Journal of Advances and Scholarly Researches in Allied Education
Vol. 15, Issue No. 11, November-2018, ISSN 2230-7540

A Proactive Network Security Technology for Managing It Infrastructure Using Honeynets

Saurabh Kawatra[1]* Dr. Raghav Mehra[2]

[1] HOD (Computer Science), Mata Jeeti Ji Girls College, Suratgarh, Rajasthan

[2] Associate Professor & Dean at Bhagwant Institute of Technology, Muzzafar Nagar (UP)

Abstract – In this paper we survey the ongoing advances in honeypot. A few striking recommendations and there investigation have been talked about. The parts of utilizing honeypot in training and in half and half condition with IDS have been clarified. In this paper we likewise characterize the utilization of mark system in honeypot for activity investigation. In the end we condenses every one of these viewpoints.

Keywords: Honeypot, Honeynet, IDS, Load Balancer and Honeywall.

----------X----------

INTRODUCTION

Honeypot technology, its underlying foundations, chronicled foundation and different ages has just been talked about in detail while doing writing survey in second section. in this part the work is moved into dissecting the open source honepots, in the accompanying area. The utilization of Honeypot in network security is accentuated and their place in the network security chain of command has been examined.

1. USE OF HONEYPOT IN NETWORK SECURITY

Honeypots enhance recognition of unauthorized action on the network. Three regular difficulties of recognition are false positives. *False negatives and data aggregation*. False positives are when frameworks erroneously ready doubts or pernicious action, i.e., framework may decipher substantial network activity as au attack. Numerous a period these sorts of alarms prompt an obliviousness factor shown by executive after a flood of false positives they may disregard the real attack stream. False negatives are the point at which an Interaction neglects to distinguish attacks.

Network interruption location frameworks confront a test of false positives as well as have issues with false negatives. Numerous NIDS frameworks, regardless of whether they depend on mark databases. Convention confirmation or some other approach, can conceivably miss new or obscure attacks. Before databases get refreshed, another attack apparatus can complete an incredible damage.

Other real issue with responsive strategy of barrier is Data total; NIDS, framework logs, application logs, firewall logs and so forth catch huge amounts of data. Ana1ysi& of this data to reveal the pernicious goal resembles seeking in nature.

Honeypots address all these three difficulties adequately. Most Honeypots have no creation movement, so there is little action to produce false positives. In a large portion of the cases, aside from mis-setup, Honeypots create legitimate cautions, enormously decreasing False positives. One of the essential advantages is that Divert can recognize another attack by prudence of the framework action not marks. Honeypots create just a few megabytes of data daily, the greater part of which is of high esteem. This makes it greatly simple to analyze valuable data from the Honeypot.

1.1 Using Honeypots in the DeMilitarized Zone (DMZ)

DMZ is a network of untrusted frameworks ordinarily used to give services to the Web, for example, email or web server. These frameworks are at a major hazard, since anybody on tile Web can start an Interaction with them, therefore making them more inclined to be the objective for threatening action. Location of such an action is extremely basic. These frameworks have a high creation esteem, so data produced inside DMZ is extremely voluminous and odds of false positives are likewise high. Just by putting a Deflect into DMZ will identify any unusual conduct.

The Deflect in DMZ will have no generation esteem, any Interaction made to it, is a caution of

vindictive action. It could give an extremely supportive network to recognize any outbound activity beginning from the email or web server themselves

A normal arrangement outline of utilizing a Divert in DMZ is demonstrated sick Figure 4.1.

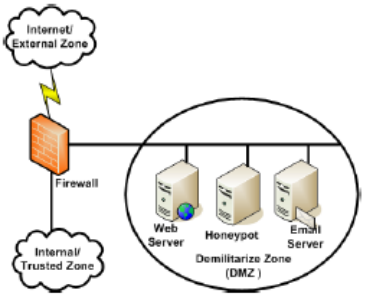

Figure 4.1: Honeypot Deployment in DeMilitarized Zone (DMZ)

In the event that any movement is distinguished from tile email or webserver focusing on DMZ-Honeypot, it is comprehended that these server(s) got traded off and are being utilized as slave(s) to filter different network(s) on the Web for vulnerabilities.

