JACR
Journal of Advances in Computer Research
ISLAMIC AZAD UNIVERSITY
SARI BRANCH

Scientific - Research Journal

Sari Branch - Iran

ISC
Islamic World Science Citation Center

Journal of Advances in Computer Research

Volume 3, Number 1, February 2012 ISSN: 2008-6148

www.jacr.iausari.ac.ir

Journal of
Advances in Computer Research

7

Volume 3, Number 1, February 2012

Publisher:
Islamic Azad University, Sari Branch

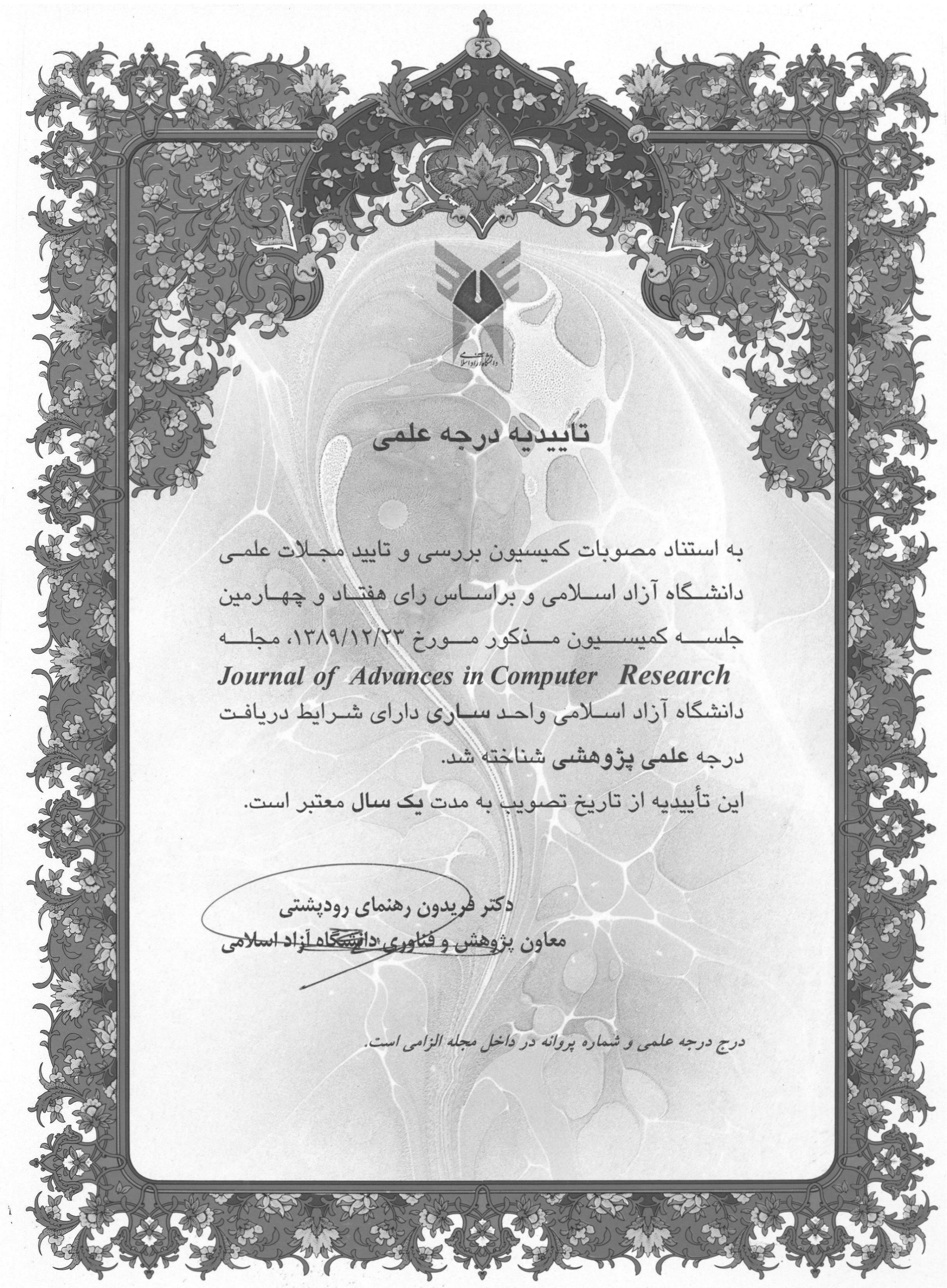
Review Editor:
Hossein Nematzadeh

Layout and uniform designer:
Ahmad Rastegar

Coordinator & Executive Expert:
Hamed Samadi

Address:
Islamic Azad University - Sari Branch, 7'th kilometer
of Darya BLV, Sari, Iran
Electronic Post:
jacr@iausari.ac.ir
Web address:
http://www.jacr.iausari.ac.ir

P.o.Box 48164-194 Telefax: +981512133725



تاییدیه درجه علمی

به استناد مصوبات کمیسیون بررسی و تایید مجلات علمی
دانشگاه آزاد اسلامی و براساس رای هفتاد و چهارمین
جلسه کمیسیون مذکور مورخ ۱۳۸۹/۱۲/۲۳، مجله
Journal of Advances in Computer Research
دانشگاه آزاد اسلامی واحد **ساری** دارای شرایط دریافت
درجه **علمی پژوهشی** شناخته شد.
این تأییدیه از تاریخ تصویب به مدت **یک سال** معتبر است.

دکتر فریدون رهنمای رودپشتی
معاون پژوهش و فناوری دانشگاه آزاد اسلامی

درج درجه علمی و شماره پروانه در داخل مجله الزامی است.

- The Journal of Advances in Computer Research has been assigned the ranking of " scientific- research journal" regarding the 74th meeting of the verification commission for Islamic Azad University journals in 13/3/2011
- This journal is also indexed in Islamic World Science Citation Center (ISC) (www.isc.gov.ir) and Scientific Information Database (SID) (www.sid.ir).

