International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:5, 2007

A Model of Network Security with Prevention Capability by Using Decoy Technique

Supachai Tangwongsan, and Labhidhorn Pangphuthipong

***Abstract*—**This research work proposes a model of network security systems aiming to prevent production system in a data center from being attacked by intrusions. Conceptually, we introduce a decoy system as a part of the security system for luring intrusions, and apply network intrusion detection (NIDS), coupled with the decoy system to perform intrusion prevention. When NIDS detects an activity of intrusions, it will signal a redirection module to redirect all malicious traffics to attack the decoy system instead, and hence the production system is protected and safe. However, in a normal situation, traffic will be simply forwarded to the production system as usual. Furthermore, we assess the performance of the model with various bandwidths, packet sizes and inter-attack intervals (attacking frequencies).

***Keywords*—**Intrusion detection, Decoy, Snort, Intrusion prevention.

I. Introduction

AT present, network security has become a primary concern in most organizations. To cope with ever increasing fraudulent offenses, computer technocrats relentlessly focus on security measures. IDS is a system which automatically monitors and analyzes the events occurring in computer or network systems for signs of security threats. In this work, the network intrusion detection system (NIDS) is applied, coupled with a decoy system for taking the attacks. When NIDS detects any intrusions, it will signal the redirection module to redirect all traffics causing intrusions to attack the decoy system. All traffics generating intrusions will be restrained from communicating with the actual production system. In the simulation, Snort [7] has been exploited, the de facto standard NIDS, as to perform intrusion detection and prevention. The model performance assessment is conducted through various bandwidths, packet sizes and inter-attack intervals (attacking frequencies). Furthermore, the redirection effectiveness of the model for redirecting intrusion traffics to the decoy system is also investigated.

Manuscript received June 14, 2007. This work was supported in part by the faculty of Science, Computer Science, Mahidol University.

Supachai Tangwongsan is an Associate Professor of the Department of Computer Science, Faculty of Science, Mahidol University, Bangkok, Thailand (phone: 66 02 354-4333; e-mail: ccstw@mahidol.ac.th).

Labhidhorn Pangphuthipong, is with the Department of Computer Science, Faculty of Science, Mahidol University, Bangkok, Thailand (e-mail: labhidhorn@yahoo.com).

II. Literature Reviews

A. Definition and Taxonomy of IDS

Generally, IDS can be classified into several types [11] as follows. Host-based IDS (HIDS) [1, 2, 3 and 11] only monitors activities of the host on which it is running. It utilizes operating system (OS) log files, error messages or service and application rights as sources for detecting intrusions. Network-based IDS (NIDS) [1, 2, 3 and 11] uses network packets on its network segment as a data source. Because NIDS is located in key areas of network infrastructure so it does not need to be installed on every host in the network. Rule-based IDS [1, 2 and 11] tries to identify an attempt to misuse the system by comparing activities with a certain set of intrusion patterns. Anomaly-based IDS [1, 2 and 11] detects intrusions by creating profiles describing the normal behavior of hosts or networks over a period of normal operation. If any occurred activities deviate from the created profiles more than the acceptable threshold, the IDS will raise an alarm. Anomaly-based IDS sometimes can detect new intrusions without having the intrusion database.

B. IDS Performance Evaluation

As IDS technology has become a standard security measurement in most organizations, various research groups have attempted to evaluate the effectiveness and efficiency of IDS. Detection capability [12] is the measurement of IDS detection strength. In other words, what kinds of intrusions the measured IDSs could detect. Scalability measurement [12] is the test to identify at which speed of traffic the IDS begins to drop packets. In false alarm evaluation [12], logs generated by IDS will be analyzed. The ratio between unexpected alarms and total alarms indicates the tendency of IDS in generating a false alarm.

C. Decoy System

Decoy system [8] is an emulated production system aiming to be exposed for intrusions instead of the real production system. In some publications, the word “Honeypot” is used instead of decoy system. Usually, a decoy system is used to lure attackers and study their intrusion tactics. The decoy system is setup to simulate a production system providing services in a data center.

