ANALYSIS OF OPENSOURCE HONEYPOTS

[1]Kusum Yadav, [2.]Dr Rakesh Kumar
[1,] Kusum Yadav Ph.D Research Scholar, JJT University Jhunjhunu
,Dr Rakesh Kumar, Director, KSJIET, Modinagar, Uttar Pradesh, India,

Honeypot technology, its roots, historical background and various generations has already been discussed in detail while doing literature review in second chapter. in this chapter the work is concentrated into analyzing the open source honepots, in the following section. The use of honey pots in network security is emphasized and their place in the network security hierarchy has beendiscussed.

1 USE OF HONEYPOTS IN NETWORK SECURITY

Honeypots add extensive value to detection of unauthorized activity on the network. Three common challenges of detection are false positives. *False negatives and data aggregation*. False positives are when systems falsely alert suspicions or malicious activity, i.e., system might interpret valid network traffic as au attack. Many a time these kinds of alerts lead to an ignorance factor exhibited by administrator following a stream of false positives they might ignore the actual attack stream. False negatives are when an organization fails to detect attacks. Network intrusion detection systems not only face a challenge of false positives but also have problems with false negatives. Many NIDS systems, whether they are based on signature databases. Protocol verification or some other methodology, can potentially miss new or unknown attacks. Before databases get updated, a new attack tool can do a great harm.

Other major problem with reactive methodology of defense is Data aggregation; NIDS, system logs, application logs, firewall logs etc. capture tons of data. Ana1ysi& of this data to uncover the malicious intent is like searching in the wild.

Honeypots address all these three challenges very effectively. Most honeypots have no production traffic, so there is little activity to generate false positives. In most of the cases, except misconfiguration, honeypots generate valid alerts, greatly reducing False positives. One of the primary benefits is that honeypot can detect a new attack by virtue of the system activity not signatures. Honeypots generate only several megabytes of data a day, most of which is of high value. This makes it extremely easy to diagnose useful information from the Honeypots.

1.1 Using Honeypots in the DeMilitarized Zone(DMZ)

DMZ is a network of untrusted systems normally used to provide services to the Internet such as e-mail or web server. These systems are at a big risk, since anyone on tile Internet can initiate a connection to them, thus making them more likely to be the target for hostile activity. Detection of such an activity is very critical. These systems have a high production value, so data generated within DMZ is very voluminous and chances of false positives are also very high. Just by putting a Honeypot into DMZ will help to detect any abnormal behavior. The Honeypot in DMZ will have no production value, any connection made to it, is an alarm of malicious activity. It could provide a very helpful mechanism to detect any outbound traffic originating from the email or web server themselves. A typical placement diagram of using a honeypot in DMZ is shown ill Figure 1.

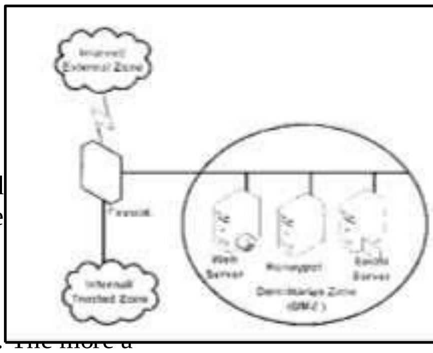


Figure 1: Honeypot Deployment in Demilitarizes Zone (DI\IZ)

If any activity is detected from tile email or webserver targeting DMZ-honeypot, it is und these server(s) got compromised and are being used as slave(s) to scan other system(s) on the vulnerabilities.

1.2 TRADEOFFS BETWEEN LEVELS OF INTERACTION

Level of Interaction is one such metric which can be used to measure and compare Honeypots. honeypot can do and the more an attacker can do to a honeypot, the greater the information that call be derived from it. However,more the interaction happens more are chances of getting potentially damaged by the attacker.