Journal of Advances in Computer Research

Vol. 3, No. 1, February 2012

Content:

Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 3, No. 1, February 2012), Pages: 65-79
www.jacr.iausari.ac.ir

Avoiding Cyber-attacks to DMZ and Capturing Forensics from Intruders Using Honeypots

Ali Salimi [1], **Peyman Kabiri** [2]✉

(1) Electronic Learning Center, Iran University of Science and Technology, Tehran, Iran
(2) School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

a_salimi@comp.iust.ac.ir; peyman.kabiri@iust.ac.ir

Received: 2012/01/03; Accepted: 2012/01/24

Abstract

Nowadays, honeypots are widely used to divert attackers from the original target and keep them busy within a decoy environment. DeMilitarized Zone (DMZ) is an important zone for network administrators, because many of the services to the public network is provided at this zone. Many of the security tools such as firewalls, intrusion detection systems and several other security systems can be used to secure DMZ. But honeypots are supplementary devices used to discover attacks and capture forensics against the attackers. The most important solution to secure the DMZ is to detect attacks against servers of this zone and void these intrusions by leading them to honeypots and capturing enough forensics against the attackers. This research work is focused on providing a solution for problem areas such as response to intrusion attempts and redirection of the intruders to honeypots. The proposed system detects malicious activities and redirects them to a decoy system to capture forensics. Honeypots are decoy systems used to interact with attackers and capture forensics from their activities. In the reported work, detection of the malicious activities is carried-out using a Network-based Intrusion Detection System (NIDS). Measuring performance of the proposed system, three important factors are implemented. These factors include accuracy, false positive rate and true positive rate. Accuracy is presented as an important factor to check the performance of the system. In our simulations, the measured accuracy is more than 99 percent. False positive rate is another important factor of this system that shows the failure rate. This parameter is measured less than 0.50 percent that shows the proposed system cannot detect all the attacks against the protected machine, but attack detection is performed using a suitable rate. The last factor of system performance is true positive rate that is measured to be 100 percent. This measurement shows that all of the legitimate traffic is directed to protected machine with proposed system.

Keywords: *Intrusion Detection, Forensics, Demilitarized Zone, Honeypot*

1. Introduction

Computer networks are widely used in many companies, so the security of these networks is a very important issue. Viruses, worms, intrusion attempts, denial of service attacks and spam are some of the best examples for these security concerns. Demilitarized Zone (DMZ) is one of the most important zones in network that contains important servers providing different types of services for users connecting from public network to private zone [1]. Web server, mail server and DNS server are samples of servers lying in DMZ where they provide important services for the Internet users. In the other words, DMZ is like a showroom to show the services of a network to outside visitors. Secure this zone, different security tools can be used; each working in a particular layer. Firewalls, intrusion detection and prevention systems, honeypots and antivirus systems are among the most famous tools in this area. Firewalls contain important rules to control the traffic from the public network to DMZ. Firewalls implement traffic control operation using a set of rules called rule table. These rules accept connections according to the rules and reject other connections that are not allowed in accordance to the firewall rules. But only using a firewall as a traffic controller is not enough, since attackers may change their address and introduce themselves as legitimate users. Other tools such as Intrusion Detection Systems (IDS) are necessary to complete the system security in the other layers. IDS is a security tool that detects intrusion attempts in a system or in a network. IDS monitors the traffic passing through a system or from a network real time and detects intrusion attempts that may occur. Honeypots are decoy systems [2, 3] whose value is to amuse the attackers in a controlled environment with no valuable data or service. To achieve this goal, it is very important to place the honeypot in a suitable location, because it must be viewed by the attacker as a normal system similar to the one that is likely the target of attackers e.g. servers in DMZ. Nevertheless, the decoy system may not be detected by attacker. The most important reason to use honeypots in DMZ is to view the tools and tactics of attackers and to capture enough forensics from their activities. Consequently, redirecting attackers to the honeypot in such a way that the attacker does not notice it is of a great importance [4, 5, 6]. Another important issue is to design the honeypot in a way that the attacker cannot detect that he or she is in a decoyed system.

In the reported work, a new architecture is proposed and implemented to recognize the attackers to DMZ and to redirect them to a decoy environment and capture useful forensics from their activities. The implemented system consists of firewall, Network-based Intrusion Detection System (NIDS) and an INTelligent Redirector System (INTRS). NIDS is used to detect intrusion attempts. INTRS redirects the malicious traffic to the honeypot once an intrusion attempt is detected by the NIDS. Later on, once enough information is collected by the honeypot, firewall is updated to recognize the ip address of attacker and to block it. In this work, Snort is selected as the NIDS by which malicious traffic is recognized from the legitimate traffic. Measuring efficiency of the system, the proposed architecture is tested using various types of traffic and with different packet sizes and types of malicious traffic packets. In these experiments efficiency of the INTRS module is assessed using an IDS installed on the honeypots.

According to the definition of honeypot by Lance Spitzner, "A honeypot is security resource whose value lies in being probed, attacked, or compromised." [2]. The most important problem by honeypots is to ensure their use by the attackers. With use and implementation of this system, attackers are unknowingly forced to use the honeypot and interact with it.

2. System Overview

To introduce the proposed system, the first step is to explain the system components. The text will also explain the operation of each component.

2.1 Firewall

Firewall is a security system that is used to control the access from a public zone to a private zone like DMZ. Firewalls provide access control and prevent users from unauthorized access to networks connected to the Internet [7]. Two important categories of firewalls are hardware firewall and software firewall. There are different vendors producing hardware firewalls that are stand-alone products. Software firewall is a protecting tool that is installed separately on a host. A set of rules can be defined and used by firewall to control the passing traffic and avoiding unauthorized accesses.

2.2 DeMilitarized Zone (DMZ)

Demilitarized zone is a security zone designed to make a safer zone to place the important servers providing services to the Internet users. DMZ is an important goal for attackers because most of the important services are located at this zone.

DMZ is the placement of servers providing different types of services for public network such as web, email and Voice Over IP (VOIP). If combination of two layers of firewalls is used, the DMZ is placed between these two bastions. Figure 1 shows the placement of DMZ between two firewalls.

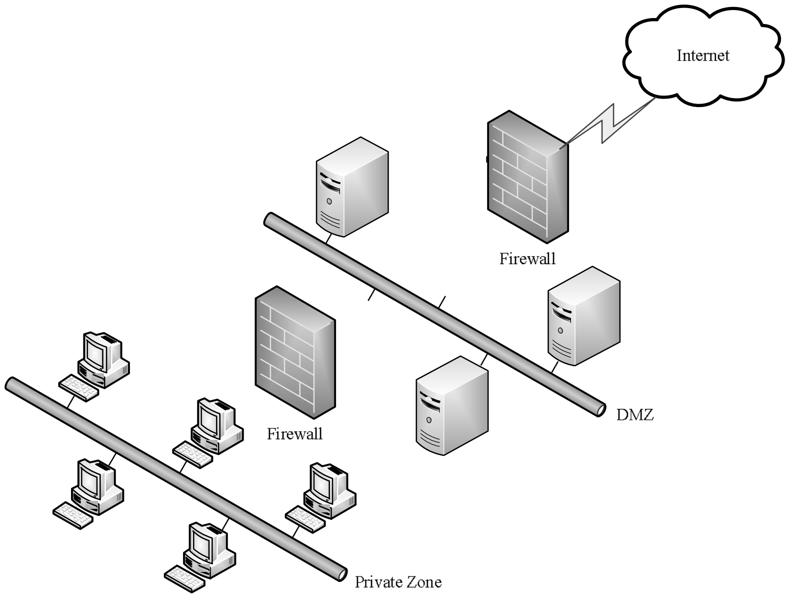


Figure 1. The location of DMZ between two firewalls

If only one firewall is used in the network, the DMZ is connected to an interface of the firewall called DMZ interface. Figure 2 presents this implementation.

