DESIGN AND EFFICIENT DEPLOYMENT OF HONEYPOT AND DYNAMIC RULE BASED LIVE NETWORK INTRUSION COLLABORATIVE SYSTEM

Renuka Prasad.B[1], Dr Annamma Abraham[2], and Abhas Abhinav[3], Sunil.V.Gurlahosur[4], Srinivasa Y[5]

[1]Research Scholar, Dr M.G.R University, Working as Lecturer in R.V.College of Engineering, Bengaluru, Karnataka
renukaprasadb@gmail.com
[2]Head , Department of Mathematics,BMSIT, Bengaluru, Karnataka
annamma65@gmail.com
[3]Head ,Research and Development, DeepRootLinux Pvt Ltd, Bengaluru,Karnataka
abhas@deeproot.co.in
[4]Student, BE ,Computer Science and Engineering, SIT,Tumkur, Karnataka
sunilcs6@gmail.com
[5]Student, BE ,Computer Science and Engineering, SIT,Tumkur, Karnataka
srinivasaven@gmail.com

ABSTRACT

The continuously emerging, operationally and managerially independent, geographically distributed computer networks deployable in an evolutionarily manner have created greater challenges in securing them. Several research works and experiments have convinced the security expert that Network Intrusion Detection Systems (NIDS) or Network Intrusion Prevention Systems (NIPS) alone are not capable of securing the Computer Networks from internal and external threats completely. In this paper we present the design of Intrusion Collaborative System which is a combination of NIDS,NIPS, Honeypots, software tools like nmap, iptables etc. Our Design is tested against existing attacks based on Snort Rules and several customized DDOS , remote and guest attacks. Dynamic rules are generated during every unusual behavior that helps Intrusion Collaborative System to continuously learn about new attacks. Also a formal approach to deploy Live Intrusion Collaboration Systems based on System of Systems Concept is Proposed.

KEYWORDS

Network Intrusion Detection, Network Intrusion Prevention, IPTABLES, Honeypot and NICS.

1. INTRODUCTION

A Comparative Study of Network Intrusion in Detection Systems in[1], In 2008 Moses Garuba, Chunmei Liu, and Duane Frates have conducted an extensive study on the different Intrusion techniques [1] and they also demonstrated that NIDS alone cannot handle both internal and external threats to computers. They also proposed that Heuristic Based solutions are better than signature based solutions. Self Adaptivity and Dynamic analysis are the key features that have to be there in any NIDS as the responsiveness for any NIDS is determined by these properties. In [2] the importance of dynamic behavior of the NIDS is demonstrated by Zang Qing Hua , Fu Yu Zhen, Xu Bu-gong . Luis Carlos Caruso and others have submitted their proof of concept on huge computing power requirement for signature based NIDS called SPP-NIDS [3]. The limitations as mentioned In [4] and [5] after a certain communication link speed NIDS will fail to perform as the load increases and softwares like SNORT [4] require a huge computing capability to handle communication line greater then 100Mbps. Miyuki Hanaoka and others have discussed the importance of collaboration between the security mechanisms and

specifically the collaboration between many NIDS[5]. They also demonstrated that redundant rules could be eliminated between the NIDS with a collaborative model. NIDS alone is not sufficient to handle entire range of threats and attacks on the computer networks. Network Intrusion Preventive mechanisms will also help significantly in reducing the effect of an attack over a computer network. Network Intrusion Preventive mechanisms like traditional firewall along with strong authenticating procedures in collaboration with NIDS will make a computer network more secured [6]. Softwares like IPTABLES [7] can be used in setting up a firewall on an operating system with Linux as a kernel, version is higher than 2.4. With iptables and NIDS a variety of security related mechanisms are implemented in psad [8]. Firewall play a vital role in NIPS, Despite taking all these precautions attacks still happen and the computer security system still fails to secure the computer networks in case of new type of attacks. Hence a mechanism where it would be possible for the attackers to get trapped unknowingly so that the systems can secure the computer networks from getting infected is essential. HoneyPots [9] can be used to secure the computer network along with NIDS and NIPS. Honeyd is a small daemon that creates virtual hosts on a network. The hosts can be configured to run arbitrary services, and their personality can be adapted so that they appear to be running certain operating systems. Honeyd enables a single host to claim multiple addresses. Honeyd improves network security by providing mechanisms for threat detection and assessment. It also deters adversaries by hiding real systems in the middle of virtual systems. The Collaboration of NIDS system like SNORT , NIPS mechanism like IPTABLES and Honeypot like honeyd will make the Intrusion Collaborative Security System more pwerful and robust. Knowing the enemies for a computer network with more practical details is demonstrated in the whitepapers of honeynet[10] project.. Every day computer networks are growing in a very large scale and there are more and more people trying to attack the networks. Hence securing the computer networks demands a distributed solution. Security system for each network will finally create a bigger system of security system which will be deployed in a distributed manner. [11], [12], [13], [14], [15], [16], and [17] gives the definition of System of Systems and explain the emerging characteristics of System of Systems where Internet is an example of such system. But here we are integrating NIDS, NIPS (IPTABLES) with Honeypot and deploy these mechanism considering Security System as a System of Systems. Though the security system is setup in each network it is very important to deploy the mechanisms at proper places. It is important the reporting mechanism is highly reliable as the centralized server will be taking the report from the sub systems. Section 2 describes the procedure of setting up and deployment of Network Intrusion Collaboration System. Section3 describes a formal approach of Systems-of-Systems towards effective deployment of Intrusion Collaborative system in a distributed manner. Several experiments are mentioned in the papers [23], [24], [25] and [26] that discusses major challenges and issues associated with the distributed deployment of Intrusion Detection Systems in a large-scale and distributed networks. Results and related discussion are done in Section4.