2 Low Level ofInteraction

Low-interaction honeypots are easy to install and they emulate few services. Attackers can scan and connect to various ports. Information gathering is very little; also attacker is limited to interact with the honeypot only. Low-interaction honeypots are primarily production honeypots that, are used to help protect an organization. There is no service on time system for the attacker to log-in, i.e., attacker is limited to interact with predestinated services that too at an interface level only. The primary value of such honeypots is to detect unauthorized scans or unauthorized connection attempts. Deployment and maintenance of such honeypot is relatively easier than other kinds of honeypots. These honeypots have low-level of risk association. Risk is low because attacker has been given limited leverage to explore and compromise. Low- interaction honeypots are limited to transactional information about, the attack, small or almost negligible amount of information is available for the attack itself. It could simply provide the researcher with rough signature of an attacklike:

- Time and date of attack.
- Source IP address and source port of

Honeypots with high level of interaction give vast amount of information about the attack, attacker and their intents. They exhibit high-level of risk and are very difficult to build and maintain. The goal of high interaction honeypots is to give the attacker access to real operating system where nothing is emulated or restricted. These honeypots greatly help to uncover tools, techniques and tactics of the black-hat community. These can discover new tools: identify new vulnerabilities in operating systems and/or applications and help to track unknown(s). Though these high interaction variants are very useful tools and exhibit numbers of possibilities to uncover motives of hackers, these honeypots are at immense level of risk. As once an attacker gained access to the system and it is compromised, little can be done to curtail it. Most of these honeypots are placed in controlled environment such as behind a reverse firewall, which allows attacker to interact and launch attack on the honeypot being posed as a production system. But will not allow launching attacks from this compromised system. Because of these complex tasks, these honeypots are extremely difficult to build and maintain. Tradeoffs of Honeypot levels of interaction are shown in Table1
the attack. and

•

port of the attack.

Destination IP address and destination

2.1 Medium Level ofInteraction

Medium interaction honeypots offer attackers more ability to interact than low interaction honeypots. A honeypot with this characteristic is designed to act beyond just making a connection at specific port. For example, apart from emulating a service at a specific port it can go further to emulate behaviour with respect to specific adversaries available on the security community lists. This customized behavioural exhibition of the honeypot, makes attacker believe that it as a production system. Thus data capture through these honeypots makes much more revelations about the actualattack.

The concept is to jail (bound) the attacker to all extent that it can not harm the system. On the other hand it gives security analyst enough information to capture the payload and analyze the attack. These are difficult to implement as compared to low level honeypots. These honeypots are more time consuming to install and configure amid require much more interaction and know-how to install. Medium level of interaction honeypots involves high level of customization arid development effort by the network security administrators. Risk involved is also higher as compared to low-interaction honeypots.

2.2 High Level ofInteraction

Level of Interaction	Installation and Confirmation	Development of Maintenance	Information Gathering	Ris Le
Low	Easy	Easy	Limited	Lo
Medium	Involved	Involved	Variable	Me
High	Difficult	Difficult	Extensive	Hig

.

Table 1: Tradeoffs of Honeypots: Levels of Interaction

3 Honeypots: Exploration andAnalysis

In order to better explore and analyze honeypots technology, during this work a test-bed has been configured ax shown in Figure 2 many honeypots were deployed and explored, as detailed in following sections.

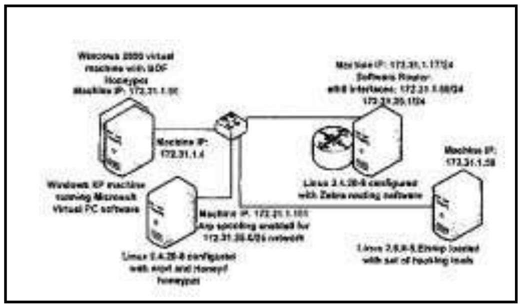

Windows 2000 virtual machine with BOF Honeypot
Machine IP: 172.31.1.4
Windows XP machine running Microsoft Virtual PC software
Software Router
Linux 2.4.20-8 configured with Zebra routing software
Machine IP: 172.31.1.58
Arp spoofing enabled for
Linux 2.4.20-8 configured with arpd and Honeyd honeypot
with set of hacking tools

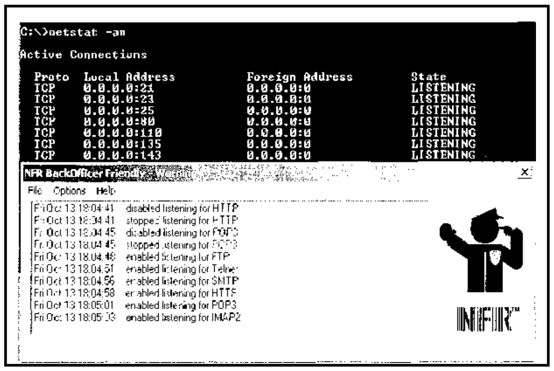
Figure 2: Test Bed for Honevpots Exploration and Analysis