2. TRADEOFFS BETWEEN LEVELS OF INTERACTION

Level of Interaction is one such metric which can be utilized to gauge and think about Honeypots. The more a Divert can do and the more an attacker can do to an Honeypot, the more prominent the data that call be gotten from it. Be that as it may, increasingly the Interaction happens more are odds of getting possibly harmed by the attacker.

2.1 Low Level of Interaction

Low-Interaction Honeypots are anything but difficult to introduce and they copy few services. Attackers can output and interface with different ports. Data gathering is practically nothing; likewise attacker is constrained to collaborate with the Divert as it were.

Low-Interaction Honeypot are basically generation Honeypot that, are utilized to help secure an Interaction. There is no administration on time framework for the attacker to sign in, i.e., attacker is restricted to collaborate with predestinated services that too at an interface level as it were.

The essential estimation of such Honeypot is to recognize unauthorized Honeypotss or unauthorized Interaction endeavors. Arrangement and support of such Divert is generally less demanding than different sorts of Honeypots. These Honeypot have low-level of hazard affiliation. Hazard is low since attacker has been given restricted use to investigate and trade off. Low-Interaction Honeypots are restricted to value-based data about, the attack, little or relatively unimportant measure of data is accessible for the attack itself. It could basically furnish the analyst with unpleasant mark of an attack like:

- Time and date of attack.
- Source IP address and source port of the attack. what's more,
- Destination IP address and goal port of the attack.

2.2 Medium Level of Interaction

Medium Interaction Honeypots offer attackers more capacity to communicate than low Interaction Honeypots. An Honeypot with this trademark is intended to act past simply making an Interaction at particular port. For instance, aside from copying an administration at a particular port it can go further to imitate conduct as for particular foes accessible on the security network records. This altered conduct show of the Divert, influences attacker to trust that it as a generation framework. In this manner data catch through these Honeypots makes considerably more disclosures about the genuine attack. The idea is to imprison (bound) the attacker to all degree that it cannot hurt the framework. Then again it gives security examiner enough data to catch the payload and break down the attack. These are hard to actualize when contrasted with low level Honeypot. These Honeypots are additional tedious to introduce and design in the midst of require significantly more communication and know-how to introduce. Medium level of Interaction Honeypots includes abnormal state of customization bone-dry advancement exertion by the network security managers. Hazard included is additionally higher when contrasted with low-Interaction Honeypots.

2.3 High Level of Interaction

Honeypot with abnormal state of Interaction give tremendous measure of data about the attack, attacker and their purposes. They display abnormal state of hazard and are extremely hard to assemble and keep up. The objective of high communication Honeypots is to give the attacker access to genuine working framework where nothing is copied or limited. These Honeypots extraordinarily help to reveal apparatuses, procedures and strategies of the dark cap network. These can find new devices: recognize new vulnerabilities in working frameworks as well as applications and help to track unknown(s). In spite of the fact that these high Interaction variations are exceptionally helpful devices and display quantities of conceivable outcomes to reveal thought processes of programmers, these Honeypots are at gigantic level of hazard. As once an attacker accessed the framework and it is imperiled, little

should be possible to shorten it.Most of these Honeypot are put in controlled condition, for example, behind a turnaround firewall, which enables attacker to communicate and dispatch attack on the Honeypot being acted like a creation framework. Be that as it may, won't permit propelling attacks from this bargained framework. Due to these intricate assignments, these Honeypots are amazingly hard to assemble and keep up. Tradeoffs of Honeypot levels of communication are appeared in Table 4.1.

Table 4.1: Tradeoffs of Honeypot: Levels of Interaction

Level of Interaction	*Installation and Configuration*	*Deployment and Maintenance*	*Information Gathering*	*Risk Level*
Low	Easy	Easy	Limited	Low
Medium	Involved	Involved	Variable	Medium
High	Difficult	Difficult	Extensive	High

3. HONEYPOTS: INVESTIGATION AND ANALYSIS

Keeping in mind the end goal to more readily investigate and break down Honeypot technology, amid this work a test¬bed has been designed hatchet appeared in Figure 4.2. Many Deflect were sent and investigated, as point by point in following areas.