2. SETUP AND DEPLOYMENT OF INTRUSION COLLABORATIVE SYSTEM

Network Intrusion Collaboration System is a combination of Intrusion Detection and Intrusion Prevention mechanisms. Our experimental setup includes SNORT as NIDS , IPTABLES as Preventive mechanism and honeyd as the honeypot and a customized statistical classifier written using shell programming language to extract information from the network data. We experimented with a network of 1500 to 1600 computers with 16 class C Subnetworks. Each Subnetwork was configured with a separate NIDS (SNORT) and a firewall(IPTABLES – F1,F2,F3 and F4 in Diagram1) to detect and prevent any intrusions. Honeypot (H1,H2,H3 and H4 in Diagram1) was introduced in each of the network and information was extracted from its log continuously to detect any abnormal behavior in the network. Few Subnetworks installed Honeypots on the virtual machines to hide from honeypot detectors. Snort rules and as well as customized rules were written and fed to the Honeypot. Firewall keeps updating the

information about the anomalous or unusual activities. Firewall was implemented using IPTABLES provided by any GNU/Linux Distribution. Debian Squeeze distribution was used to setup the firewall in each Subnetwork. On some machines fedora11 was used. Each Subnetwork had computers ranging from 75 to 100 with different operating systems running on them. Each Subnetwork had both wired and wireless switches. Only one DHCP server was used for the entire network. Each Subnetwork had the freedom to setup their own proxies (PR1,PR2,PR3 and PR4) and filter traffic according to their need.

As described in the Diagram1, Snort will detect all the known attacks based on its signatures. Firewall will prevent unauthorized activities. In the event of malicious or anomalous behavior statistical information can be extracted [18] by the scripts and immediately reported to the classifier (C1,C2,C3 and C4 in Diagram1) which will continuously keep creating a knowledge base (D1,D2,D3 and D4) of the behavior of the entire subnetwork which will be finally sent to a centralized server (D-main in Diagram1) that keeps track of the activity of the entire network and helps the administrator in taking decisions. Administrator will decide on blacklisting IP [19] in case of any attacks , threats or anomalous behavior based on the information obtained from attack classifier. NIDS was also deployed at appropriate locations to check the internal attacks in every subnetwork. The information about the activity of the subnetwork within the network was always collected and sent to the central server to take corrective measures to avoid threats from internal resources. An aggregate DDOS attack pattern generator [20] was used to test the capability of the Collaborative System in tracing the internal intrusions and attacks. Also other attacks were generated using software tools like metasploit [21] and nmap [22] to test Network Intrusion Collaboration Systems-of-Systems.

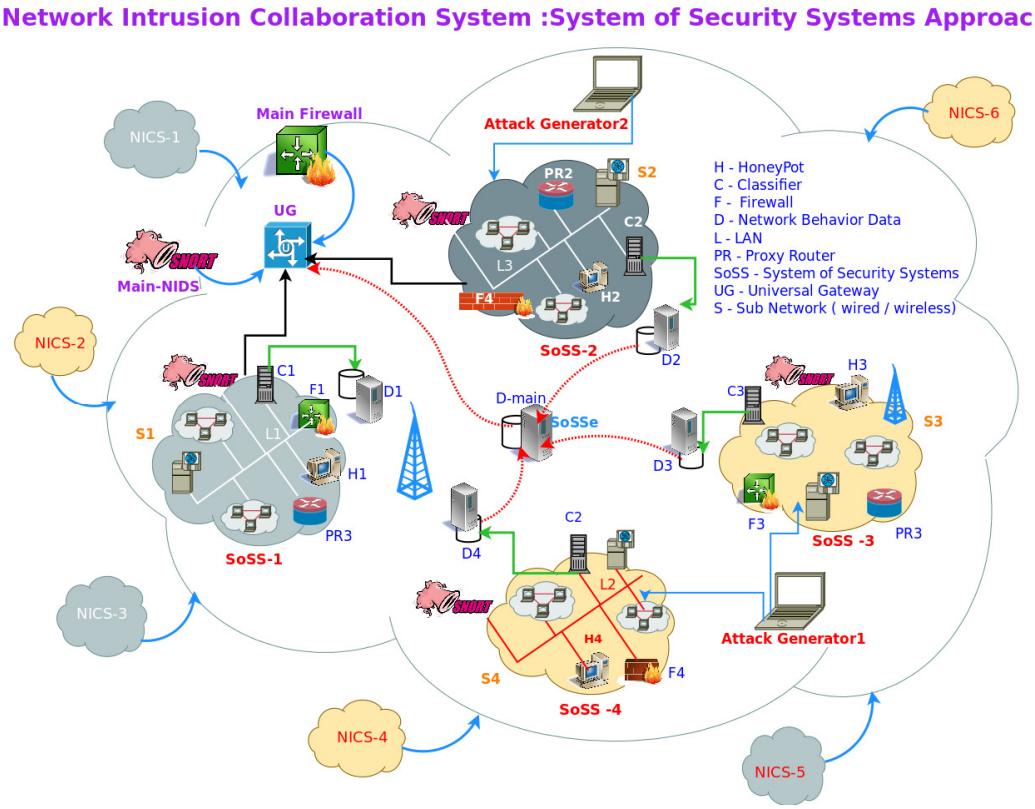

Diagram 1. Setup and Deployment of Network Intrusion Collaboration System-of-Systems

As discussed in the papers [25], [26], [27] and [28] it is evident that when the size of the computer network scales up the deployment of any security mechanisms will become more and more complex. Lot of challenges will have to be faced and many issues have to be addressed. The Foremost task is the reliability of the information, hiding the honeypot , collecting the right information from the right resource in right time and the counter measures that have to be taken during any threat or intrusions and similar such activities. Also anomalous behavior have to be tracked and continuously keep learning about the network behavior and build knowledge base which has to be further shared with the other nodes. [11], [12], [13], [14], [15] ,[16] and [17] discusses the complexity involved in System of Systems . We consider and propose that Intrusion Collaboration Systems should also be treated as a System of NIDS and NIPS and Honeypot and many other complex systems including the Subnetworks , These System of Security Systems(SoSS) in each subnetwork must collaboratively work together to fight against intrusions. SoSS will eventually become Operationally independent of one another, Managerially independent of each System, Deployable in an Evolutionary manner, Emergent, Distributed Geographically, and Heterogeneous while Networking with Systems. With such a complexity involved to automate the entire process a formal approach would be ideal to deploy live network intrusion collaboration system.