3.1 Backofficer Friendly(Bof)

BOF is a low-interaction honeypot designed to run out of the box on all windows platform; Unix version needs to be compiled and then run. It is easy to install and configure; also as it falls under low-interaction category, its capabilities are limited. BOF call monitor up to seven emulated services. There is almost no customization option available. It is very limited feature honeypot. In this work, BOF was deployed on a windows 2000 virtual machine and attack was launchedfrom
172.31.1.4 (windows xp) and 172.31.1.17 (Redhat Linux 2.4.20-8). Following were the findings/observations from BOFdeployment:

- Primary purpose of this low interaction honeypot is to act as a burglar alarm, alerting whenever something was scanning a.System.
- Whenever a connection is made to any one of the seven services, the attempt is logged and an alert. Isgenerated.
- If an attack is made to any other port BOF remains unaware of any maliciousactivity.

BOF provides very little value to incident response. as interaction is very limited and restricted only to scan detections. BOF can he used as a limited research tool for trend analysis purposes over a period of time, but again this will have only a transactional value. Figure 3 shows a BOF honeypot running on and detecting the scans.

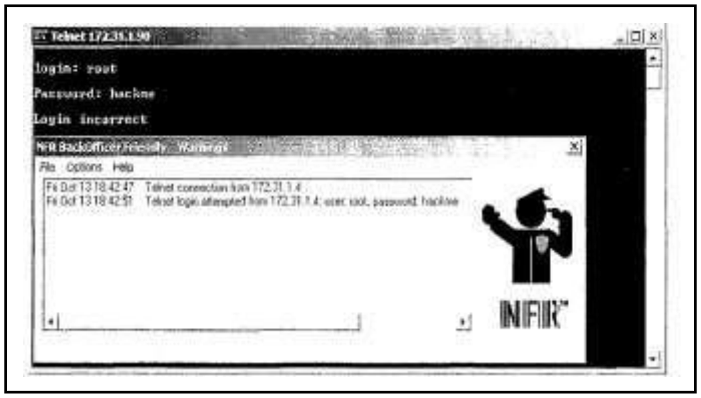


```
C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:23             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:25             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:110            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:143            0.0.0.0:0              LISTENING
```

```
Fri Oct 13 18:04:41   disabled listening for HTTP
Fri Oct 13 18:04:41   stopped listening for HTTP
Fri Oct 13 18:04:45   disabled listening for POP3
Fri Oct 13 18:04:45   stopped listening for POP3
Fri Oct 13 18:04:49   enabled listening for FTP
Fri Oct 13 18:04:51   enabled listening for Telnet
Fri Oct 13 18:04:56   enabled listening for SMTP
Fri Oct 13 18:04:58   enabled listening for HTTP
Fri Oct 13 18:05:01   enabled listening for POP3
Fri Oct 13 18:05:03   enabled listening for IMAP2
```

2. FTP: File Transfer Protocol, listening on port21.
3. Telnet: Listening on i)Ort23.
4. SMTP(25)
5. HTTP(80): BOF does not offer any functionality on port(443) used as SSLport.
6. POP3 (TCP,11O),and
7. IMAP, port143.

BOF also otters fake replies, hut this capability is also hunted and easily guessable by the attacker. For example, Figure 4.4 shows an attempt to telnet, a login and password response constitutes a fake reply, hut it was observed BOF accepts and shows even password as clear text. Fake replies do not show any http banner but only logs this activity.



```
login: root
Password: hackme
Login incorrect
```

```
Fri Oct 13 18:42:47   Telnet connection from 172.31.1.4
Fri Oct 13 18:42:51   Telnet login attempted from 172.31.1.4; user: root, password: hackme
```

Figure 4: BOF Honevpot Telnet fake replies There is no way of remote administration of BOF and it also does not send the alert notification remotely, thus further limiting its capabilities. BOF can he identified by fingerprinting its service. Fingerprinting can be clone by collecting network transactional data and analyzing it.