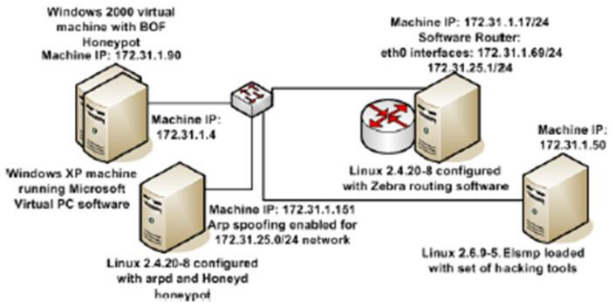

Figure 4.2: Test Bed for Honeypots Exploration and Analysis

3.1 Back Officer Friendly (BOF)

BOF is a low-Interaction Honeypot intended to come up short on the crate on all windows stage; Unix form should be aggregated and after that run. It is anything but difficult to introduce and arrange; likewise as it falls under low-Interaction classification, its abilities are restricted.

BOF ring screen to seven imitated services. There is no customization choice accessible. It is exceptionally restricted component Divert.

In this work, BOF was conveyed on a windows 2000 virtual machine and attack was propelled from 172.31.1.4 (windows xp) and 172.31.1.17 (Redhat Linux 2.4.20-8). Following were the discoveries/perceptions from BOF arrangement:

- Primary reason for this low communication Divert is to go about as a thief caution, alarming at whatever point something was Honeypotsing a. Framework.
- Whenever an Interaction is made to any of the seven services, the endeavor is logged and a caution. Is created.
- If an attack is made to any other port BOF stays unconscious of any vindictive action.

BOF gives next to no an incentive to episode reaction. as communication is exceptionally constrained and limited just to Honeypots identifications. BOF can be utilized as a constrained research device for pattern examination purposes over some undefined time frame, however again this will have just a value-based esteem. Figure 4.3 demonstrates a BOF Divert running on and distinguishing the sweeps.

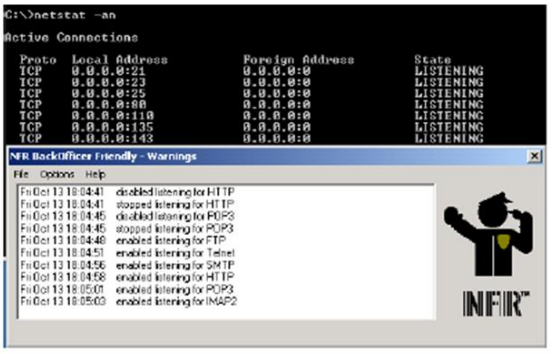

Figure 4.3: Back Officer Friendly Honeypot Running and Emulating services

Working of BOF

BOF works by making open attachments, which tie to a particular arrangement of ports. At the point when an Interaction is made to the port, port audience members through three-way handshake process: logs tile endeavor, produces a caution, and shuts the Interaction. BOF offers following seven services:

1. Back Hole: A windows-based trojan, tuning in on port UDP 31337.
2. FTP: Record Exchange Convention, tuning in on port 21.
3. Telnet: Tuning in on i)Ort 23.
4. SMTP (24)
4. HTTP(80): BOF does not offer any functionality on port(443) used as SSL port.
6. POP3 (TCP,11O), and
7. IMAP, port 143.

BOF likewise otters counterfeit answers, cottage this ability is additionally chased and effectively guessable by the attacker. For instance, Figure 4.4 demonstrates an endeavor to telnet, a login and secret key reaction establishes a phony answer, cottage it was watched BOF acknowledges and indicates even secret word as clear content. Counterfeit answers don't demonstrate any http flag yet just logs this movement.

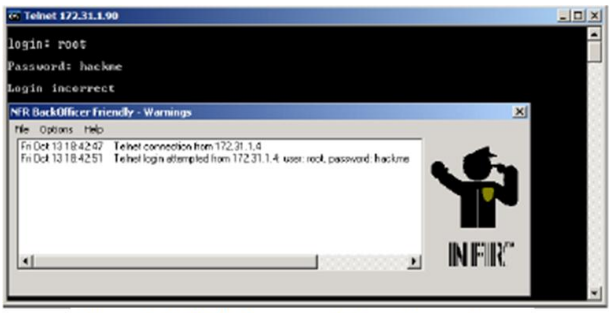

Figure 4.4: BOF Honeypot Telnet fake replies

There is no chance to get of remote organization of BOF and it likewise does not send the ready warning remotely, subsequently additionally restricting its capacities. BOF can be distinguished by fingerprinting its administration. Fingerprinting can be clone by gathering network value-based data and investigating it.

