disable the attack.

4.3 Automated Log Monitoring

An automated log monitoring tool like Swatch can be used to monitor the system logs, Unix

logs in this case, and then send automated email alerts to administrators when a system log triggers the alarm. Different alerting approaches, e.g. displaying a honeypot that has problems with a different color and then sending an email can be used to bring the administrator's attention to the honeynet.

4.4 Bandwidth Restrictions

The techniques described in this section help to control the bandwidth that is allowed at a specified interface, either for inbound or outbound traffic. However, during this research I have not found any honeypot implementation that uses them, so their effectiveness and accuracy in honeynet data control is not verified in this paper, most honeypots research use the Honeynet Project techniques.

4.4.1 FreeBSD's Dummynet

Dummynet is a tool that was originally developed for testing some network protocols, and has a bandwidth restriction capability. The FreeBSD dummynet is one of the approaches to deploy these by using the ipfw rules in the FreeBSD ipfw firewall. Dummynets intercepts packets as they move up the protocol stack and are passed through queues and pipes to enforce bandwidth restrictions, add latency, packet loss, multiple paths effect etc depending on the ipfw rules." Pipes are fixed-bandwidth channels. Queues represent instead queues of packets, associated with a weight, which share the bandwidth of the pipe they are connected to proportionally to their weight." [15].

The following are examples of how FreeBSD dummynet can use ipfw rules to impose bandwidth restrictions per protocol taken from [15].

Examples

- These rules limit the total ICMP traffic (inbound and outbound) to 50Kbit/s
ipfw add pipe 1 icmp from any to any
ipfw pipe 1 config bw 50Kbit/s queue 10
- These rules limit inbound traffic to 300Kbit/s for each host on your network 10.1.2.0/24.
ipfw add pipe 2 ip from any to 10.1.2.0/24
ipfw pipe 2 config bw 300Kbit/s queue 20 mask dst-ip 0x000000ff

4.4.2 Cisco's Committed Access Rate

Cisco's Committed Access Rate: This is a feature in some Cisco IOS versions that allow placing some upper bound limits on the transmission rate of the network interface; hence it can be deployed on the network edge router to limit the transmission rate of traffic entering and leaving the network. The traffic can be limited based on the incoming interface, IP access list, MAC addresses. One can configure multiple policies on an interface such that different traffic will be treated differently depending on the policy. In a honeynet setup, a router can be placed at the edge of the honeynet to separate it from the production network and rest of Internet. This router will then be configured to limit on the transmission rate from the

honeynet to the outside network, one May use honeypots MAC addresses in the matching policy. These can be implemented on some Cisco routers like the “To configure CAR (or DCAR on Cisco 7000 series routers with the RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor”

A common application of these can be where ISPs try to limit customers connecting to a high speed interface to only transmit and receive at the rate that they have purchased. Hence CAR will allow sharing of bandwidth of the interface among several customers and enforce per host bandwidth limitations [6]. Other examples of bandwidth restrictions include Linux’s Advanced Routing and Traffic Control , and Juniper’s Traffic Policing.

5. Where to Place a Honeynet in the Company Network

There are three options of where one can place their honeynet in the network with respect to other networking devices and these are either externally facing the Internet, Internally behind the firewall or in the DMZ zone [5, pg 54]. Each of these locations has their own advantages and disadvantages and the choice depends on factors like amount of network resources the company has, the objectives of deploying a honeynet e.g. a research based institution that wants to capture as much hacker details as possible will likely put their honeynet externally , whereas an organization that wants to have up-to-date details of the possible exploits to their production systems will likely place their honeynet internally. The expertise of the administrators can also be a factor as some deployments are more complex than others. Also note that if a Honeynet Project honeywall is placed in front of the honeypot in any of these situations it can improve both the data control and data capture.

5.1 External placement

In this setting the honeynet is put right behind the ADSL modem or any other router connecting the company to the Internet. The honeypot shares the same IP address subnet with the production systems .This method can be suitable when one has only 1 public IP address and wants to setup just a single computer honeypot for a small research.

A major advantage for this type is that it can capture more details than others because there are not many filtering devices e.g. firewall in front of the honeynet. It is very easy to deploy, one does not need to worry about configuring which port traffic from the firewall should be allowed to pass to the honeypot or which unwanted traffic to redirect to the honeypot e.g. RPC traffic going to port 135 can all be redirected to the honeypot. [5].It can also be considered the least expensive because it doesn’t require many networking devices to deploy and have it running.

However a trade-off for this is the data control may not be enough and the production systems will be at risk of the attacker using the honeynet to compromise them.

5.2 Internal placement

This is where the honeynet is placed behind the company's firewall and other external defence mechanisms. It is best suited as security backup to give early warnings to exploits that have managed to pass through the company's defense systems like firewalls, while at the same time monitoring internal activities for exploits that originate internally.

A problem may be that it might be difficult to provide local data control or another router may have to be placed in front of the honeypot to protect it.