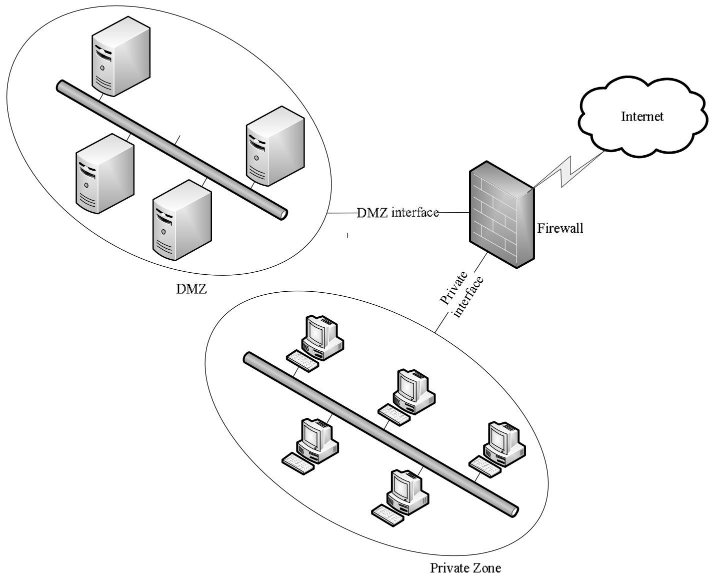


Figure 2. Location of DMZ behind one firewall

In both of these scenarios as shown in figure 1 and 2, DMZ is placed behind the firewall and is protected using one security level i.e. firewall [1].

2.3 Intrusion Detection System

IDS is a monitoring and alerting system that monitors network traffic or system activities and reports detected malicious activities to a management system. IDS can be classified into two types; one is the Host-based IDS (HIDS) and the other one Network-based IDS (NIDS). HIDS runs on individual hosts or devices on the network to only monitor the network traffic on that host and reports suspicious activities. NIDS is like a black box connected to the network and its network module is in promiscuous mode. It is very important to place NIDS in a suitable point or points within the network to monitor the network traffic. Traffic passing from a switch is a good feed for an NIDS. The main operation of a NIDS is to monitor the input and output traffic. However, this procedure can create a bottleneck at installation point of NIDS. Low computational power of an NIDS in high speed networks may force it to slowdown traffic of the packets or to drop them either randomly or systematically.

From another point of view, there are two types of IDS [8-9]:

I) Signature-based method: IDS monitors traffic of the packets in a network and compares them against predefined and preconfigured attack patterns known as signatures [10]. Rule-based and graph-based approaches are two common methods used in this type of IDS.

II) Anomaly-based method: IDS detects intrusions by comparing current behavior of the network versus normal behavior of the network. The anomalous behavior can vary in different situations in contrary to the rule-based system that uses predefined signatures and compares the network traffic with them. Zero-day attacks may also be detected using anomaly-based IDS [11-12].

Snort [13, 14] is a rule-based NIDS. It detects intrusions attempts and generates alerts using predefined rules in its database.

2.4 Honeypot

Honeypot is a trap, designed to entrap intruders [15]. Honeypot is a decoy system and it is placed in the network to lure intruders. Once honeypot is attacked by an intruder, his/her actions are captured in honeypot. Consequently, it will be possible to monitor the behavior and activities of that attacker and capture enough forensics against the attacker. At the same time, honeypot will buy time for the administrator to respond by keeping the intruder busy while alerting the network administrator. There are three general types of honeypots: low-interaction honeypots, medium-interaction and high-interaction honeypots [2-3], [15].

Low-interaction honeypots are used to simulate some protocols and cannot provide a full access to the honeypot by the attacker. Medium-interaction honeypots work at the service level and simulate some services. They provide a similar environment for attackers because of providing a higher level of interaction with them.

High-interaction honeypots are complete systems with a full operating system. They are designed to simulate a system with all of its components. They provide a vast amount of information about the attackers and their behavior. This feature will increase risk of larger attacks such as a DoS attack to the honeypot and danger of losing the control over the honeypot [16]. Honeynets are an example of high-interaction honeypots.

2.5 Redirector systems

Redirector is used to change the path of malicious traffic to the honeypot. Attackers mainly target major systems within DMZ such as mail servers, web servers and DNS servers. A good example of attack redirection is depicted in figure 3 in which a honeypot is placed near a mail server to detect attacks against the mail server.

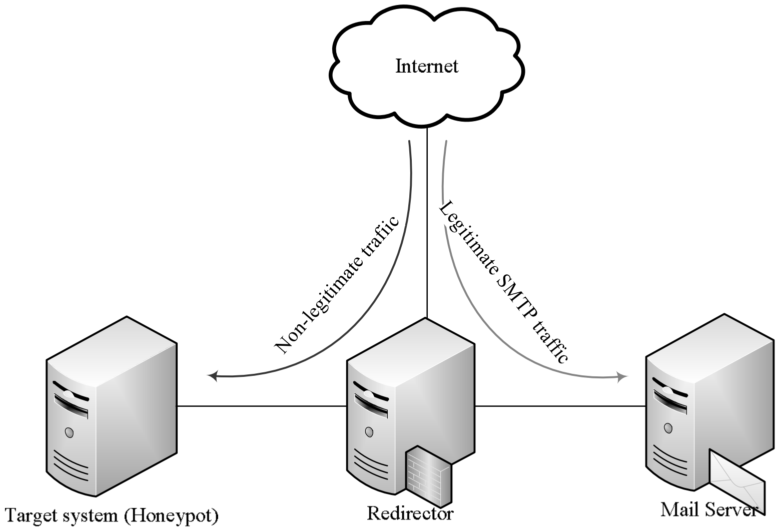


Figure 3. Redirection of attacks against a mail server

According to figure 3, the redirector system is designed in such a way that it can change path of the traffic destined for port 25 to a different system.