3. 2 Honeyd

Honeyd is outlined as a low-Interaction Divert. It offers copied benefits on a UNIX stage. It is utilized to identify attacks or an unauthorized action. Since it is Open Source, it is very adaptable and new administration imitated can be created. Honeyd identifies movement on any TCP port and the imitated services help to hoodwink attackers and catch their exercises. It can accept the identity of any working framework, and can be arranged to offer extraordinary TCP/IP "services" like HTTP, SMTP, SSH, telnet and so forth. Honeyd is utilized in honeynet inquire about ordinarily to set up virtual Honeypots to connect with an attacker.

Honeyd fundamentally works in a virtual area, by utilizing unallocated IP addresses. It can screen a great many non-existent IP addresses for Interactions. Honeyd accept a personality of the framework by an example arrangement document and tunes in on a particular IP address. It can copy many working frameworks in the meantime. One of the significant preferred standpoint of Honeyd is that it not just copies services hovel additionally imitates IP Stack for various identity of working frameworks. This element cheats an attacker by offering definite working framework attack however the framework is phony. Figure 4.4 shows aftereffect of nmap (fingerprinting device) against honeyd, default layout utilized was for Windows XP machine.

create default set default personality “Microsoft Windows XP Home Edition”

Set default tcp action reset

Set default udp action reset

Set default icmp action open

add default tcp port 88 “sh scripts/misc/test.sh”

add default tcp port 139 open

add default tcp port 137 open

add default udp port 137 open

add default udp port 135 open

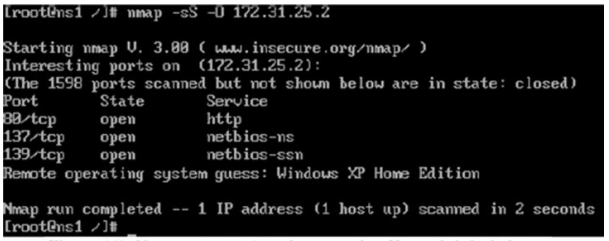

Figure 4.5: Nmap output: Attacker scanning Honeyd default host

Honeyd can reenact a whole network topology inside one machine with numerous jumps, parcel misfortunes and inertness. This would reenact complex networks. It could likewise show a pretend network to an attacker who gets trapped in a honeynet. A portion of the significant highlights accessible in Honeyd are as per the following:

- Nniap and X unique mark database signature mapping
- Service imitating of different services
- Open source and effortlessly adjustable
- Simulation of huge network topologies
- Configurable network trademark like idleness and bandwidth
- Supports numerous passage switches to serve different networks
- Integrates physical machines into the network topology
- Asymmetric steering
- GRE burrowing for setting up disseminated networks
- main working framework identities

Working of Honeyd

At the point when an IP address of a nonexistent framework is attacked, honeyd expect the character of the person in question and cooperates with the attacker. Making itself as a casualty is the key point which makes it conceivable to track the pernicious action. Copied services are just restricted to TCP, no UDP benefit is accessible. Likewise ICMP benefit is for resound demands and answer as it were.

On the off chance that there is a network that has no creation framework, that whole network is coordinated to the honeyd Deflect. This is called 'dark holing" which is great procedure for the estimation of mechanized network wide marvels, for example, all inclusive focused on web worms or outputs. In this work, both dark holing and ARP spoofing has been utilized to investigate the working of honeyd.

IP highway 172.31.24.0 244.244.244.0 172.31.24.1

The entire movement for the 172.31.24.0 network is coordinated towards the honeyd Deflect.

In arp satirizing technique, honyed relies on Arpd utility. Ethernet utilizes Macintosh identifier (48 bit) to perceive any framework on the network. The initial three octets speak to the producer and last three are special identifier for the network interface card (NIC). So as to achieve the goal, framework must know the Macintosh address. Each framework keeps an ARP table for this reason. At the point when parcel achieves the network of the goal framework, the ARP table is Honeypotsed and after that the bundle, is sent to the individual framework. In the event that framework does not discover passage in the ARP table it, approaches the network for the equivalent. This establishes an ARP <who-has tell > ask.