III. System Design

A. Conceptual Design of NIPS

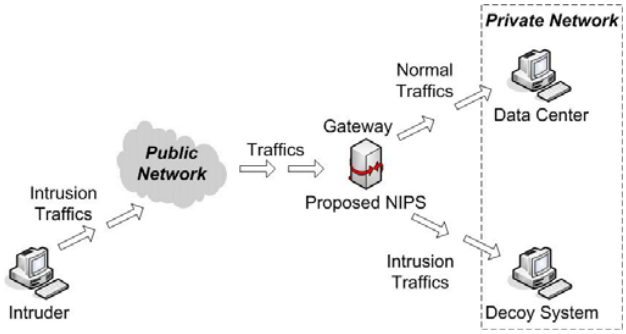

Fig. 1 Malicious traffic redirection

In general, intrusions usually enter a private network from the compromised public network. As depicted in Fig. 1, our proposed NIPS interfaces with three systems, private network, public network and the decoy system. The decoy system should be setup similar to a production system featured with the same characteristics. Intrusions will be trapped and unable to differentiate between the decoy and production system. The proposed system will monitor traffics traversing through it and inspect for signs of intrusions and then route intrusion traffics to the decoy interface with traffic redirection technique.

Traffic redirection is the technique to change the destination of traffic from one route to another as desires. As the proposed system sits inline as a gateway to the private network, every traffic entering the private network has to pass and be examined by the NIPS. At gateway, the system is composed of NIDS and redirection module. When NIDS detects intrusions, it will signal redirection module to redirect all traffics from IP address(es) initiating intrusions to the decoy system.

All traffics from IP address(es) initiating intrusions will be quarantined from communicating with the actual production system in the private network within a pre-defined time. After the predetermined quarantining period, it is back to normal, in which any non-malicious banned IP address(es) could again communicate with the production system.

B. Network Intrusion Detection (NIDS)

The NIDS main tasks are to capture packets, detect intrusions and output intrusion alerts. In the design, when NIDS detects intrusions, it will trigger a set of alert information. After that the alert information will be extracted and IP address will be stored in a database processed by the redirection module.

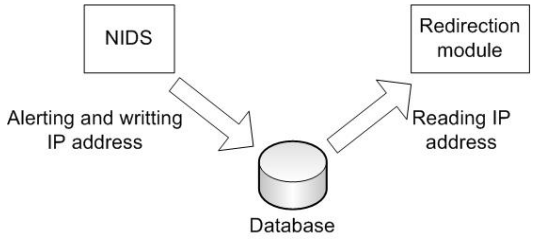

Fig. 2 Communication between NIDS and redirection module

C. Redirection Modules

Our redirection module consists of two sub-modules for performing different processes, traffic redirection and redirection recovery. Traffic redirection receives intrusion IP address from NIDS and redirect traffic from intrusion IP address to the decoy system. Redirection recovery resets the redirection activity of an IP address. When the predefined period of redirection is over, the redirection recovery will recover the redirection state and go back to normal.

D. Decoy System

The decoy system is setup to simulate the target production system in the private network. Conceptually, the decoy system emulates some characteristics of entire production LAN on a single host. It simply shares the same IP address and is installed with the software or services similar to the production system. Because the decoy system shares the some features with the real production system, the intrusions would attack the decoy system instead (as the mission is accomplished).

IV. System Simulation

This section describes the NIPS model simulation. The prototype system is simulated on a gateway system connecting to three network systems, public, private and decoy system. In system simulation, each private and decoy system is simulated on a single hardware. The complete NIPS prototype system requires us to simulate three sub-system simulations as follows.