5.3 DMZ placement

This is the most common deployment [5] Here the honeynet is placed in the company DMZ zone in the same subnet alongside some legitimate public access DMZ servers like web server, email server etc. It will be able to capture early details of attacks to devices in the DMZ zone. It is possible to enforce very strong data control measures than in the other two options like placing an additional router between the honeynet and the DMZ firewall to control the outgoing connections from the honeypots, and since the production networks are still protected behind the DMZ zone they are at less risk to be attacked if one of the honeypots are compromised.

A disadvantage of this application is that the honeypot will not be able to capture attacks that make it past the DMZ zone into the internal production network and any other exploits that originates from inside the network. The administrator is also required to separate which traffic is allowed to the honeynet and not to other legitimate servers and also which traffic to automatically redirect to the honeypot. According to [5], this is the most complex to deploy.

6. Honeynet Taxonomy

[20] Suggests that a honeypot taxonomy can be built by grouping honeypots according to their security goals, which results in four groups being prevention, detection, reaction and research. Honeypots achieve prevention by tricking hackers into thinking that they are interacting with real operating systems, so hackers spend their time and effort attacking the honeypots and during this time real systems are protected.

A honeypot can also be used to detect hacker's activities and raise alerts based on the activities taking place in the system e.g. a certain number of outbound connections can imply that a machine somewhere is trying to use our production system for malicious purposes, or detecting many scan ports to different honeypots in the honeynet can imply an automated worm tool is trying to infect our production network. Using honeynets for research purposes tries to find tools; methods etc. used by hackers and come up with ways to improve the defence and control mechanisms, and these can just be deployed on a lab where there are no other production network machines.

A reaction honeypot on the other hand is used to try and keep the security on the production system up-to-date, a honeypot is configured to be just like the production system and monitored to see the details of any attacks on it, discover vulnerabilities and appropriate patches for them. This information is then used to harden the real production system and it reduces the organizational losses that can be incurred by taking the production system offline for analysis for long periods in case it is compromised.

[20] Further suggests that honeynet taxonomy can be built by their application goals, e.g. a honeypot can be aimed at catching DoS attacks, a worm honeypot, a spam mail filter honeypot [20].

7. Conclusion

Information from this research shows that a lot of research work has been done by the honeynet project in the development and improvement of tools to use when deploying honeynet. Honeypots are very important to compliment other security in the organization by capturing hacker information that is analysed. A low-interaction honeypot is a honeypot that uses emulated services and scripts to interact with the attacker while a high-interaction honeypot uses real operating systems. There are mainly two types of honeynets; a physical honeynet and a virtual honeynet. And these have undergone stages of improvement in their data control and data capture modules which have lead to the GenI, GenII and GenIII honeynet architectures. If other researchers can try to test and use other possible data control and data capture technologies not used by the Honeynet project it can be more beneficial to the honeynet development, and may help discover some limitations and improvements to the current technologies.

REFERENCES

- [1] Curran K. et al, Monitoring hacker activity with a Honeynet, Int. J. Network Mgmt 2005,January 2005, pg 123-134
- [2] Dornseif M. et al, NoSEBrEaK- Attacking Honeynets, 2004 IEEE, June 2004, pg 123-129
- [3] The Honeynet Project, Know Your Enemy Learning About Security Threats, Addison Wesley, Boston, USA, July 2004, 2nd ed.
- [4] [VHBook] Provos N. and Holz T., Virtual Honeypots from Botnet Detection to Intrusion Detection, Addison Wesley, Boston, USA,July 2007
- [5] R. A Grimes, Honeypots for Windows, Kinetic Publishing Services, 2005
- [6]http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfcarr_ps1835_TSD_Products_Configuration_Guide_Chapter.html
- [7] Warkentin, Merrill(Editor). Enterprise Information Systems Assurance and System Security : Managerial and Technical Issues. Hershey, PA, USA: Idea Group Publishing, 2006. p 266. <http://site.ebrary.com/lib/unblib/Doc?id=10116544&ppg=280>
- [8] Romney G. W et al, IT Security Education is Enhanced by Analyzing Honeynet Data, 2005 IEEE, pg F3D-10 – F3D-14
- [9] Yan L K , Virtual Honeynets Revisited,2005 IEEE, pg 232 -239.

[10] <http://www.honeynet.org>

[12] Spitzner L, The Honeynet Project: Trapping the Hackers, 2003 IEEE, pg 15 – 23.

[13] Chamales G , The Honeywall CD-ROM,2004 IEEE, pg 77 – 79.

[14] Russell, Ryan(CB). Hack Proofing Your E-Commerce Site: The Only Way to Stop a Hacker Is to Think Like One. Rockland, MA, USA: Syngress Publishing, 2000. p 98.
<http://site.ebrary.com/lib/unblib/Doc?id=10007042&ppg=125>

[15] http://info.iet.unipi.it/~luigi/ip_dummysnet/