3.SYSTEMS-OF-SYSTEMS APPROACH TOWARDS DEPLOYING INTRUSION COLLABORATION SYSTEM : A FORMAL APPROACH

Intrusion Collaboration System can be defined as an Infinite group[27] of systems. System of Security System (SoSS) with binary operation Bin satisfying axioms G1, G2, G3 and G4.

G1: Closure Axiom

$$\forall\, a1,a2 \in groupG, a1Bina2 \in G \quad \text{(i)}$$

$$\forall\, SoS1,SoS2 \in SoSS, SoS1BinSoS2 \in SoSS \quad \text{(ii)}$$

G2: Associative Axiom

G2: Associative axiom or Associative Law

$$\forall a,b,c \in groupGa*(b*c)=(a*b)*c \quad \text{(iii)}$$

For a System of System to be called as group, when the elements try to establish relations with one another under a particular binary operation it should adhere to the associative axiom which states that

$$\forall SoS1,SoS2,SoS3 \in SoSS, SoS1Bin(SoS2BinSoS3)=(SoS1BinSoS2)BinSoS3 \quad \text{(iv)}$$

G3: Identity Axiom

G contains an element 'e' such that

$$\forall a \in Ga*e=e*a=a; \text{`` } e \text{'' } isanidentityelement \quad \text{(v)}$$

System of Systems contains an element called as identity element 'Se' such that

$$\forall\, SoSa \in SoSS, SoSaBinSoSe=SoSeBinSoSa=SoSa \quad \text{(vi)}$$

G4: Inverse axiom

$$\forall a \in G \exists a^{-1} \in G \mid a*a^{-1}=a^{-1}*a=e, wherea^{-1} iscalledasinverseofa. \quad \text{(vii)}$$

With respect to SoS

$$\forall SoSa \in G \exists SoSa^{-1} \in G \mid SoSa*SoSa^{-1}=SoSa^{-1}*a=e \quad \text{(viii)}$$

$$,whereSa^{-1} iscalledasinverseofSa\ .$$

So any non empty set that satisfies all the above four axioms will be called as Group.
There will be scenarios where the sequence of occurrences of the elements is not important i.e S1, S2, S3 could be operated with the binary operation Bin in any order but still the solution is obtained. That means even though the binary operation is applied with the different combinations of the S1, S2, S3 would not matter and what matters is the only value that would be obtained after the application of the binary operation over S1,S2 and S3. this is formally derived in the axiom G5 called commutative law as stated below .
G5: commutative law :

$$\forall a,b \in G, a * b = b * a \quad \text{(ix)}$$

is also satisfied, then G is called as an Abelian Group with respect to * or commutative group with respect to *.
With respect to SoS the commutative axiom states that

$$\forall\ SoS1, SoS2 \in SoSS, SoS1 BinSoS2 = SoS2 BinSoS1 \quad \text{(x)}$$

is satisfied then such a System of Security System should be defined as an Infinite Abelian Group. This definition will helpful when defining such a type of systems where there is flexibility in the order of joining of the elements into a group but the order of application of binary operations should not be changed .

5. WORKING OF NETWORK INTRUSION COLLABORATIVE SYSTEM-OF-SYSTEMS

Setting up of SNORT for main network and in each subnetwork was done according to the instructions given in [28], Firewall (IPTABLES) for main network and in each subnetwork was done as given in [29], To setup honeypot (honeyd), procedure given in [30] was followed. To view the alerts and other information of NIDS individually BASE was used. Log analyzer for honeypot was used to view the statistics individually and to track the abnormal behavior of the network. Attack generators were used to generate DDOS and IPSpoof attacks to test the System-of-Security Systems.