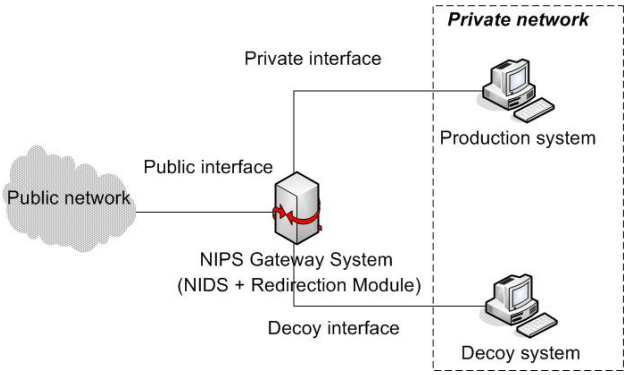

Fig. 3 Network simulation diagram

A. NIDS Simulation

In NIDS simulation, we have exploited Snort intrusion detection capability. The version of Snort used is 2.0.2 which

supports users to customize output-plugins on demand. The Snort's output-plugins is modified to extract only intrusion IP address by attaching C codes to Snort's C codes. These codes, brought from Bait & Switch project [10], are modified to suit with our simulation. To customize output-plugins, the Martin Roesh's instruction from Snort official website [5] has been followed.

B. Redirection Module Simulation

In order to route packet as desires, some information is required to be appended to **rt_tables** in Linux's **iproute2** [14] facility only once. In **iproute2**, ip rule is the routing policy database (RPDB) management that controls route selection algorithm while ip route is the routing table management. The RPDB explicitly allows matching packet source address, packet destination address, TOS, incoming interface and using fwmark values for matching IP protocols and transport ports. **"fwmark"** helps us to mark network packet with a certain number for using in a special purpose.

Some C codes need to be programmed to co-operate Snort and redirection modules. The redirection module has two processes running concurrently. The first process is responsible for reading intrusion IP address from file and redirecting traffic. The second process checks for expiry IP address quarantined in banning list and reroute expiry traffic back to the production system. In traffic redirection, we use **fwmark** facility to mark IP address to be routed to decoy system.

C. Decoy System Simulation

The decoy's IP address setting is the same with the production system and installed with a website providing file browsing and downloading. The hardware specification of both production and decoy system are the same as NIPS prototype system.

V. Experimental Results

This section describes experiments conducted through our research. The first experiment presents intrusion detection by Snort NIDS and the second experiment is to simulate the prevention of attacks to the private network as the proposed model.

A. Intrusion Detection with Snort

Target network is monitored by installing Snort on the target network. Attacks are launched from attack network to target network. In the experiment, three tools are used to generate attacks, Striker, Elite and Engage packet builder. The Engage packet builder is not the attacking tool but it can help us to craft packets with on demand. Then, we also deliberately craft a normal TCP packet with both IP identification and sequence number with 3868 and send it to the target machine.

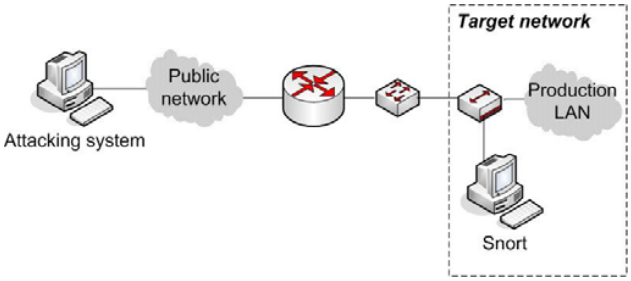

Fig. 4 Network simulation diagram

After a certain period of launching attacks, the first two attacking tools produced usual intrusion alerts as expected. The problem is that a crafted normal traffic launched by the Engage packet builder is alerted as an attack. Because IP identification and sequence number of 3868 are marked as one of the Snort's intrusion signatures called "DoS Land attack". In this case, the alerts of normal traffic are called false alarm which is one of the main problems of IDS today.

B. Intrusion Prevention with Snort and Decoy System

In order to prevent intrusions by using Snort and decoy system, the simulation details have been elaborated in the previous section. In the experiment, network simulation is composed of five machines, production host, decoy system, gateway. The two remaining machines are assumed to be from public network intending to communicate with production host in private network. Both production host and decoy system have the same IP address but connecting to the different network interfaces. The decoy system is also set up with another Snort for checking the results.