As a part of honeynet project, similar redirector systems such as honeywalls, are also designed to redirect malicious traffic to honeypots [17]. As the honeynet project site indicates "The Honeynet Project is a leading international security research organization, dedicated to investigating the latest attacks and developing open source security tools to improve Internet security" [17]. Honeywall is a package presented as Honeywall CDROM by Honeynet team. Honeywall controls the input and output traffic of the network as a gateway and has four important responsibilities. These responsibilities are data control, data capture, data analysis and data collection [17]. One of the most important failure points of honeynet is when attackers attempt to break into the honeypot. Here they can use the honeypot to attack to the other system resources and harm them using larger bandwidth and more availability. The proposed system that will be explained in details in section 3, avoids this failure by separating the network interface of server from the honeypot and isolating them. Another failure point for high-interaction honeypots is the risk of detection. To reduce this risk, in the proposed architecture, honeypot is simulated similar to a DMZ server. This solution will reduce the risk of honeypot being detected by the attackers.

An important difference between the proposed architecture and the honeynet system is explained in the following. In honeynets, there is a network of honeypots. Each honeypot can be used by the attackers to attack other honeypots or the protected servers. But in the proposed system, there is no connection between the honeypot and the main protected system. Having the honeypots isolated, a big problem in honeynets is solved by this architecture. In the proposed architecture, the IP address of honeypot is equal to the IP address of the protected main system in DMZ without any network conflict. Honeypot is also used to capture forensics from the attackers and detect their new tactics. Another important problem with the honeypots solved by the proposed architecture is the problem of transparency where the attacker is redirected to honeypot.

3. Another important difference between the proposed architecture and honeynet is the use of Snort as IDS in the proposed architecture. Honeynets use honeywall as a preconfigured package which is both firewall and IDS and also redirects packets. But in the proposed architecture, firewall, IDS and redirector are separate from each other. Snort is the best open source IDS that is used to detect attacks. IPtables is the free firewall used by Linux operating system and our redirector module is a perl code that can be upgraded in future to have more features such as packet dropping and so on. The most important difference between this architecture and honeynet is that the honeypot in the proposed system is completely similar to the main system in DMZ. Even the IP address of our proposed honeypot is the same as the protected system. **System Architecture**

To explain the operation of the designed system, it is most important to know the relationship between these parts of the architecture and understand the operation and reaction of the system when attacks occur.

3.1 Architecture Design

Attackers usually use public networks to attack the systems providing services for users. To divert attackers from the main servers in DMZ and redirecting them to a decoy system, the proposed system is designed as in figure 4. Architecture of the system consists of one input interface and an even number of output interfaces. Output interfaces are used to lead the traffic either to a server or to a honeypot that is designed to protect from that server.

As shown in figure 4, a honeypot is used to act as the server that is placed in DMZ. If an attacker decides to attack the server, the proposed system detects the attack using

Snort IDS and re-routs the traffic from targeted server to the decoy system that is called the honeypot.

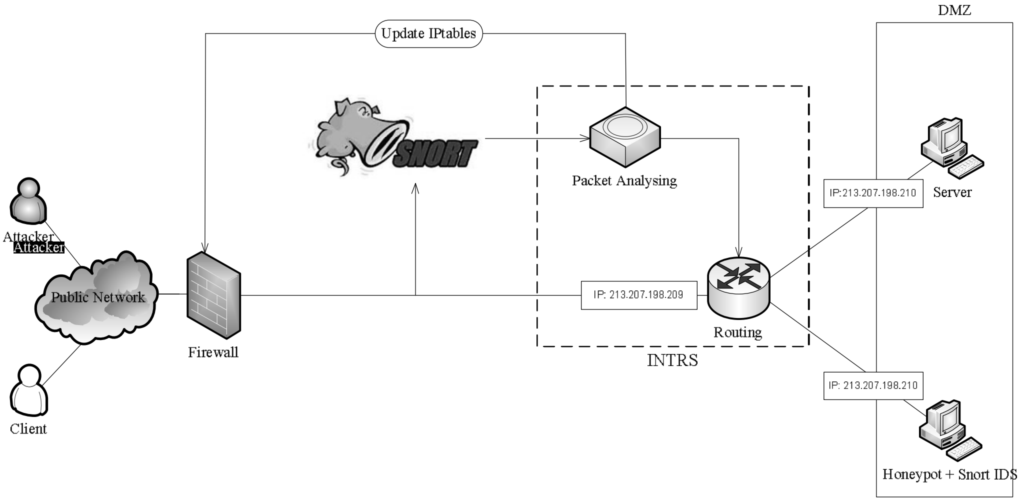


Figure 4. The proposed architecture

Another important technique used in the designed system is the solution used to redirect the proposed traffic using INTRS. In this solution, path of the traffic is changed from one destination to another as required. The proposed system is placed in the path of traffic from public network to DMZ, sniffing the packets and making decision about the appropriate destination for the traffic. If the traffic is normal, no attack is detected and the main server is selected as for the destination. Otherwise, when a malicious behavior is observed, the selected destination is the honeypot.

3.2 Network-based Intrusion Detection System

To sniff the traffic passing through the designed system and detect malicious activities, a rule-based NIDS is used. The output of NIDS is an alert which contains the information about the attack and the IP address of the attacker. This alert is used by INTRS to make decision about the appropriate route and the correct response that should be sent to the firewall.

3.3 INTelligent Redirector System (INTRS)

After detecting an attack using NIDS, the proposed system should redirect the malicious traffic to the decoy system. The main operation of INTRS is to take the input data from the output of NIDS which is an alert and then redirect the packet traffic to the designated destination in accordance with the information within the alert. If the alert log file is empty, it means that there is not any detected intrusion and the traffic should pass through its original path to the server. If the NIDS alert is triggered and the alert will contain information about intrusion attempt, INTRS will then redirect the malicious traffic to the honeypot. Consequently, all the attacker activities are logged in the honeypot. INTRS also sends updates to firewall to upgrade its rules in order to avoid more attacks from a distinct IP address. In this way, the attack is detected and redirected to honeypot to capture forensics from the attacker's activities and the rules of firewall are updated by which system will be protected against similar attempts from that address in the future.

3.4 Enforcing the Honeypot on Attackers

Honeypot is the decoy system designed to be placed in DMZ to secure the servers from the attacks. It is also an exact copy of the protected sever, because it should exactly look like that server from the attackers' point of view. In this scenario, the attacker will attack the honeypots thinking that he/she is attacking the DMZ servers. One of the most important problems solved by the proposed system is to force the attackers to interact with honeypot.