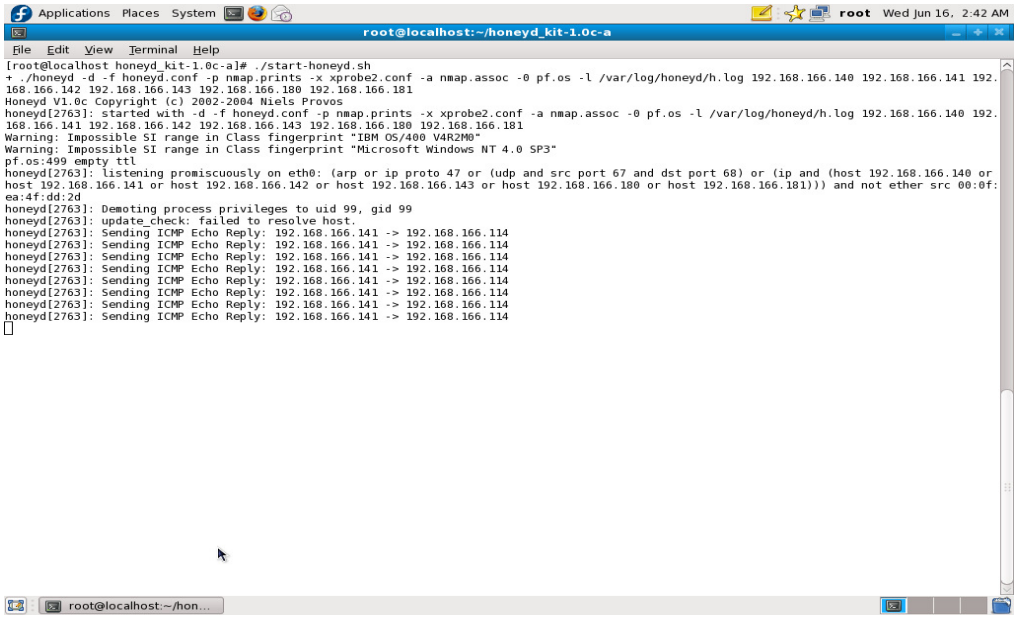

Diagram 2 :Honeyd when started to send reply for ICMP requests

When Spoofing Attacks were generated honed will start logging the abnormalities as we have customized to log the abnormalities in the network traffic . In Diagram 2 few logs are visible.