Arpd is kept running on indistinguishable framework from Honeyd i.e. IP 172.31.1.1.41 for this situation. Arpd observes all the movement on the network. Presently when the attacker endeavors to associate with a framework which isn't accessible on the neighborhood arrange. Arpd will then send an ARP answer back, saying that the Macintosh address of the Honeyd has a place with the nonexistent ip address. Attacker presently sends the attack string which is caught by honeyd Deflect. Along these lines, attacker will never understand that attack string is being sent to nonexistent framework yet being taken care of by a Honeyd Deflect by means of arp spoofing.

An attacking framework 172.31.1.4 interfaces with TCP port 88, Deflect starts a web server emulator and communicates with the attacker, accordingly catching all exercises. Honeyd likewise exhibits its ability to trick fingerprinting devices like Nmap and X. Nmap is a standout amongst the most well-known devices used to unique finger impression a working framework. It sends certain parcels to the objective and contrasts the outcomes and the database of known marks. Honeyd utilizes the equivalent database(s) nmap.assoc and mnap.prints to answer against fingerprinting instruments. This implies if Honeyd is copying window 2000 and it is fingerprinted by Nmap, Honeyd will react with Windows 2000 marks and the attacker is cheated in imagining that attack is focused towards Windows 2000. Honeyd was arranged with following layout:

Make windows set windows identity "Microsoft Windows 2000 Server SP2" set windows default TCP activity reset include windows TCP port 88 "pen/devices/honeyd/contents/iis/main.pl" tie 172.31.24.10 windows

Every layout speaks to a working identity, it could be a working framework like Windows 2000 or a network gadget like Cisco switch. This decides how the framework will act at the IP stack level. IP stack conduct is related with NMAP unique finger impression database as appeared underneath for Microsoft window 2000 SP2:

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)

T2
(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)

T3
(Resp=Y%DF=Y%W=5B4|B68%ACK=S++%Flags=AS%Ops=MNNT)

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 choices are permitted when setting the activity part on a specific port:

- Reset this implies that Honeyd will send TCP reset (RST) answer for TCP Interactions. This imitates a shut port.
- Open this implies Honeyd will recognize the Interaction on this port, as though benefit is open for the network get to.
- Block Honey will cleave and disregard all Interactions with the port hence imitating a firewall conduct.
- Content This is constrained to TCP benefits and will call a content to imitate the administration and communicate with the attacker.

Data social event should be possible with the assistance of two strategies: syslogd as well as a sniffer. As a matter of course, Honeyd logs all TCP and ICMP endeavors to syslogd daemon. This data is constrained to value-based viewpoint just and gives joined view into the real attack. What's more, Honeyd benefit copying contents can have a logging capacity. As Honeyd is open source item it is effortlessly adjustable to incorporate all the more logging capacity, hence enhancing its utility. Also, sniffer can be utilized to catch the network activity interfacing with the Honeypot. Interaction an excessive number of TCP mainstream services sent data sick clear content preferences of Telnet, FTP, HTTP, this caught data is exceptionally useful in further researching the attack. Caught data examination will create an extraordinary attack signature.

Honeyd has no worked in warning network, so a different arrangement must be utilized. Honyed being a low communication Divert presents constrained hazard factor.

3.3 Honeynets

Honeynets are high-Interaction Honeypots. No services are imitated, and no confined situations are made. Genuine frameworks are offered to the attacker behind some entrance control gadget. The framework arrangement can be heterogeneous i.e. the frameworks inside a Honeynet are genuine creation frameworks. Honeynets are extremely adaptable apparatus. Honeynets cheat attackers, distinguish attacks and catch the obscure.

Honeynets require a broad measure of time and assets to construct, actualize and keep up. This technology includes enormous incentive as research Honeypot. These are utilized chiefly to address following security concerns:

- Who are the attackers?
- What instruments they utilize?
- What strategies do they utilize?
- What motivates at that point

Honeynets can gather top to bottom data about the attackers, for example, their keystrokes when they bargain the framework, their visit sessions with their associates, the devices they used to test and adventure, powerless frameworks.

As research Honeypot, Honeynets additionally exceed expectations at pattern investigation and factual displaying. The data accumulated can be utilized to anticipate attacks, going about as an early cautioning framework.