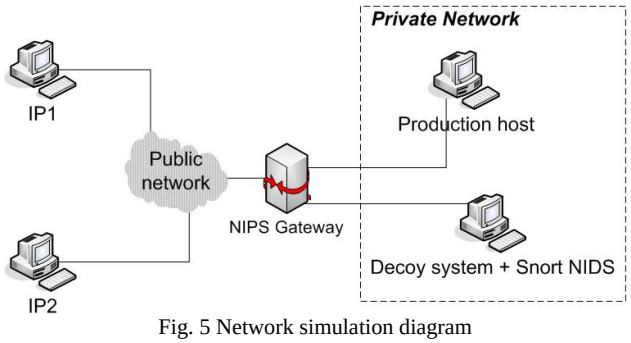

Fig. 5 Network simulation diagram

A web site is provided on both production host and decoy system which clients from public network can browse to download files. Then we begin with downloading file from production host by machines, IP1 and IP2, from simulated public network. When checking at the LAN connection status of each host in public network, it states that both machines are communicating with the production host.

After that, a kind of attacks is launched for one time, from machine IP2. An alert is found on Snort at the decoy system and all subsequent traffics from IP2 are redirected to decoy system for 60 seconds while IP1 is still communicating with the real production host.

After a predefined redirection period, machine with IP2 could again communicate with the real production host. An

alert of Snort at decoy system confirms the successful redirections of the prototype system.

C. Performance Measurement of the Proposed NIPS

Scalability measurement and redirection capability are considered as the key performance indicators of the prototype system. Another dropping factor to be observed is attacking frequency. It is believed that inter-attack interval has an effect on dropping percentage. For the redirection capability, the effectiveness of system for redirecting malicious traffics to the decoy system is the major concerns. The first experiment is to individually evaluate Snort performance as depicted in Fig. 6.

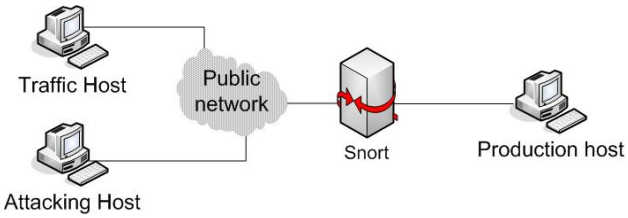

Fig. 6 Network emulation diagram of individual Snort test

Snort is stressed with various packet sizes of background traffic (10 - 50 Mbps) and inter-attack interval of 1 second for identifying at which environment that Snort will output an acceptable dropping percentage. Traffic host generates background traffic for 60 seconds in various rates for stressing Snort. Attacking host launches attacks with inter-attack interval of 1 second for 10 times. Some parts from each bandwidth of the individual Snort evaluation are shown in Tables I and II.

TABLE I
PACKET SIZE OF 512 BYTES AND INTER-ATTACK INTERVAL OF 1 SECOND

Speed of background traffic (Mbps)	Packets analyzed	Total packets launched	Packets dropped	Packets dropped (%)	Alerts of Snort (Number of times)
10	291699	292350	651	0.223	20
20	579131	581939	2808	0.483	20
30	839370	843975	4605	0.546	20
40	1097644	1103876	6232	0.565	20
50	1359815	1368460	8645	0.632	17

TABLE II
PACKET SIZE OF 1024 BYTES AND INTER-ATTACK INTERVAL OF 1 SECOND

Speed of background traffic (Mbps)	Packets analyzed	Total packets launched	Packets dropped	Packets dropped (%)	Alerts of Snort (Number of times)
10	146016	146016	0	0	20
20	291915	292081	166	0.057	20
30	437873	437873	0	0	20
40	587506	587506	0	0	20
50	729654	729778	124	0.017	20

Two tables above express the relation between packet size and Snort dropping percentage that is the bigger packet size results the more dropping percentage.

The second experiment is the evaluation of proposed NIPS. To perform a performance measurement, four performance indicators are introduced as follows.

(a) Dropping percentage When Snort is stopped to perform intrusion detection task, Snort will summarize and display all statistics relating to the previous intrusion detection.

(b) Redirection capability percentage This indicates intrusion prevention effectiveness of our proposed NIPS.

(c) Intrusion penetration percentage This parameter describes intrusions erroneously prevented by our system which enter private network.