4. System Implementation

The implementation of the system architecture is discussed and explained in this section. According to figure 4, the solution to make a secure DMZ is to trap attackers and redirect their malicious traffic to the honeypot to capture forensics and preventing them from attacking the main server providing services in DMZ. The proposed system is implemented with virtual machines connecting to each other using host-only network adapters.

4.1 Attacker and user machines

Attacker and user machines are designed to simulate an access to the DMZ from a public network. In this scenario, these machines are placed in the public zone. Linux Fedora 14 is used as the operating system in this simulation. On the attacker side, some attacks are simulated to test the system. Similarly, the protected traffic is produced by the user to test the operation of the proposed system.

4.2 The proposed system and its components

The proposed system is the most important part of the simulation and acts as the brain of this architecture. This system consists of three modules: firewall, NIDS and INTRS.

Firewall is used to control IP packets passing through a system. Using Linux operating system, iptables can be used as a built-in firewall. Iptables is a software firewall, which can be used to control the flow of traffic from a public network to the trusted private network such as DMZ. To configure this flow control, iptables uses a set of rules. These rules define path of traffic with respect to the alerts generated by the NIDS. If NIDS detects malicious traffic, it will trigger an alert and INTRS will update firewall rules to deny attacker accessing the DMZ as well as redirecting it to the honeypot. This policy will guarantee the security of both the DMZ and the server. NIDS is used to detect subversive activities. At this implementation, Snort version 2.9.0.4 with General Public License (GPL) is used as IDS. Snort is one of the most familiar open source network-based intrusion detection systems (NIDS). Figure 5 shows the structure of Snort IDS.

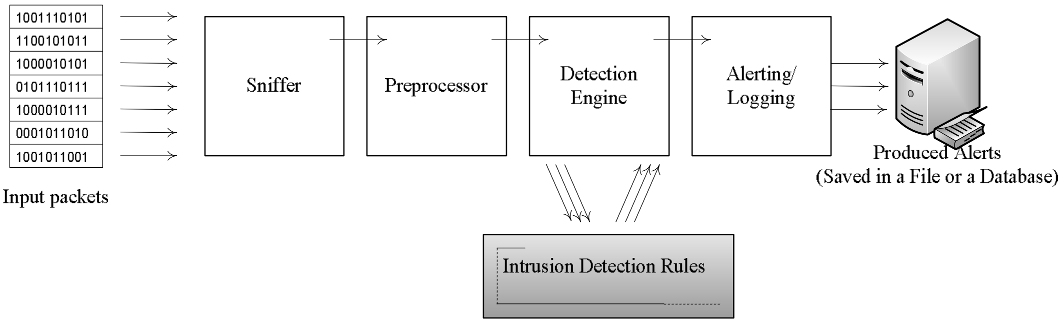


Figure 5. The Structure of Snort IDS

Snort consists of four main parts. Sniffer listens to the traffic. Snort's network traffic interface operates in the promiscuous mode. To control the special behavior of the packets, preprocessing is required. The preprocessor unit receives packets from sniffer and compares them with some of its predefined plug-ins. If the packet's behavior is similar to the plug-ins, preprocessor will send that packet to the detection engine. Detection engine compares packets received from preprocessor against its defined rules. If packet makes any rule to fire, the alert generator unit is called and the corresponding alert is produced and is recorded in a database.

Snort can analyze traffic of the network and log all the passing packets via the IP protocol. Snort also can analyze protocols, search the contents of IP packets and match their contents against its rules. For each known attack, Snort uses a unique Signature ID (SID) that is defined to present that attack as a rule.

As presented in figure 4, using alerts generated by the Snort, INTRS should direct traffic of the packets to their desired destination (Linux interface). INTRS is in Perl script and uses iproute2 in Linux to redirect a packet to the desired interface. Iproute2 is a Linux control tool that controls the IP networking traffic. It consists of a set of commands and utilities that can control the traffic in different ways. The “ip route” command only manages and controls routing table; but iproute2 controls the route selection algorithm. In policy routing, different aspects of routing such as source, destination and type of service (TOS) are tended. Routing Policy Data Base (RPDB) is a set of rules like an access list and allows matching the source, destination, TOS and incoming interface in Linux. Fwmark is the tag for packet filtering that is used to mark packets with malicious behavior so that they are routed to honeypot.

4.3 The Protected Server and the Honeypot

The protected server is a service provider that is located in DMZ and installed to interact with those users whom connect to the system and send normal traffic to it. The honeypot is a decoy system that is designed to collect information from the attackers. To capture useful forensics from the attacker's behavior, a complete system is installed completely similar to the main server as a high-interaction honeypot. All of the features of this system such as its operating system, IP address, configuration and all the other parameters are simulated in such a way that the attacker cannot notice if he or she is lured in to a trap. In this implementation, the operating system of both the server and the honeypot are Linux fedora core 14 that comes with GPL.

5. Experimental Results

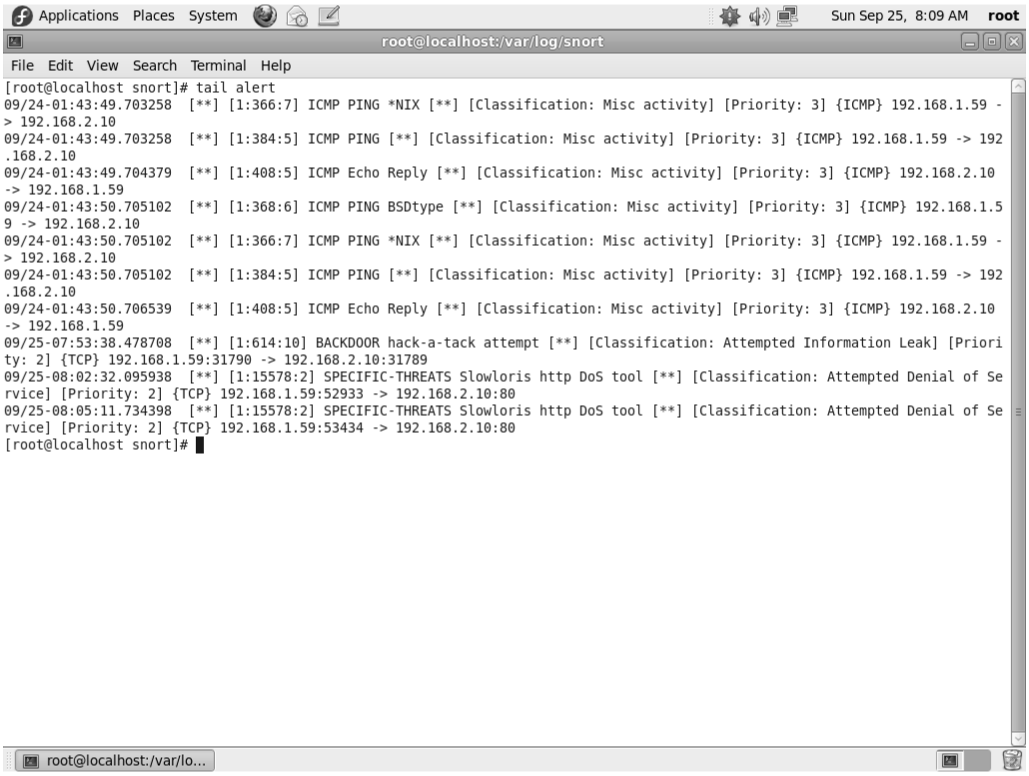
Once system is implemented, it is the time to test and evaluate it. In this section, results of the proposed architecture will be presented. As mentioned earlier, to secure DMZ, The proposed system controls input traffic, detects the malicious traffic from normal traffic and selects the appropriate destination depending on the traffic type and updates the firewall by sending new rules to it.