Figure 3: Back Officer Friendly Honeypot Running and Emulating services

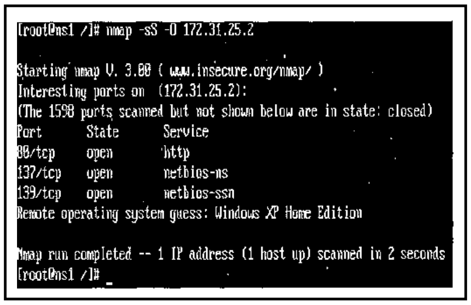
Working of BOF

BOF works by creating open sockets, which bind to a specific set of ports. When a connection is made to the port, port listeners through three-way handshake process: logs tile attempt, generates an alert, and closes the connection. BOF offers following seven services:

1. Back Orifice: A windows-based trojan, listening on port UDP31337.

4.3.2 Honeyd

Honeyd is designed as a low-interaction honeypot. It offers emulated services on a Unix platform. It is used to detect attacks or an unauthorized activity. Since it is Open Source, it is highly customizable and new service emulated can be developed. Honeyd detects activity on any TCP port and the emulated services help to deceive attackers and capture their activities. It can assume the personality of any operating system, and can he configured to offer different TCP/IP "services" like HTTP, SMTP, SSH, telnet etc. Honeyd is used in honeynet research typically for setting up virtual honeypots to engage an attacker. Honeyd basically works in a virtual domain, by using unallocated IP addresses. It can monitor millions ofnonexistent IP addresses for connections. Honeyd assumes an identity of the system by a sample configuration file and listens on a specific IP address. It can emulate many operating systems at the same time. One of the major advantage of Honeyd is that it not only emulates services hut also emulates IP Stack for different personality of operating systems. This feature helps to deceive an attacker by offering exact operating system attack though the system is fake. Figure 4.5 shows result of nmap (fingerprinting tool) against honeyd, default template used was for Windows XPmachine.



```
[root@ns1 /]# nmap -sS -O 172.31.25.2

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (172.31.25.2):
(The 1598 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
137/tcp    open        netbios-ns
139/tcp    open        netbios-ssn
Remote operating system guess: Windows XP Home Edition

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
[root@ns1 /]# _
```

Figure 5: Nmap output: Attacker scanningHoneyd defaulthost

Honeyd is able to simulate an entire network topology within one machine with multiple hops, packet losses and latency. This would simulate complex networks. It could also present a make-believe network to an attacker who gets snared in a honeynet. Some of the major features available in Honeyd are as follows:

- Nniap and X fingerprint database signaturemapping
  Service emulation of various services Open source and easily customizable Simulation of
- large network
topologies
-
- Configurable network characteristic like latency andbandwidth
- Supports multiple entry routers to serve multiplenetworks
- Integrates physical machines into the networktopology
  Asymmetric routing
- GRE tunneling for setting up distributed networks
- main operating systempersonalities

Working of Honeyd

When an IP address of a nonexistent system is attacked, honeyd assumes the identity of the victim and interacts with the attacker. Making itself as a victim is the key point which makes it possible to track the malicious activity. Emulated services are only limited to TCP, no UDP service is available. Also ICMP service is for echo requests and reply only.

If there is a network that has no production system, that entire network is directed to the honeyd honeypot. This is called „black holing" which is powerful technique for the measurement of automated network wide phenomena, such as globally targeted internet worms or scans. In this work, both black holing and ARP spoofing has been used to explore the working of honeyd.

IP route 172.31.25.0 255.255.255.0
172.31.25.1

The whole traffic for the 172.31.25.0 network is directed towards the honeydhoneypot.

In *arp spoofing method*, honyed depends upon Arpd utility. Ethernet uses MAC identifier (48 bit) to recognize any system on the network. The first three octets represent the manufacturer and last three are unique identifier for the network interface card (NIC). In order to reach the destination, system must know the MAC address. Every system keeps an ARP table for this purpose. When packet reaches the network of the destination system, the ARP table is checked and then the packet, is sent to the respective system. If system does not find entry in the ARP table it, asks the network for the same. This constitutes a ARP <who-has tell > request.