(d) Gateway mishandling percentage This parameter indicates the gateway mishandling percentage, in this context, that our NIPS may destine the normal traffics to the decoy system.

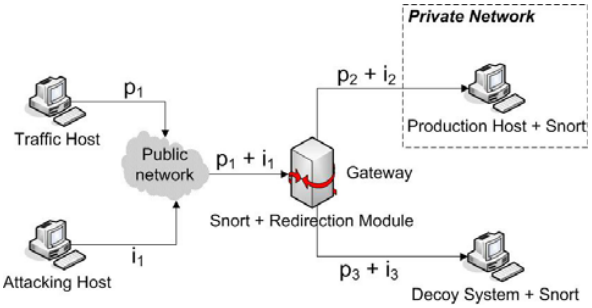

Fig. 7 Network emulation diagram of NIPS measurement

From the Fig. 7, normal and intrusion traffics originated from different points are substituted as notation **p** and **i** respectively.

$\mathbf{p_1}$ and $\mathbf{i_1}$ are Traffic from Public to Gateway.
$\mathbf{p_2}$ and $\mathbf{i_2}$ are Traffic from Gateway to Private.
$\mathbf{p_3}$ and $\mathbf{i_3}$ are Traffic from Gateway to Decoy.

With the above notations, performance of the proposed NIPS can be theoretically figured out with the following formulas.

(a) Redirection Capability Percentage = $(\mathbf{i_3} / \mathbf{i_1}) * 100$.
(b) Intrusion Penetration Percentage = $(\mathbf{i_2} / \mathbf{i_1}) * 100$.
(c) Gateway Mishandling Percentage = $(\mathbf{p_3} / \mathbf{p_1}) * 100$.

Similar to the individual Snort evaluation, traffic host generates background traffic for 60 seconds in various rates for stressing Snort and redirection module (10, 50 and 90 Mbps). Simultaneously, attacking host attacks with attacking frequencies of every 1, 3 and 5 second, for 10 times. To record $\mathbf{i_2}$, $\mathbf{i_3}$ and $\mathbf{p_3}$, Snort is installed at both production and decoy system.

In the experiment, several packet sizes ranging from 256 – 1024 bytes have been used. We have tried to control the dropping percentage to be lower than 1.5 % as the satisfactory level and found that with packet size of 896 byte system begins to conform to that rate. The results of some rows reveal the strange results. Usually, $\mathbf{i_1}$ would be greater than or equal to $\mathbf{i_2} + \mathbf{i_3}$, but in some rows $\mathbf{i_1}$ were less than $\mathbf{i_2} + \mathbf{i_3}$. This indicates that Snort at gateway failed to analyze some intrusion packets. When the redirection module flagged an IP

address as a source of intrusions, all subsequent traffic, even the intrusions, are automatically routed to the decoy system. This indicates that the alerts from Snort at production and decoy system not to be affected from high dropping percentage of Snort at gateway. That's why, in the experiment, the number of $\mathbf{i_1}$ are less than $\mathbf{i_2} + \mathbf{i_3}$.

Tables III, IV and V show some parts of the experimental results with traffic rate of 10, 50, 90 Mbps, packet size of 1024 bytes and inter-attack interval of one second. All averages of experimental results are summarized in Tables VI and VII.

TABLE III
EXPERIMENTAL RESULTS WITH 10 MBPS BW AND 1024 BYTES PACKET

Speed	Drop %	p_1	i_1	i_2	p_3	i_3
10 Mbps	0	72963	20	1	0	19
10 Mbps	0.002	72931	20	1	0	19
10 Mbps	0	74176	20	1	0	19
10 Mbps	0	73080	20	1	0	19
10 Mbps	0	74179	20	1	0	19
Average	0.0004	73465.8	20	1	0	19

TABLE IV
EXPERIMENTAL RESULTS WITH 50 MBPS BW AND 1024 BYTES PACKET

Speed	Drop %	p_1	i_1	i_2	p_3	i_3
50 Mbps	0.004	364796	20	1	0	19
50 Mbps	0	364812	19	1	0	18
50 Mbps	0	364812	19	1	0	18
50 Mbps	0	364812	20	1	0	19
50 Mbps	0	364783	19	1	0	18
Average	0.0008	364803	19.4	1	0	18.4