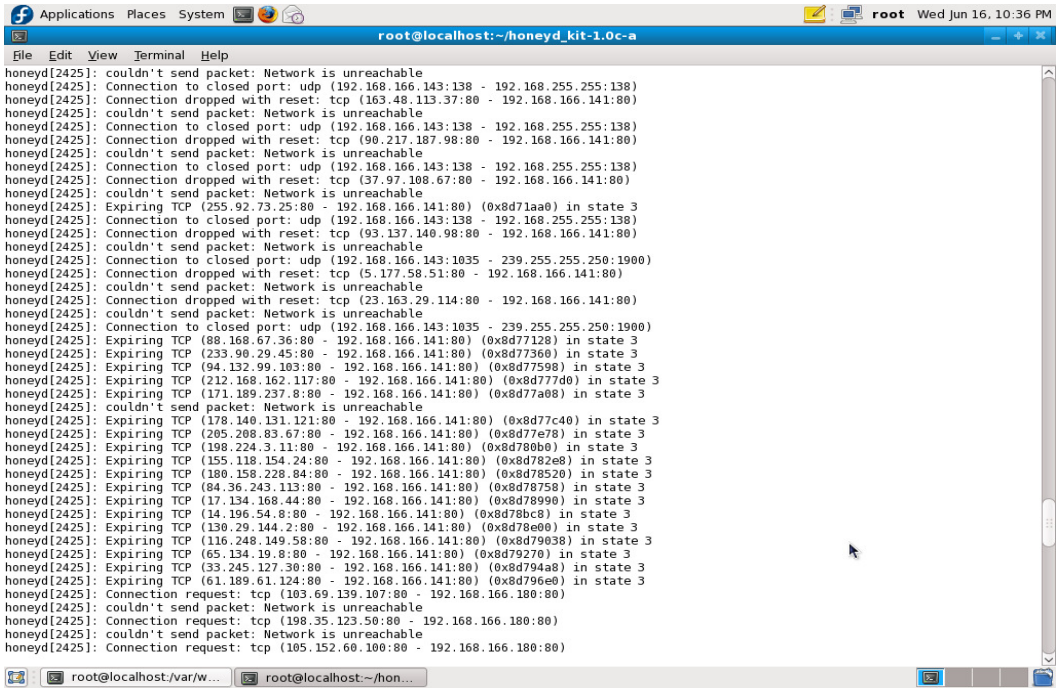

Diagram3: Honeyd when started to send reply for ICMP requests under IP spoofing

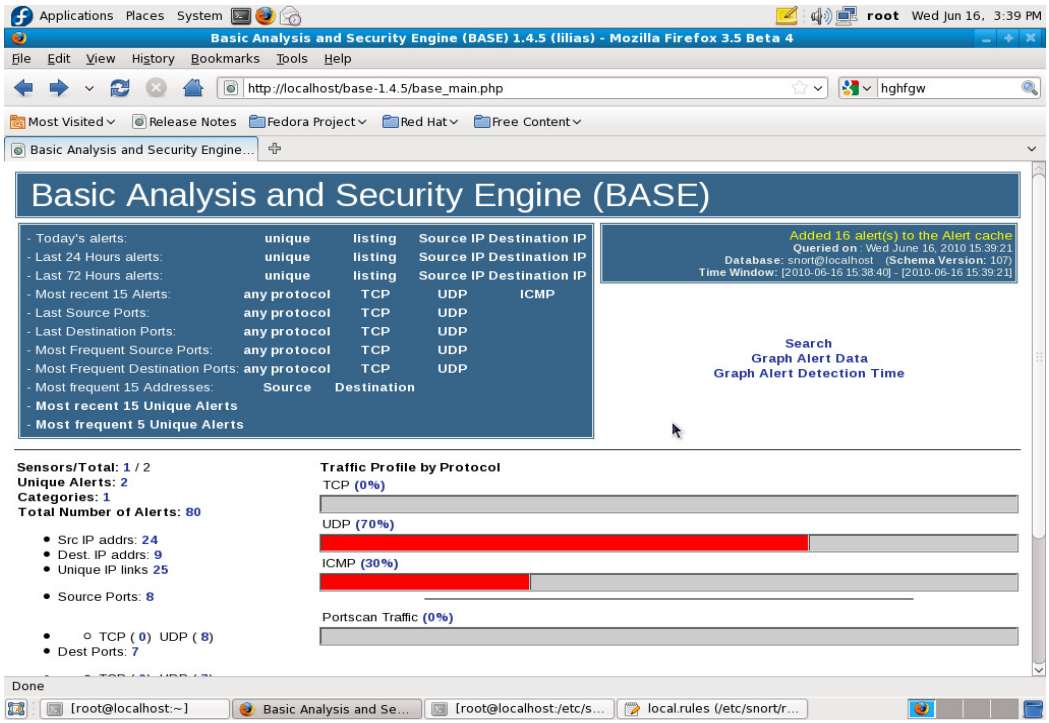

Diagram 4: View of Snort alerts using BASEnort alerts using BASE

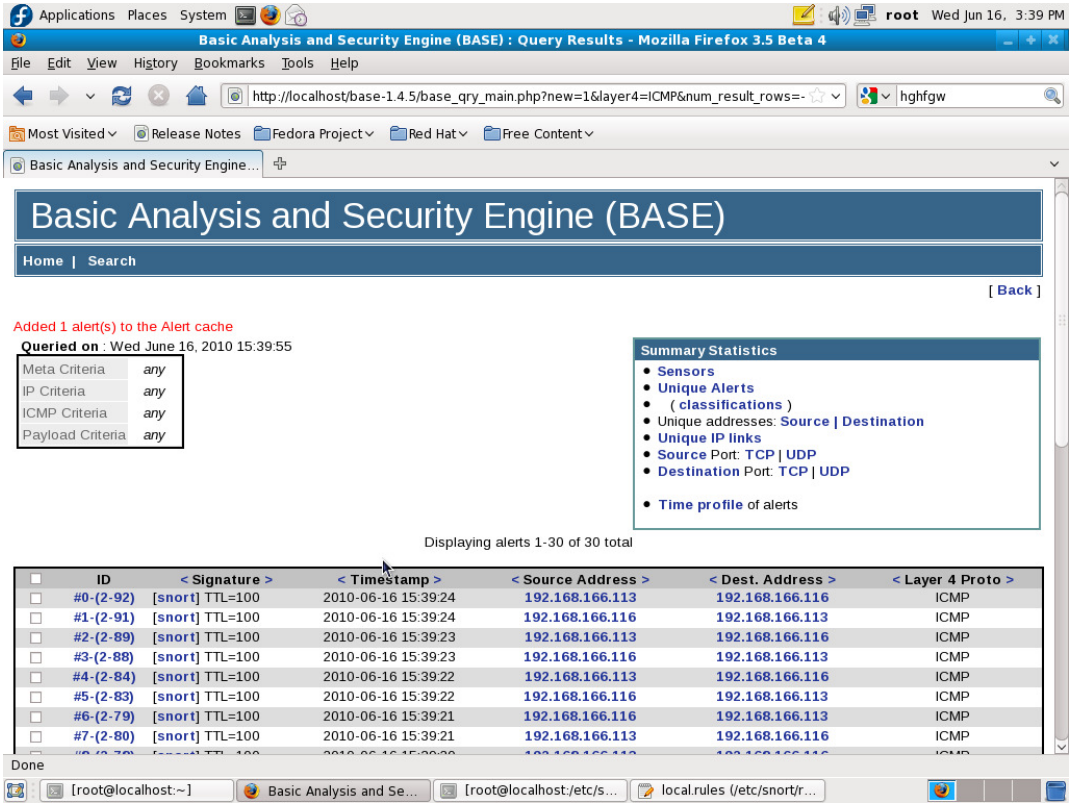

Diagram5 : An in-depth view of some of the alerts

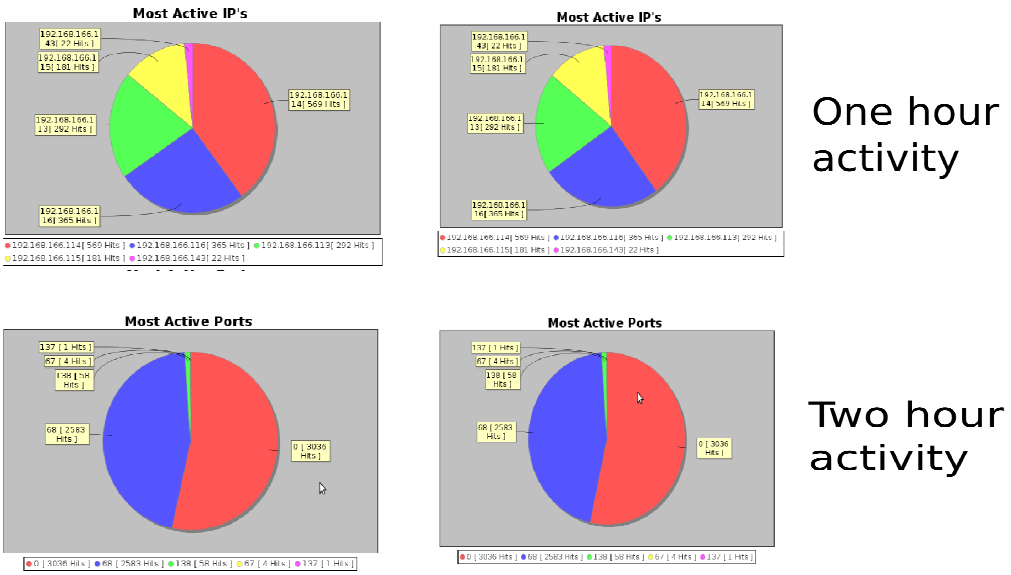

Diagram 6: Honeyd log analyzer output without Attacks

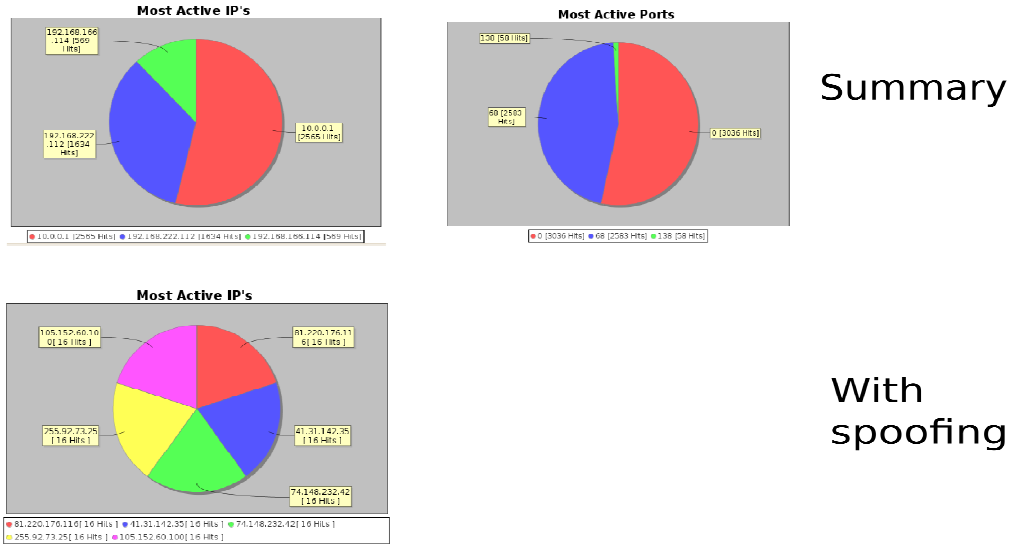

Diagram 7: Honeyd log analyzer output with and without IPSpoof Attack

As given in Diagram1, Collecting information from each subnetwork knowledge base (D1,D2,D3,D4 etc) to the main network knowledge base (D-main) can be done automatically using shell scrips like given below.

#!/bin/sh – This is a shell script.
for computer in the group(Subnetwork) – On each of these "n" machines in turn...
do
rsh $computer scp logfile user@eachcomputer – Copies the log file from each machine
done

The above script will work provided the following this are taken care : Public key files have to be placed in the remote computer user accounts and ssh-clients(ssh, scp) should be allowed to access the remote accounts. On each of the machines in the local network, that is each System of System ssh-agent program has to be invoked which will be running as a background process and then during the login session keys have to chosen, keys have to be loaded into the agent using the ssh-add program. Finally once there is a secured connection between the machines which are used to send the knowledge base from each network to the main system shell scripts as mentioned can be executed to automatically collect information. The scripts could also be loaded either to crontab or schedule manually whenever the information has to be collected.

Attacks were generated using nmap, metasploit and customized scripts for DDOS attacks

#nmap -v -O --osscan-guess 192.168.1.1 | egrep 'MAC Address:|Devicetype:|Running:|OS details:|Uptime guess:|Network Distance' >> systemdetails1.txt

#nmap -v -O --osscan-guess 192.168.1.1 | egrep 'MAC Address:|Device type:|Running:|OS details:|Uptime guess:|Network Distance' >>systemdetails1.txt

#nmap -v -O --osscan-guess 192.168.1.1 | egrep 'MAC Address:|Device type:|Running:|OS details:|Uptime guess:|Network Distance' >>systemdetails1.txt

#nmap -v -O --osscan-guess 192.168.1.1 | egrep 'MAC Address:|Device type:|Running:|OS details:|Uptime guess:|Network Distance' >>systemdetails1.txt

Metasploit was used to write few exploits. Custom-built attacks were generated to test the capabilities of Network Intrusion Collaborative System like

Pseudo-code for IP-Spoofing is as follows

1. target_ip <- argv[1]
2. Perform steps 3 to 9 till program is interrupted / terminated
3. ip_rand <- rand()
4. a <- ip_rand & 0xFF
5. b <- (ip_rand >> 8) & 0xFF
6. c <- (ip_rand >> 16) & 0xFF
7. d <- (ip_rand >> 24) & 0xFF