Arpd is run on the same system as Honeyd i.e. IP 172.31.1.1.51 in this case. Arpd watches all the traffic on the network. Now when the attacker attempts to connect to a system which is not available on the local network. Arpd will then send an ARP reply back, saying that the MAC address of the Honeyd belongs to the nonexistent ip address. Attacker now sends the attack string which is captured by honeyd honeypot. In this way, attacker will never realize that attack string is being sent to nonexistent system but being handled by a Honeyd honeypot via arpspoofing.

An attacking system 172.31.1.4 connects to TCP port 80, honeypot initiates a web server emulator and interacts with the attacker, thus capturing all activities. Honeyd also demonstrates its capability to fool fingerprinting tools like Nmap and X. Nmap is one of the most common tools used to fingerprint an operating system. It sends certain packets to the target and compares the results with the database of known signatures. Honeyd uses the same database(s) nmap.assoc and mnap.prints to reply against fingerprinting tools. This means if Honeyd is emulating window 2000 and it is fingerprinted by Nmap, Honeyd will respond with Windows 2000 signatures and the attacker is deceived in thinking that attack is targeted towards Windows 2000. Honeyd was configured with followingtemplate:

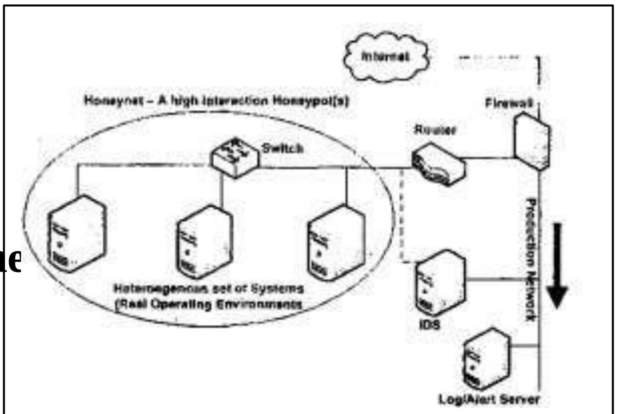
```
        Create windows set windows personality "Microsoft Windows 2000 Server
SP2"
```

```
        set   windows   default   TCP   action
reset
        add   windows   TCP   port   80   "pen
/tools/honeyd/scripts/iis/main.pl" bind 172.31.25.10 windows
```