5.1 Attack Simulation and the System function

The first step for testing the proposed system is to simulate attacks and make attack scenarios in a way that the proposed system is working in a real environment. Many tools can be used for system simulation. In this experiment, Ping of death, Slowloris and Nemesis packet builder used for attack simulation. The Nemesis packet builder is not a tool for attacking networks but it can help us to craft packets and inject them into the network. It is a GPL software, downloaded free and installed on the attacker host using Linux operating system [18]. Nemesis has a command-line interface for its configuration. It can be used and installed in both Linux and Windows operating systems. Nemesis is one of the best and important tools to test different security devices such as NIDS, firewall and so on. Nemesis can be used as an automatic script-based engine capable of generating attack scenarios with a simple command-line interface. Nemesis can generate and inject different types of packets to the network. In this simulation, hack-a-tack Trojan horse is tested. Its packet is generated by Nemesis and Snort is known to the attack with the SID number 614. Figure 6 shows the alert generated by the Snort as well as different other types of the detected attacks that are simulated in this experiment

Ping triggers Snort's SID number 366 for ICMP request. Snort also records ICMP reply from decoy system to attacker with SID 408. This is a good evidence to prove that the designed system generates correct response for the attacks. Two other simulators are more efficient for testing the system. Slowloris is a GPL attacking tool written by Robert "RSnake" Hansen [19]. Using Slowloris, the attacker will be able to take down the web server of victim's system using minimal bandwidth. This attack will cause many problems for other ports and services of the victim machine. Slowloris tries to make a Denial of Service (DoS) attack by creating many connections to the victim web server and holding them open as long as possible. To reach this goal, the attacker opens connections to the target system and sends partial requests. Subsequent HTTP headers are sent, but the requests are never completed. When an affected system keeps much connections open, its connection buffer will be overflowed and it will not accept more connections. Therefore, the victim machine will not be able to operate because of overflow. Slowloris is a DoS tool which triggers Snort's SID number 15578.

Using simulated attacks to test the system, it detects the attacks using Snort and redirects them to the honeypot by INTRS, sending an update to firewall at the same time by INTRS that may also add suitable route to the routing table to redirect the intruder to the honeypot. The proposed system also changes the path of malicious traffic from the server to the decoy system as a response to the attack. Consequently, attacker is redirected to the honeypot where forensics is recorded. To make sure that the system is operating correctly, another IDS is installed on honeypot. Alerts produced by the IDS installed on the honeypot are compared versus those generated by the first IDS. In this way efficiency of the system is measured.


Applications Places System
Sun Sep 25, 8:09 AM root
root@localhost:/var/log/snort
File Edit View Search Terminal Help
root@localhost:/var/lo...

```
[root@localhost snort]# tail alert
09/24-01:43:49.703258  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.59 -
> 192.168.2.10
09/24-01:43:49.703258  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.59 -> 192
.168.2.10
09/24-01:43:49.704379  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.2.10
-> 192.168.1.59
09/24-01:43:50.705102  [**] [1:368:6] ICMP PING BSDtype [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.5
9 -> 192.168.2.10
09/24-01:43:50.705102  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.59 -
> 192.168.2.10
09/24-01:43:50.705102  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.59 -> 192
.168.2.10
09/24-01:43:50.706539  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.2.10
-> 192.168.1.59
09/25-07:53:38.478708  [**] [1:614:10] BACKDOOR hack-a-tack attempt [**] [Classification: Attempted Information Leak] [Priori
ty: 2] {TCP} 192.168.1.59:31790 -> 192.168.2.10:31789
09/25-08:02:32.095938  [**] [1:15578:2] SPECIFIC-THREATS Slowloris http DoS tool [**] [Classification: Attempted Denial of Se
rvice] [Priority: 2] {TCP} 192.168.1.59:52933 -> 192.168.2.10:80
09/25-08:05:11.734398  [**] [1:15578:2] SPECIFIC-THREATS Slowloris http DoS tool [**] [Classification: Attempted Denial of Se
rvice] [Priority: 2] {TCP} 192.168.1.59:53434 -> 192.168.2.10:80
[root@localhost snort]# █
```

Figure 6. A sampled alert file of Snort

5.2 Measuring Efficiency of the System to Secure DMZ

To check the system efficiency in securing DMZ, some factors are defined and their values are calculated in the experimental environment during its operation. Using these factors efficiency of the system is calculated. As presented in figure 7, the proposed model has two input packet generator systems, user host and attacker host. u1 and a1 are the traffic introduced by the users, so the total input traffic to the system is u1+a1.The proposed system detects some attacks and redirects them to the honeypot. But some of the traffic is directed to the server. Let u2 and a2 to be the legitimate and the malicious traffic directed to the server and u3 and a3 are redirected traffic of the legitimate and the malicious traffic directed to the honeypot. Measuring efficiency of the system, following indicators are defined:

True Positive Rate (TPR) = a3/(u3+a3) (1)
False Positive Rate (FPR) = a2/(u2+a2) (2)
ACCuracy (ACC) = (u2+a3)/(u1+a1) (3)

These indicators are measured in the next section to measure the system efficiency in different manners. The computed accuracy is the accuracy of the proposed INTRS. Since Snort is used as the IDS, the total accuracy of the proposed system can be calculated as a function of the Snort accuracy.