TABLE V
EXPERIMENTAL RESULTS WITH 90 MBPS BW AND 1024 BYTES PACKET

Speed	Drop %	p_1	i_1	i_2	p_3	i_3
90 Mbps	0	656684	20	1	0	19
90 Mbps	0	656737	20	1	0	19
90 Mbps	0.006	655293	19	1	0	18
90 Mbps	0	656591	20	1	0	19
90 Mbps	0.001	661212	20	1	0	19
Average	0.0014	657303.4	19.8	1	0	18.8

TABLE VI
AVERAGE RESULTS FROM PACKET SIZE 896 BYTES OF BACKGROUND TRAFFIC

Speed (Mbps)	Dropping %		
	1 Sec.	3 Sec.	5 Sec.
10	0.0452	0.0195	0.0105
50	0.0569	0.0244	0.0163
90	0.0572	0.0252	0.0184
Speed (Mbps)	Redirection Capability %		
	1 Sec.	3 Sec.	5 Sec.
10	95	94.47	95
50	94.94	94.44	93.47
90	94.26	94.26	93.84
Speed (Mbps)	Intrusion Penetration %		
	1 Sec.	3 Sec.	5 Sec.
10	5	5.52	5
50	5.05	5.55	6.52
90	5.23	5.23	6.15

TABLE VII
AVERAGE RESULTS FROM PACKET SIZE 1024 BYTES OF BACKGROUND TRAFFIC

Speed (Mbps)	Dropping %		
	1 Sec.	3 Sec.	5 Sec.
10	0.0002	0	0
50	0.0005	0	0
90	0.0008	0.0002	0.0002
Speed (Mbps)	Redirection Capability %		
	1 Sec.	3 Sec.	5 Sec.
10	95	95	94.9
50	94.6	94.7	94.8
90	94.6	94.9	92.3
Speed (Mbps)	Intrusion Penetration %		
	1 Sec.	3 Sec.	5 Sec.
10	5	5	5.05
50	5.1	5.2	5.1
90	4.55	3.57	7.6

The average dropping percentage of zero in Table VII are carried out from a set of the experimental results which, through our set of experiments, all dropping percentages are at zero. This implies that the average dropping may not exactly be at zero but close to zero percentage. Finally, the average dropping percentage can be summarized from various traffic rates and attacking frequencies from the experiment as shown in Figs. 8 and 9.

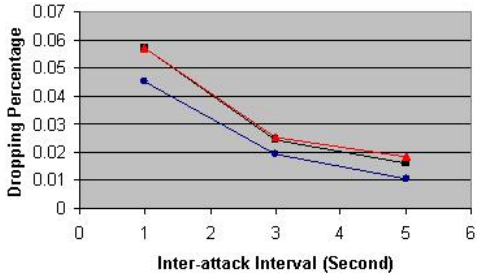

Fig. 8 Dropping tendency of packet size 896 bytes

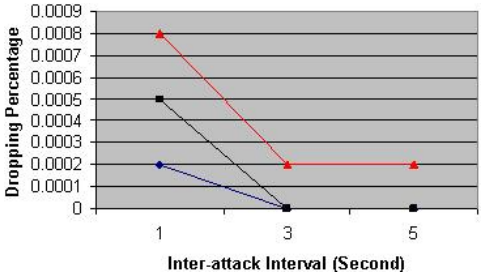

Fig. 9 Dropping tendency of packet size 1024 bytes

The results summarized from Tables VI and VII indicate that NIPS prototype can prevent intrusions with the effectiveness in the neighborhood of 92 - 95 %. The intrusion traffics can penetrate to private network with the percentage around 4 - 8 % and the gateway does not allow any percentage of gateway mishandling.

Tables I – VII and graph in Figs. 8 and 9 give a conclusion that the dropping tendency will lift up when traffic rate is higher and higher and will mitigate when inter-attack interval increases.