Working of Honeynets

A Honeynet is comprised as a network of numerous frameworks. It is an independent domain with three basic components: data control, data catch and data accumulation. Data control is the controlling of the blackhat action. Once blackhat takes control of a Deflect inside the honeynet, action should be controlled so attacker cannot hurt any non honeynet frameworks. Data catching is catching of all the action that happens inside the honeynet. Data gathering is the total of the considerable number of data caught by different honeynets. Figure 4.6 demonstrates the essential engineering of a Honeynet. In the test lab under this work, different honeynet ages were conveyed and broke down. Data caught and examination is exhibited in this area.

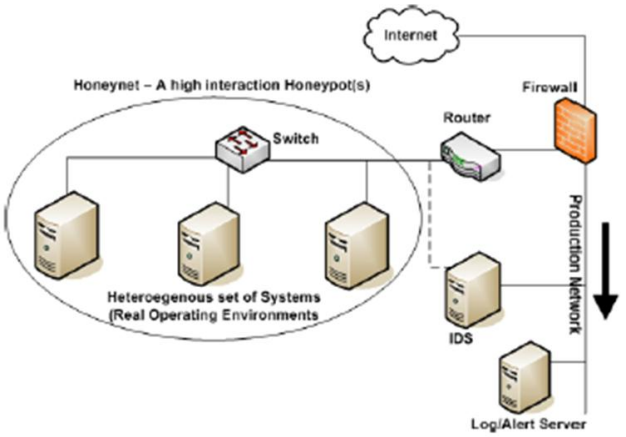

Figure 4.6: Basic Architecture of Honeynet

Firewall machine gives Data control highlights and IDS machine gives data catch include separated from the Log server which get data from the Honeynet utilizing an undercover channel. Undercover Direct setup in the exploration lab was finished utilizing the sebek technology. Where data is sent by Honeynet to the log server utilizing UDP port 1101.

In the test condition setup for this work, firewall is designed utilizing three network interface, one for the honeynet, one for the Web availability and other for generation arrange. IDS machine has two interfaces, one interface has been given an IP address while other is kept IP-less which is being utilized for sniffing purposes, to record the network action, and this gives a stealth interface for data catch. Three casualty machines, Linux 2.4.x. Linux 2.6.x and Windows 2000 were introduced with default designs. Honeynet data control at firewall level gives Interaction obstructing and Interaction restricting usefulness.

Data Catch in a Honeynet is classified into following four classifications:

- Network exchange recording
- Network movement recoding
- Host movement recording
- IDS alarms

Network exchanges happening in the honeynet incorporate inbound correspondence and Interaction endeavors from the Web, inside Interactions between the machine inside the nectar net and the outbound correspondence started by the nectar net. Outbound Interaction front the nectar net is an unequivocal pointer of the threatening movement. Network movement recording gives greatest level of points of interest on the gatecrasher exercises. Host movement recording incorporates the account of the attacker's keystrokes and other host process interchanges. Host logs are amazingly helpful for breaking down attack follows. At last, IDS cautions add structure to arrange activity examination and permit to make a move in light of what is happening in the honeynet.

Linux IP Tables was utilized in time test setup for network exchange recording. Following are the modules stacked on the firewall machine eth0 172.31.1.17. eth1 202.164.44.99, eth2 Microsoft loopback connector (utilized for remote administration purposes).

```
[root@ns1 /]# lsmod
Module                  Size  Used by    Not tainted
ipt_limit               1560   4  (autoclean)
ipt_state               1048  22  (autoclean)
ip_conntrack_irc        4112   0  (unused)
ip_conntrack_ftp        5296   0  (unused)
ipt_LOG                 4152  14
iptable_mangle          2776   0  (autoclean) (unused)
iptable_nat            21720   1  (autoclean)
ip_conntrack           26976   4  (autoclean) [ipt_state ip_conntrack_irc ip_co
ntrack_ftp iptable_nat]
iptable_filter          2412   1  (autoclean)
ip_tables              15096   8  [ipt_limit ipt_state ipt_LOG iptable_mangle ip
table_nat iptable_filter]
```

The accompanying data catch from the firewall ix separated from the syslog devil logs

Oct 18 18:31:24 nsl kernel: IP_conntrack version 2.1 1023 buckets, 8184 max) -292 bytes per contract