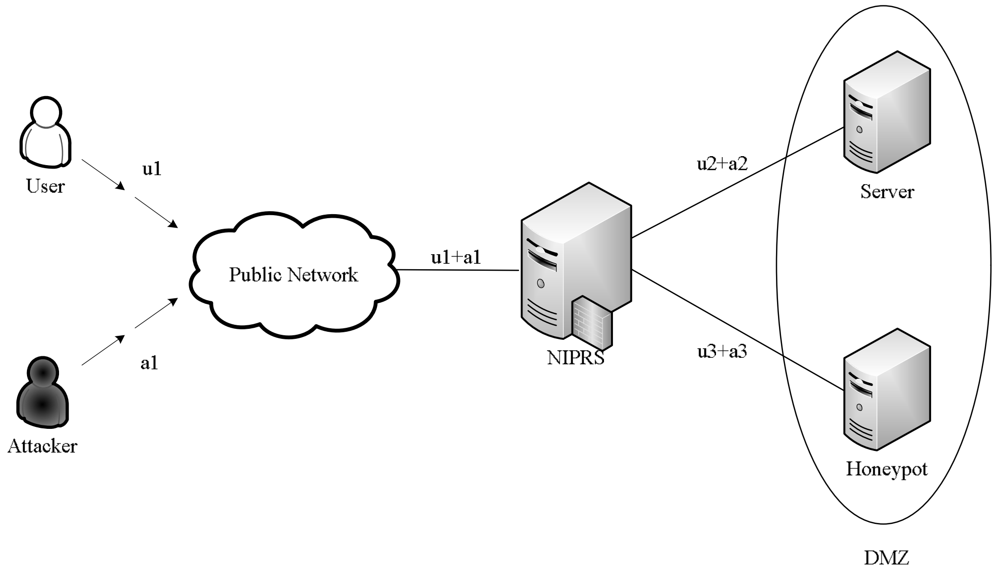


Figure 7: Measuring the traffic

5.3 Testing the Proposed System Using Attack Scenarios

To test the system, four attack scenarios are defined and the systemis tested using these scenarios. User and attacker machine produce the input traffics of the system. The attack scenarios are described in the following.

5.3.1 Scenario A

In this scenario, Ping of death is used as the attack. System is tested for 30 seconds, 60 seconds and 90 seconds attack durations. The input traffic is produced by the user and attacker machine and the parameters discussed in section 5-2 are presented in table 1.

Table 1. Measured parameters of simulation at scenario A

Time Range	u1	a1	u2	a2	u3	a3	TPR	FPR	ACC
30	420	40	420	2	0	38	1	0.47 %	99.57 %
60	895	80	895	2	0	78	1	0.22 %	99.79 %
90	1240	120	1240	2	0	118	1	0.16 %	99.85 %

5.3.2 Scenario B

Slowloris is another attacking tool to test the system. The second scenario is called scenario B. In this scenario, the system is similarly tested for 30 seconds, 60 seconds and 90 seconds attack durations. The input traffic is produced by the user and attacker machine and the parameters of efficiency are reported in table 2.

Table 2. Measured parameters of simulation at scenario B

Time Range	u1	a1	u2	a2	u3	a3	TPR	FPR	ACC
30	303	20	303	1	0	19	1	0.33 %	99.69 %
60	517	40	517	1	0	39	1	0.19 %	99.82 %
90	775	60	775	1	0	59	1	0.13 %	99.88 %

5.3.3 Scenario C

The next simulation is called scenario C and uses Nemesis as attacking simulator. In this scenario, system is tested for 30 seconds, 60 seconds and 90 seconds attack durations similar to the previous scenarios. The input traffic is produced by the user and attacker machine and the parameters of efficiency are presented in table 3.

Table 3. Measured parameters of simulation at scenario C

Time Range	u1	a1	u2	a2	u3	a3	TPR	FPR	ACC
30	295	5	295	1	0	4	1	0.34 %	99.67 %
60	420	8	420	1	0	7	1	0.24 %	99.77 %
90	610	9	610	1	0	8	1	0.16 %	99.84 %

5.3.4 Scenario D

The last scenario is designed to test the system with all types of attacks described in section 5-1. The system is again tested for 30 seconds, 60 seconds and 90 seconds attack durations. The user and attacker machines produce the appropriate traffic and the parameters discussed in section 5-2 are measured as table 4. This is a complete test scenario and the results can be used for analyzing the efficiency of the proposed system.

Table 4. Measured parameters of simulation at scenario D

Time Range	u1	a1	u2	a2	u3	a3	TPR	FPR	ACC
30	461	21	461	1	0	20	1	0.22 %	99.79 %
60	981	20	981	1	0	19	1	0.10 %	99.90 %
90	1164	20	1164	1	0	19	1	0.08 %	99.92 %

The reported results in tables 1 to 4 show that the accuracy of our proposed system to protect the DMZ is more than 99 percent. Legitimate traffic produced by user machine is not redirected to honeypot means that u3 is equal to zero. Thus, True Positive Rate (TPR) is equal to 1. Another important parameter that shows the system failure rate is False Positive Rate (FPR). FPR shows the rate of incorrect detections of the system. According to the results of simulations in tables 1 to 4, FPR is less than 0.50 percent that means the proposed system is up to our expectations and operates with a good efficiency.

The rate of attack avoiding to the DMZ is more than 99 percent, but the intrusion traffic can break into the main server placed in DMZ. The results in tables 1 to 4 show that system accuracy increases when flow of the malicious traffic to the server decreases. Nevertheless is should be reminded that the overall performance of the proposed system is totally dependent on the performance of the IDS.

6. Conclusions

The reported experiment explains the proposed architecture for avoiding cyberattacks to DMZ and capturing forensics from intruders using honeypots. Other features such as response to attacks and gaining trust of the attacker are also provided by the proposed architecture. Using this architecture, attacks can be detected using a NIDS. Consequently, using INTRS, changing the path of malicious traffic produced by attacker to the honeypot, useful forensics can be captured. According to the results of tables 1 to 4, only small part of the malicious traffic detected by the Snort can reach to the DMZ server, causing a small amount for FPR. Increasing INTRS efficiency will reduce the FPR. It is important to remember that the reported results are dependent on the accuracy of Snort, this is because Snort is used as the IDS at the entrance gateway of the proposed system where all the input packets are controlled. Consequently, the overall performance is calculated as a function of Snort's accuracy.

Another problem that may occur in the system is to have the path of legitimate traffic changed to the honeypot incorrectly. TPR shows how good the proposed system is performing its task. If TPR is equal to 1, no incorrect traffic path change is happened. But if it is less than 1, this means that system redirects legitimate user traffic to the honeypot. Tables 1 to 4 show that TPR is equal to 100 percent, because u3 is always equal to zero that means the system never changes the legitimate traffic to the honeypot system.