Each template represents a working personality, it could be an operating system like Windows 2000 or a network device like Cisco router. This determines how the system will behave at the IP stack level. IP stack behavior is associated with NMAP fingerprint database as shown below for Microsoft window 2000SP2:

```
        Fingerprint Microsoft Windows 2000 Server SP2 Class Microsoft | Window | NT/2K/XP
              |    general    purpose    TSeq (Class=RI%gcd=<6%SI=<25224&>22C%IPID=I) T1
(DF=Y%W=5B4|B68%ACK=S++%Flags=AS%Ops=MNNT) (Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)T3
        T4(DF=N%W=0%ACK=O%Flags=R%Ops=)
        T5 (DF=N%W=0%ACK=S++%Flags=AR%Ops=) T6(DF=N%W=0%ACK=O%Flags=R%Ops=)
T7  (DF=N%W=0%ACK=S++%Flags=AR%Ops=)PU (DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%
RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Where Tseq is the TCP sequenceability test T1 is a SYN packet with a bunch of TCP options to open port. T2 is a NULL packet w/options to open port. T3 is a SYN|FIN|URG|PSH packet w/options to open port. T4 is an ACK to open port w/options. T5 is an SYN to closed port w/options. T6 is an ACK to closed port w/options. T7 is a FIN|PSH|URG to a closed port w/options. PU is a UDP packet to a closed port. Following options are allowed when setting the action part on a particularport:
Reset this means that Honeyd will send TCP reset (RST) reply for TCP connections. This emulates a closed port. Open this means Honeyd will acknowledge the connection on this port, as if service is open for the network access. Block Honey will chop and ignore all connections to the port thus emulating a firewall behaviour. Script This is limited to TCP services and will call a script to emulate the service and interact with the attacker. Information gathering can be done with the help of two methods: syslogd and/or a sniffer. By default, Honeyd logs all TCP and ICMP attempts to syslogd daemon. This information is limited to transactional aspect only and gives united view into the actual attack. In addition to this, Honeyd service emulation scripts can have a logging capability. As Honeyd is open source product it is easily customizable to include more logging capability, thus improving its utility. Secondly, sniffer can he used to capture the network traffic interacting with the honeypot. Connection too many TCP popular services sent data ill clear text likes of Telnet, FTP, HTTP, this captured data is very helpful in further investigating theattack.


Internet
Honeynet – A high Interaction Honeypot(s)
Firewall
Router
Switch
Production Network
Heterogeneous set of Systems
(Real Operating Environments)
IDS
Log/Alert Server

Captured data analysis will produce a unique attack signature. Honeyd has no built-in notification mechanism, so a separate solution must be used. Honyed being a low interaction honeypot introduces limited riskfactor.

3.3 Honeynets

Honeynets are high-interaction honeypots. No services are emulated, and no caged environments are created. Real systems are offered to the attacker behind some access control device. The system configuration can be heterogeneous i.e. the systems within a Honeynet are true production systems. Honeynets are very flexible tool. Honeynets deceive attackers, detect attacks and capture the unknown. Honeynets require an extensive amount of time and resources to build, implement and maintain. This technology adds tremendous value as research honeypots. These are used mainly to address following security concerns:

Who are the attackers? What tools theyuse?

What tactics do they employ? What motivatesthen

Honeynets can collect in-depth information
about the attackers, such as their keystrokes when they compromise the system, their chat sessions with their peers, the tools they used to probe and exploit, vulnerable systems. As research honeypots, Honeynets also excel at trend analysis and statistical modeling. The information gathered can be used to predict attacks, acting as an early warning system.

Working of Honeynets

A Honeynet is constituted as a network of multiple systems. It is a self-contained environment with three critical elements: data control, data capture and data collection. Data control is the controlling of the blackhat activity. Once blackhat takes control of a honeypot within the honeynet, activity needs to be controlled so that attacker can not harm any non honeynet systems. Data capturing is capturing of all the activity that occurs within the honeynet. Data collection is the aggregation of all the data captured by multiple honeynets. Figure 4.6 shows the basic architecture of a Honeynet. In the test lab under this work, various honeynet generations were deployed and analyzed. Data captured and analysis is presented in this section.

Figure 6: Basic Architecture of Honeynet.

Firewall machine provides Data control features and IDS machine provides data capture feature apart from the Log server which receive data from the Honeynet using a covert channel. Covert Channel setup in the research lab was done using the sebek technology. Where data is sent by Honeynet to the log server using UDP port 1101. In the test environment setup for this work, firewall is configured using three network interface, one for the honeynet, one for the Internet connectivity and other for production network. IDS machine has two interfaces, one interface has been given an IP address while other is kept IP-less which is being used for sniffing purposes, to record the network activity, and this gives a stealth interface for data capture. Three victim machines, Linux 2.4.x.Linux
2.6.x and Windows 2000 were installed with default configurations. Honeynet data control at firewall level provides connection blocking as well as connection limiting functionality. Data Capture in a Honeynet is categorized into following four categories:

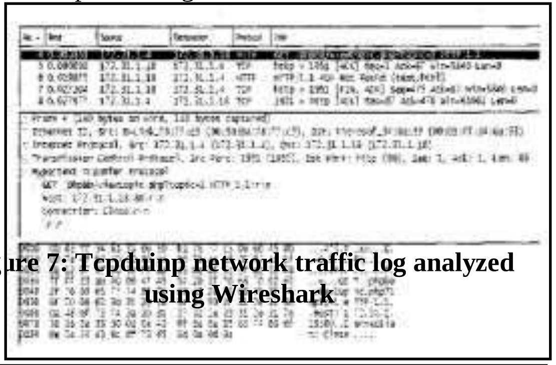
Network transactionrecording Network trafficrecoding
Host activity recording IDS alerts

Network transactions occurring in the honeynet include inbound communication and connection attempts from the Internet, internal connections between the machine within the honey net and the outbound communication initiated by the honey net. Outbound connection front the honey net is a decisive indicator of the hostile activity. Network traffic recording provides maximum level of details on the intruder activities. Host activity recording includes the recording of the attacker"s keystrokes and other host process communications. Host logs are extremely useful for analyzing attack traces. Finally, IDS alerts add structure to network traffic analysis and allow to take action based on what is going on in the honeynet. Linux IP Tables was used in time experimental setup for network transaction recording. Following are the modules loaded on the firewall machine eth0 172.31.1.17. eth1 202.164.55.99, eth2 Microsoft loopback adapter (used for remote management purposes). Table 4.2 shows various IP table Log entries and their respectivemeaning.

Oct 15 18:31:24	Syslog Date-time Stamp
ns1	Hostname of the log producing machine
Kernel:	System kernel
INBOUND ICMP/TCP	Log comment
IN=eth1:	Network interface for incoming packets
IN=eth0:	Network interface on which packet is forwarded
SRC=202.164.55.101 DST=172.31.1.50	Source and Destination address
SPT=1801 and DPT=80	Source and Destination port address

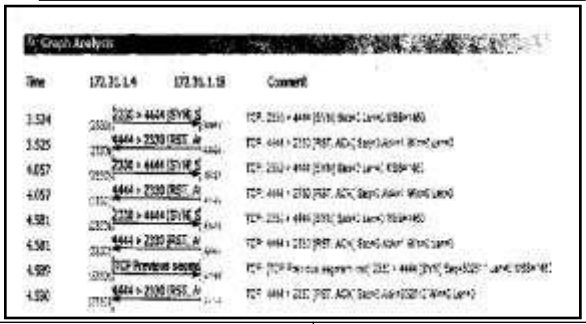
Table 2: IP Tables Log entries and theirmeaning

TCP log shown above shows IP address 202.164.55.101 connecting to the machine 172.31.1.50 at port number 80 i.e. http service. This is log snippet of connection initiation phase as can be seen from the SYN bit of three ways handshake is set on. Figure 4.7 Shows the tcpdump collected at IDS machine, showing phpBB attack. This exploits two arbitrary PEP code execution flaws in the phpBB forum system. The problem is that tire „highlight parameter in the "viewtopic.php" script is not verified properly and will allow an attacker to inject arbitrary code via preg _replace(). Figure 4.8 shows flow graph of the attack signatures capture usingsnort.



[illegible]

Figure 7: Tcpduinp network traffic log analyzed using Wireshark