Oct 18 18:39:08 nsl kernel: INBOUND ICMP: IN=eth1 OUT=eth0

SRC=202. 164.55.101 DST=172.31.1.50 LEN=60 TOS=0x00 PREC=0x00 TTL127

ID=18522 PROTO=ICMP TYPE=8 CODE=0 ID=1024 SE=512

Oct 18 18:39:13 nsl kernel: INBOUND ICMP: IN=ethl OUT=eth0

SRC202. 164.55.101 DST=172.31.1.50 LEN=60 TOS=0x00 PREC=0x00 TTL=127

ID=18523 PROTO=ICMP TYPE=8 CODE=0 ID=1024 SEQ=768

Oct 18 18:46:14 nsl kernel: INBOUND TCP: IN=ethl OUT=eth0

SRC202. 164.55.101 DST =172.31.1.50 LEN=48 TOS=0x00 PREC=0x00

TTL=127 ID=18714 DF PROTO=TCP SPT=1801 DPT=80 WINDOW=65535 RES=0x00

SYN URGP=0

Oct 18 18:46:32 nsl kernel: INBOUND TCP: IN=ethl OUT=eth0

SRC202. 164.55.101 DST =172.31.1.50 LEN=48 TOS=0x00 PREC=0x00

TTL=127 ID=18722 DF PROTO=TCP SPT=1802 DPT=80 WINDOW=65535 RES=0x00

SYN URGP=0

Table 4.2 shows various IP table Log entries and their respective meaning.

Table 4.2: IPTables Log entries and their meaning

Entry	*Meaning*
Oct 15 18:31:24	Syslog Date-Time Stamp
nsl	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 addresses
SPT=1801 and DPT=80	Source and Destination port addresses

TCP log appeared above shows IP deliver 202.164.44.101 associating with the machine 172.31.1.40 at port number 80 i.e. http benefit. This is log bit of Interaction commencement stage as can be seen from the SYN bit of three different ways handshake is determined to. Figure 4.7 Demonstrates the tcpdump gathered at IDS machine, indicating phpBB attack. This adventures two subjective Get up and go code execution defects in the phpBB discussion framework. The issue is that tire 'feature parameter in the "viewtopic.php" content isn't Honeypotsed legitimately and will enable an attacker to infuse self-assertive code by means of preg _replace(). Figure 4.8 shows stream chart of the attack marks catch utilizing grunt.

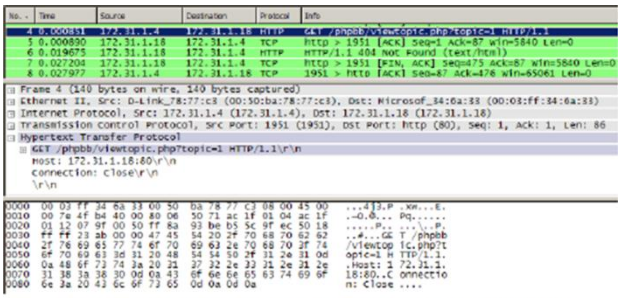

Figure 4.7: Tcpdump network traffic log analyzed using Wireshark

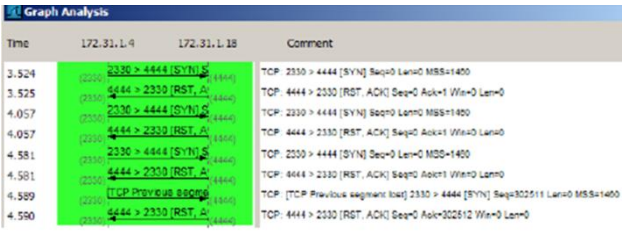

Figure 4.8: Flow graph statistics of an attack

One downside with the GenI honeynet is that it is anything but difficult to get identified which gives an insignificant ability to ponder the attacks. Impediments are piece of data to the confined number of permitted active Interactions from the honeynet and the utilization of layer 3 interchanges. GenII honeynets give more stealthy task. In age II honeynets data control and data catch are actualized on a solitary gadget, called Honeywall. This design additionally gives new keystroke logging running at both honeywall and honeynet. These advances bring down the likelihood of honeynets being distinguished by blackchats, bring down the danger of losing data, Honeypots encoded correspondence on the Honeypots and give a glass-box observing apparatus about the Divert's maternal state.