The reported results in section 5-3 show that the designed system can make a secure DMZ by detecting the attacks to the DMZ servers and redirecting them to a decoy system. The proposed architecture is a solution for the problem of attack redirection to honeypots. The proposed system complies with expectations and diverts attacks to the server with an acceptable accuracy. This system redirects attacks to the honeypot where forensics will be recorded. Physical separation of the DMZ from the honeypot zone in the proposed architecture enables the proposed system to prevent attacks from compromised honeypots on servers in DMZ. This feature is a major strength for the proposed system once compared with honeynets.

7. Future Works

The proposed architecture detects rule-based attacks since Snort is used as for the IDS. However, new attacks that are not known and some call them zero-day attacks always threat the systems. In the designed model, if anomaly-based IDS is used, the system will be able to detect new methods of attacks against the DMZ. Thus one of the most important issues for the future works can be the detection of zero-day attacks using anomaly-based IDS.

Another issue that can be mentioned is reducing FPR of the system. This problem can be solved by improving INTRS. Optimizing INTRS and its relation with NIDS is also left for the future work.

8. References

[1] Bauer, M. (2001), Designing and Using DMZ Networks to Protect Internet Servers. Linux Journal. Vol. 17(83).

[2] Spitzner, L. (2002), Honeypots: Tracking Hackers. Addison Wesley Pearson Education, Boston, MA.

[3] Maheswari, V. and P.E. Sankaranarayanan (2007), Honeypots: Deployment and Data Forensic Analysis. Internal Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), pp. 129-131.

[4] Alata, E., I. Alberdi, V. Nicomette and M. Kaaniche (2008), Internet Attacks Monitoring with Dynamic Connection Redirection Mechanisms. Journal in Computer Virology. Vol. 4: pp. 127-136.

[5] Kim, I. and M. Kim (2007), The Decoy Port: redirecting hackers to honeypots. Proceedings of the 1st international conference on Network-based information systems 2007 (NBiS'07).

[6] Yang, Y. and J. Mi (2010), Design and Implementation of Distributed Intrusion Detection System Based on Honeypot. 2nd International Conference on Computer Engineering and Technology (ICCET), Vol. 6: pp. 260-263.

[7] http://en.wikipedia.org/wiki/Firewall_(computing) as visited on 25 November 2011.

[8] Thakar, U., N. Dagdee, and S. Varma (2010), Pattern Analysis and Signature Extraction for Intrusion Attacks on Web Services. International Journal of Network Security & its Applications (IJNSA), Vol. 2(3).

[9] Kabiri, P. and A.A. Ghorbani (2005), Research on Intrusion Detection and Response: A Survey. International Journal of Network Security, Vol. 1(2): pp. 84-102.

[10] Rikhtechi, L. and A.R. Roozbahani (2010), Creating a Standard Platform for All Intrusion Detection/Prevention Systems. 2nd International Conference on Computer Modeling and Simulation (ICCMS), Vol. 3: pp. 41-44.

[11] Garcia-Teodoro, P., J. Diaz-Verdejo and G. Macia-Fernandez (2009), Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges. Computers and Security Journal, Vol. 28: pp. 18-28.

[12] Hu, J., X. Yu, D. Qiu and H. H. Chen (2009), A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. IEEE Network, Vol. 23: pp. 42-47.

[13] Caswell, B., J. Beale, and A.R. Baker (2007), Snort IDS and IPS Toolkit. Syngress Publishing .

[14] Zhou, Z., C. Zhongwen, Z. Tiecheng and G. Xiaohui (2010), The study on network intrusion detection system of Snort. 2nd International Conference on Networking and Digital Society (ICNDS), Vol. 2: pp. 194-196.

[15] Mokube, L. and M. Adams (2007), Honeypots: Concepts, Approaches and Challenges. ACM-SE 45 Proceedings of the 45th annual southeast regional conference, New York, NY, USA, pp. 321-326.

[16] Briffaut, J., J.F. Lalande, and C. Toinard (2009), Security and Results of a Large-Scale High-Interaction Honeypot. Journal of Computers, Vol. 4(5): pp. 395-404.

[17] Know Your Enemy: Honeynets. http://www.honeynet.org/papers/honeynet as visited on 9 February 2012.

[18] Nemesis, http://nemesis.sourceforge.net as visited on 9 February 2012.

[19] Slowloris, http://ha.ckers.org/slowloris as visited on 9 February 2012.

Journal of

Advances in Computer Research

Instructions for Authors

Aims and Scope

The ***Journal of Advances in Computer Research*** is published quarterly by the Islamic Azad University-Sari Branch. Editorial board of the journal welcome contributors from all around the world to provides a forum for publication of significant advancements and developments in all areas of computer and information technology.

All submitted manuscripts, including conference papers, will be peer reviewed by qualified scholars assigned by the editorial board. The journal only accepts and publishes the original and high quality papers relevance to the journal's scope.

Manuscript Preparation

Manuscript should be in English. Submission of a manuscript indicates that it has neither been published nor been submitted for publication elsewhere and is result of research performed by the author(s). Appearance in a conference proceeding is not considered as a prior publication. The contributors are expected to consider the following notes:

- Manuscripts should be typewritten in a font size of at least 12 points, on one side of A4 paper, double-spaced, with adequate margins.
- A list of three to five keywords should be included at the foot of the abstract.
- References should be numbered in brackets and appeared in sequence through the text. List of references should be provided at the end of the paper.
- Figure captions are to be indicated under the illustrations. They should sufficiently explain the figures.

Manuscript Submission

The manuscript can be submitted electronically to the journal via its email at jacr@iausari.ac.ir or the journal web site: http://www.jacr.iausari.ac.ir.

Page Charges

There is no page charge for publication in Journal of Advances in Computer Research.

- The Journal of Advances in Computer Research has been assigned the ranking of " scientific- research journal" regarding the 74th meeting of the verification commission for Islamic Azad University journals in 13/3/2011
- This journal is also indexed in Islamic World Science Citation Center (ISC) (www.isc.gov.ir) and Scientific Information Database (SID) (www.sid.ir).

Journal of

Advances in Computer Research

Content 7

Volume 3, Number 1
February 2012

Content 6

Volume 2, Number 4
November 2011

Content 5

Volume 2, Number 3
August 2011

Content 4

Volume 2, Number 2
May 2011

Volume 3, Number 1, February 2012