8. create command string with appropriate data Payload , data.txt(contains randominfo) , TCP header info , randomly spoofed source_ip (a.b.c.d), dest_ip (target_ip) , IP header info , source_port (random_int) , dest_port (80). 9. system(command)

Pseudo code for live network intrusion collaboration system-of-systems

1. Setup Subnetworks depending upon the requirement and assign Class C addresses ,
2. Install NIDS and Firewall at the main gateway,
3. Setup firewall in each subnetwork and proxy server if required.
4.Setup NIDS (SNORT) in each subnetwork to detect the inside attackers.
5.Setup the Honeypot, embed the snort rules inside honeypot ,
6.Add the log extractor to the crontab of the main gateway.
7.Setup SSH-Server and SSH-Clients and setup the identity element (SOSe in Diagram1)
8.Regulary (depending on the requirement) update the knowledge base about the network behavior (either manually or automatically)
9.Take decisions dynamically depending on the information collected from the entire subnetworks

5. RESULTS AND DISCUSSION

With the introduction of the NIDS, Firewall, Classifier and Honeypot in each SOSS , The security level of each subnetwork is increased by 50 %. Also the efforts in tracing the abnormal activities is minimized exponentially since most of the traffic filtering can be done at the firewall and NIDS. In Subnetwork firewall and NIDS takes care of detecting major known anomalous behavior using signatures and concentration will be on the new type of attacks which will be easily traced with the honeypot and reported to the classifier and again reported to the main server to record the abnormal activity to take corrective measures by the administrator. This mechanism also reduced the number of alarms usually raised by the NIDS upto 70%. Honeypots and NIDS detect the abnormal behavior during an internal and external DDOS attack within 5 seconds and were able to take corrective measures within 7 seconds whereas a network without Honeypot was clogged within 12 seconds and the switches were completely non functional and entire system was supposed to be shut down. With the introduction of the firewall and a proxy router at each subnetwork , traffic filtering task was simplified. Universal Gateway had the major responsibility of deciding the genuineness of an activity. Always it is possible to sneak into the network but the intruder or attacker will always look for the compromised system and Honeypots will be able to easily trap them and report their interactions to the classifier to dynamically either blacklist those machines or prevent them from doing further damage, System of Systems approach will help in automating the process of information collection from the classifiers, several scripts were written with rssh commands to

collect the logs from each of the Honeypots and also the classifiers were automated which had helped in implementing the live Network intrusion Collaboration Systems-of-Systems.