CONCLUSIONS

This section focuses on the investigation of opensource Honeypots and shows their utilization in the network security pecking order. Tradeoffs between levels of Interactions are accounted for. Finish points of interest opensource Honeypots including Back officer Well disposed, Honeyd and Honeynets are exhibited by setting up these at the work environment. It was discovered that by exploiting virtualization programming like Microsoft Virtual PC (as is done in the exploratory setup) physical necessities of setting up a honcynet can be extraordinarily decreased. These virtual honeynets permit to run this proactive security technology all the more proficiently. This part accomplishes second target of the theory work.

Ideas of self-regulation, secretive channel correspondence, data control and data catch while keeping up the inward condition of the honeynet is utilized broadly sick the proposed network, which is explained facilitate in the following part of this proposition work.

REFERENCES

1. Spitzner, L. (2002). Honeypots: Tracking Hackers. 1st ed. Boston, MA, USA: Addison Wesley.

2. Mokube, I. & Adams M. (2007). Honeypots: Concepts, Approaches, and Challenges. ACMSE 2007, March 23-24, 2007, Winston-Salem, North Carolina, USA, pp. 321-325.

3. Aaron Lanoy and Gordon W. Romney (2006). Senior Member, IEEE [2006] A Virtual Honey Net as a Teaching Resource.

4. F. A. Shuja. (2005, November). Virtual Honeynet: Deploying Honeywall using VMware, Pakistan Honeynet Project [Online], Available: http://www.honeynet.org.pk/honeywall/roo/

5. (2005, August). Know Your Enemy: Honeywall CDROM Roo 3rd Generation Technology, Honeynet Project & Research Alliance, [Online] Available: http://www.honeynet.org, Last Modified: 17 August, 2005.

6. G. Romney, et. al. (2004). "A Teaching Prototype for Educating IT Security Engineers in Emerging Environments," Presented at the IEEE ITHET 2004 Conference in Istanbul, Turkey, June 2, Published in IEEE Xplore.

7. Cliaord Stoll (1988). Stalking the Wily Hacker. Communications of the ACM. Pp. 484- 497.

8. Ram Kumar Singh & Prof. T. Ramanujam (2009). Intrusion Detection System Using Advanced Honeypots.

9. Martin Roesch (2005). Snort- Lightweight Intrusion Detection for Networks, Proceedings of LISA'99: 13th System Administration Conference, Seattle, Washington USA, 2005

10. The Honeynet Project (2005). Know Your Enemy: Honeynets (May 2005) http://www.honeynet .org/papers/honeynet/.

11. Honeynet Research Alliance (2003). Project Honeynet Website. Retrieved May 16th 2003 from the World Wide Web: http://project.honey.org

12. Brian Scottberg et. al. (2002). Internet Honeypot: Protection or Entrapment, 2002.

13. The Honeynet Project (2001). Know Your Enemy: Honeynets, April 2001.

14. The Honeypot Project (2002). Know Your Enemy: Revealing the Security tools, tactic, and motives of Blackhats community, 2002.

15. B. Baker J. Pincus (2004). Beyond stack smashing: recent advances in exploiting buffer overruns,Http:/ /research.Microsoft.Corn/users/jpincus/ beyoridstacksmashing.pdf, 2004.

16. T. Anderson, T. Roscoe, D. Wetherall (2004). "Preventing Internet Denial-of-Service with Capabilities," In ACM SIGCOMM Computer Communication Review, Volume 34, Issue 1.

17. A. D. Keromytis, V. Misra, and D. Rubenstein (2002). "SOS: Secure Overlay Services," in the Proceedings of. ACM SIGCOMM, 2002.

18. J. Li, J. Mirkovic, M. Wang, and P. Reither (2002). "Save: Source address validity enforcement protocol," Proceedings of IEEE INFOCOM, pp. 1557-1566.

19. M. Handley (2005). "Internet Architecture WG: DoS-resistant Internet subgroup report," 2005. http://www.communications.net/object/download/1543/doc/mjh-dos-summary.pdf.

20. B. Baker J. Pincus (2004). Mitigations for Low-Level Coding Viilneiabthties: Incompa'rabzlity and Limitations, http://research.miciosoft.com/ users/ jpincus/mitigations.pdf, 2004.

Corresponding Author

Saurabh Kawatra*

HOD (Computer Science), Mata Jeeti Ji Girls College, Suratgarh, Rajasthan

drsaurabhkwatra@rediffmail.com