```
3.534   TCP: 2330 > 4444 [SYN] [illegible]
3.525   TCP: 4444 > 2330 [RST, ACK] [illegible]
4.057   TCP: 2330 > 4444 [SYN] [illegible]
4.057   TCP: 4444 > 2330 [RST, ACK] [illegible]
4.581   TCP: 2330 > 4444 [SYN] [illegible]
4.581   TCP: 4444 > 2330 [RST, ACK] [illegible]
4.589   TCP: [TCP Previous segment lost] [illegible]
4.590   TCP: 4444 > 2330 [RST, ACK] [illegible]
```

Entry	Meaning

Figure 8: Flow graph statistics of an attack

One drawback with the GenI honeynet is that it is easy to get detected which provides a minimal capability to study the attacks. Limitations are clue to the restricted number of allowed outgoing connections from the honeynet and the use of layer 3 communications. GenII honeynets provide morestealthy operation. In generation II honeynets data control and data capture are implemented on a single device, called Honeywall. This architecture also provides new keystroke logging running at both honeywall and honeynet. These advances lower the possibility of honeynets being detected by blackchats, lower the risk of losing data, counteract encrypted communication on the honeypots and provide a glass-box monitoring tool about the honeypot‟s maternalstate.

4.4 CONCLUSIONS

This chapter concentrates on the analysis of opensource Honeypots and demonstrates their use in the network security hierarchy. Tradeoffs between levels of interactions are reported. Complete details opensource honeypots including Back officer Friendly, Honeyd and Honeynets are clemoiistratec1 by setting up these at the workplace. It was found that by taking advantage of virtualization software like Microsoft Virtual PC (as is done in the experimental setup) physical requirements of setting up a honcynet can be greatly reduced. These virtual honeynets allow to run this proactive security technology more efficiently. This chapter achieves second objective of the thesis work. Concepts of self-containment, covert channel communication, data control and data capture while maintaining the internal state of the honeynet is used extensively ill the proposed framework, which is elaborated further in the next chapter of this thesiswork.