5.1 Advantages of NICS approach

Parameter	Only with NIDS [1]	NIDS+ Firewall [2]	NIDS+ Firewall+ Honeypot [3]	NIDS+ Firewall+ Honeypot+ Classifier [4]	NICS Approach [5]
Security Level	20 %, Only Detection	60-70% Detects + Prevents	70 – 80% [2] +Trap Attackers	80-90 %,[3] +Dynamic Response	90 – 95 % [4] + Distributed
Resources Used (CPU + memory..)	2.00%	5.00%	15 – 20 %	20 – 25 %	> 25 %
External Intrusion /Attacks / Threats	Detected	[1]+ Prevented	[2] + Trapped	[3] + Reported & Responded	[4] + Live
Internal Intrusion /Attacks / Threats	Cant Detect	Can,@ each SoS	Can be trapped	Trapped + Reponded	[4] + Live Protection.
False Alarms	More	25 -30% < [1]	30 - 40% < [1]	40 - 50 % < [1]	60 – 70 % < [1]
GUI for results maintenance	Required [BASE]	[1] + custom	Custom Scripts	Custom-built Scripts	Custom-built Scripts
Statistical Reports generated Max Link Speed Handled	SNORT provides Upto 10 GBps	[1] + Iptables Upto 20 GBps	[2] + log Analyser Upto 25 GBps	[3]+Statistical information Tested upto 50 GBps	[4] + Live responses Tested upto 50 GBps
Effect of DDOS attacks	N/W fails in 20 sec	N/w Fails within 1 minute	N/W failures are less	N /W failures are < [3]	Network failures are Negligible
Effects of Ip Spoofing	N/W clogs in 5 secs	N/W clogs in 2 mins	N/w wont clog, logs generated	N/w wont be clogged ,logs generated	No n/w clogs, very specific alerts raised
Intrusion Detection Time	3-5 sec for known attacks	10 %-20% > [1]	26% >[1] can also detect new attacks	40 % > [1], can detect new attacks	50% prevented and Live Intrusion Detection

Table 1 : Advantages of Network Intrusion Collaboration System

Care should be taken while deciding the number of systems to be at compromised state. The major problem in creating a honeypot is that they take more resources and also measures have to be taken to hide the honeypot. Otherwise once the honeypots are detected the attacker will come to know about it and will not try to do further interactions. Installing honeypots on virtual machines will make the intruder tough to detect the honeypots. Services like ssh should be run in nonstandard ports so that the intruder will be blocked from trying standard ports for attacks . In each of the machine and as well at the proxy routers and firewalls of the the subnetworks care has to be taken about the traffic that each machine is allowed to generate. This will prevent around 30 % to 50% of the attacks that could be possible from an insider. Guest Attacks , like

when a new person enters the network with a laptop or any other device at a compromised state , then there are always chances of attacks getting multiplied. During this situation appropriate permissions and network policies have to be set for the guests otherwise it will be a very hard task to recover from an attack once they start multiplying.

5.2 Individual queries for detailed analysis in each subnetwork during portscan

mysql>select s.sig_name, count(*) as count from event e, signature s where e.signature=s.sig_id group by e.signature order by count desc;

sig_name	count
(portscan) Open Port	188
COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	144
(snort decoder) Bad Traffic Same Src/Dst IP	115
BAD-TRAFFIC same SRC/DST	115
(portscan) TCP Portscan	33
(portscan) TCP Portsweep	5
(spo_bo) Back Orifice Snort buffer attack	4
(spo_bo) Back Orifice Traffic detected	4
(http_inspect) BARE BYTE UNICODE ENCODING	1
(spp_ssh) Protocol mismatch	1
COMMUNITY SIP DNS No such name threshold - Abnormally high count of No such name responses	1

6. CONCLUSIONS

A Honeypot based Network Intrusion Collaboration System which is capable of generating dynamic rules during any anomalous behavior in the network or a possible intrusion is presented. The NICS designed is a collection of several existing Free and Open Source Softwares customized for the specific need that helps in implementing both preventive and detective mechanisms of network security.

The proposed NICS works in a distributed and collaborative manner. System of System Engineering concepts are adopted to deploy the NICS. A formal approach to deploy the NICS also presented. NICS is tested against many different types of intrusions and attacks. Several attacks were generated using the softwares like metasploit, nmap and customized scripts. A log analyzer will be continuously running in each subnetwork to collect variety of logs and an information extraction module will generate statistics to find the abnormalities in the behavior of the network.

Honeypot based NICS was capable of identifying the customized intrusion and other abnormalities in the traffic over network which were generated during attacks faster than conventional methods.

Looking at the results obtained, the number of false alarms generated, it if noticed that the computer network could be secured upto 95% from both internal and external attacks or threats or intrusions. In future we will be focusing on designing solution which is more robust. Also we will be focusing on increasing the intrusion detection rate, reducing false alarms and include the live attackers suspecting module.

7. ACKNOWLEDGEMENTS

I thank our institution management in providing the infrastructure and support in carrying out the experiments. I thank the principal of R.V.C.E for his support and also the Director of Master of Computer Application Department for their motivation and support. My special thanks to all those who supporting in implementing the ideas.