VI. Discussion and Conclusion

This section presents an overall discussion and conclusion in accordance with the experimental results previously presented in section 5.

A. Discussion

The discussion mainly focuses on the following aspects namely, performance, resource usage, program complexity and system error and effectiveness.

1) System Performance

From the experimental results, it is observed that individual Snort has a better dropping percentage than the Snort coupled with redirection module. With the packet size of 512 bytes, the individual Snort gives the satisfactory dropping rate while Snort coupled with redirection module need to be fed with packet size of 869 bytes. Though the bandwidth is increased to 90 Mbps, the system produces a satisfactory rate of dropping percentage lower than 1.5 % with packet size up to 869 bytes and over. Due to the fact that Snort is exploited as NIDS, the overall performance of our system depends upon Snort performance as well.

2) Program Complexity

Snort plays a key role of program complexity. Roughly, Snort contains around 90,000 lines of C codes functioning intrusion detection tasks. The attached codes of output-plugins and redirection module are around 550 lines of C codes. In network programming, inter-process communication is the basic methodology for co-operating two processes working together. In prototype system, co-operation between Snort and redirection process is established through a file containing intrusion IP address outputted from Snort process. The redirection module requiring pthread programming has two processes running concurrently. The first process is responsible for reading intrusion IP address from file and redirecting traffic. The second process checks for expiry IP address quarantined in banning list and reroute expiry traffic back to the production system.

3) System Error and Effectiveness

As discussed in section 5 that four performance indicators are used to measure NIPS prototype system namely, dropping, redirection capability, intrusion penetration and gateway mishandling. The last three indicators help identify the effectiveness of prototype system. The three performance indicators derived from the experimental results are summarized in the Tables VI and VII in section 5. When Snort detects intrusion packets, before the redirection process of IP address initiating intrusions will start, some malicious packets would still penetrate to the production system. From the Tables VI and VII, the experimental results of NIPS prototype system represents that the system has the redirection effectiveness around 92 - 95 % with intrusion penetration of 4 – 8 % and 0 % gateway mishandling.

B. Conclusion

Based on the experimental results conducted in section 5, our NIPS prototype system meets the requirement and achieves the objective to prevent or avoid intrusions penetrating into the private network. Because we rely on using Snort detection capability for intrusion detection processes, the NIPS prototype system performance depends mostly on to Snort performance, dropping tendency. As a result the NIPS prototype system shares the same rate of detection capability, scalability, false alarm and all shortcomings inherited from Snort.

References

[1] Herve Debar, Marc Dacier and Andreas Wespi, Towards a Taxonomy of Intrusion Detection Systems, IBM Research Division, Zurich Research Laboratory, Switzerland, 1998.
[2] Rebecca Bace and Peter Mell, Intrusion Detection Systems, National Institute of Standard Technology, 2001.
[3] Joseph S. Sherif and Tommy G. Dearmond, Intrusion Detection: Systems and Models, Califinia Institute of Technology, JPL, USA, IEEE 2002.
[4] Neil Desai, *'Increasing Performance in High Speed NIDS'*, Snort Official Website, [http://www.snort.org/]; 2002.
[5] Jack Koziol, Intrusion Detection with Snort, Sam Publishing, 2003.
[6] Zhi-Hong Tian, Bin-Xing Fang, Xiao-Chun Yun, An Architecture for Intrusion Detection using Honey Pot. National Computer Information Content Security Key Laboratory, Harbin University of Technology, China, IEEE 2003.
[7] Jack Whitsitt and Alberto Gonzalez, The Bait and Switch Honeypot, [http://baitnswitch.sourceforge.net/]; 2003.
[8] Carl Endorf, Eugene Schultz and Jim Mellander, Intrusion Detection & Prevention, McGraw-Hill / Osborne, 2004.
[9] Leonardo Lemes Fagundes, Luciano Paschoal Gaspary, Network-based Intrusion detection systems Evaluation through a Short Term Experimental Script, IEEE 2004.
[10] Alexey N. Kuznetsov, *'IPROUTE2 Utility Suite Documentation'*, [http://www.policyrouting.org/iproute2.doc.html].