REFERECES

[1] Moses Garuba, Chunmei Liu, and Duane Fraites (2008) "Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems " Fifth International Conference on Information Technology: New Generations , Page 593-598, DOI 10.1109/ITNG.2008.231

[2] Zang Qing Hua , Fu Yu Zhen, Xu Bu-gong (2008)," A New Model of Self Adaptive Network Intrusion Detection System" , 978-1-4244-1823-7/08, Page 436-440

[3] Luis Carlos Caruso, Guilherme Guindani, Hugo Schmitt,Ney Calazans, Fernado Moraes, 2007 SPP-NIDS-A Sea of Processor Platform for Network Intrusion Detection, 18th IEE International Workshop on Rapid System Prototyping,

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

[5] Miyuki Hanaoka, Kenji Kono, and Toshio Hirotsu, 2009, "Performance Improvement by means of Collaboration between Network Intrusion Detection Systems",2009 Seventh Annual Communications Networks and Services Research Conference, DOI 10.1109/CNSR.2009.48, page 262-269

[6] Simon P. Chung and Aloysius K., Mok phchung, mok , 2005, Collaborative Intrusion Prevention, N00014-03-1-0705

[7] http://www.netfilter.org/projects/iptables/index.html

[8] Michael Rash (2007) "Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort" No Starch Press, II edition

[9] http://www.honeyd.org/

[10] http://www.honeynet.org/papers/

[11] Jamshidi, M., Large-Scale Systems - Modeling and Control, North-Holland Publishing Company, New York, NY, 1983 (Also 2nd Edition, prentice Hall, 1997).

[12] Crossley, W.A., "System of Systems: An Introduction of Purdue University Schools of Engineering's Signature Area.

[13] Sage, A.P. and C.D. Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," Information, Knowledge, Systems Management, Vol. 2, No. 4 (2001), pp. 325-345.

[14] Keating, C. B., Fernandez, A., Jacobs, D., and P. Kauffmann, "A Methodology for Analysis of Complex Socio-technical Processes," Business Process Management Journal, Vol. 7, No. 1 (2001), pp. 33-49.

[15] Kotov, V., "Systems of Systems as Communicating Structures," Hewlett Packard Computer Systems Laboratory Paper HPL-97-124, (1997), pp. 1-15.

[16] Carlock, P.G., and R.E. Fenton, "System of Systems (SoS) Enterprise Systems for Information-Intensive Organizations," Systems Engineering, Vol. 4, No. 4 (2001), pp. 242-261.

[17] Manthorpe, W.H., "The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for APL," John Hopkins APL Technical Digest, Vol. 17, No. 3 (1996), pp. 305-310.

[18] Renuka Prasad.B, Dr.Annamma Abraham, Chandan. C, Prabhanjan.A, AjayBilotia, Information Extraction for Offline Traffic Anomaly Detection in NIDS, IJCSNS, 2008, Vol8, No 9, Pages 309-315

[19] Renuka Prasad.B, Dr.Annamma Abraham "Hybrid Framework for Behavioral Prediction of Network Attack Using Honeypot and Dynamic Rule Creation with Different Context for Dynamic Blacklisting", 2010, ISBN 978-1-4244-5726-7 Page 471-476.

[20] Renuka Prasad.B, Dr Annamma Abraham, Suhas.V, Kundan Kumar , "DoS Attack Pattern Generator For Training The Neural Network Based Classifier To Dynamically Blacklist IP in HoneyPot Based NIDS/NIPS" , 2010, International Conference on Contours of Computing , Springerlink, ISBN -978-84-8489-988-2

[21] http://www.metasploit.com

[22] http://www.nmap.org

[23] Leila Rikhtechi Afshin and Rezakhani Roozbahani, Creating a Standard Platform for All Intrusion Detection/Prevention Systems , 2010 Second International Conference on Computer Modeling and Simulation, Pages 41-44

[24] Li Tian, "Design and Implementation of a Distributed Intelligent Network Intrusion Detection System", 2010 International Conference on Electrical and Control Engineering, Pages 683-686

[25] You Yang and Jua Mi , "Design and Implementation of Distributed Intrusion Detection System based on Honeypot" , 2010, 2nd International Conference on Computer Engineering and Technology, Volume 6 ,Page 260-266

[26] Haifeng Wang and Qingkui Chen "Design of Cooperative Deployment in Distributed Honeynet System", 2010, 14th International Conference on Computer Supported Cooperative Work in Design, Pages 711-716

[27] Joseph J Rotman (2008) ,An Introduction to the Theory of Groups , Springer Publication, IV edition

[28] http://www.aboutdebian.com/snort.htm

[29] http://www.aboutdebian.com/firewall.htm

[30] http://www.symantec.com/connect/articles/open-source-honeypots-part-two-deploying-honeyd-wild

Authors

Renuka Prasad B is currently working as Lecturer in MCA department of R.V.College of Engineering, Bangalore, Karnataka,India pursuing his Phd in Dr MGR Educational and Research Institute, Tamil Nadu, India. His area of interest are Network Security, FOSS in Education and Neural Networks

Prof. Dr Annamma Abraham is currently working as Professor and Heading the Mathematics department of BMS Intitute of Technology, Bangalore, Karnataka, India. Her area of Interests are Fluid Dynamics, Nano Technology, Network Security, Algorithms, Neural Networks

Abhas Abhinav is the founder and hacker-in-charge of DeepRoot Linux, a company he founded in August 2000. His area of interest are Mail Server , Network Security, Information Security, Web Application Development. Distributed Network Management

Sunil V.Gurlahosur has pursued Bachelors Degree in Computer Science & Engineering from Siddaganga Institute of Technology, Tumkur, India.His areas of interest are Computer Networks, Information & Network Security and Web Programming.

Srinivasa Y has pursued Bachelors Degree in Computer Science & Engineering from Siddaganga Institute of Technology, Tumkur, India. His areas of interest are JAVA, J2EE, Web Programming, AJAX, Client-Server Architecture.