Improving Exposure of Intrusion Deception System through Implementation of Hybrid Honeypot

Masood Mansoori[1], Omar Zakaria[2], and Abdullah Gani[1]
[1]Faculty of Computer Science and Information Technology, University of Malaya, Malaysia
[2]Faculty of Defence Science and Technology National Defence University of Malaysia, Malaysia

Abstract: *This paper presents a new design hybrid honeypot to improve the exposure aspect of intrusion deception systems and in particular, research server honeypots. A major attribute in the design of a server honeypot is its passiveness, which allows the honeypot to expose its services and passively wait to be attacked. Although passiveness of a server honeypot simplifies the analysis process by classifying traffics as malicious, however it also lessens its ability to lure attackers through exposure of vulnerable service. As a result it captures smaller amount of data on attacks for analysis. Client honeypot designs, on the other hand, contain modules that actively interact with outside networks, expose vulnerabilities in client side software, and identify malicious content, hosted on webservers. The proposed hybrid system integrates active module concept of a client honeypot into a server honeypot. The active module interacts with webservers utilising a custom crawler and browser, publicises the honeypot's IP address and therefore improves exposure of server honeypot's vulnerable services. The findings presented in this paper show that interaction with webservers improves exposure, and results in significantly higher number of attacks, which in turn, increases the probability of discovering new threats. The findings also characterise most attacks to be worm based and directed at windows based hosts and services.*

Keywords: *IDS, server honeypot, client honeypot, hybrid honeypot.*

Received April 27, 2010; accepted October 24, 2010

1. Introduction

Honeypots are simple and cost effective solutions that provide direct and indirect value to an organisation's information security [15]. The direct value of honeypots resides in their detection capabilities aside from their simplicity in concept, installation, and management. Its indirect value is the results of analysis performed on captured data through the deployment of thousands of honeypots around the world. The results of the analysis have identified security threats to production servers and hosts of many organisations; the tools and tactics used by attackers. This information, in turn, can help in the development of proper tools and techniques to counter the identified threats and the attack tools used.

Honeypots are differentiated based on the level of interaction with the attackers (i.e., high, medium, or low interaction), and the purpose of deployment (i.e., production or research). Research honeypots are the tools used by researchers to identify attack patterns, the tools and techniques used and have been proven to be effective in such areas [18]. However, like other technologies, there are shortcomings in exploiting the full potentials of research honeypots.

Current research on server honeypot systems is focused on increasing the deception capabilities of server honeypots to avoid detection, and subsequently, to capture extensive data on the attackers. Other properties of server honeypots such as exposure of vulnerable services, play a vital role in a research honeypot's objective to capture a larger number of attacks for analysis. The objective of this research is to determine whether active interaction with webservers, currently, implemented in client honeypots, can be utilised as a means of increasing exposure of server honeypots. In this research, a hybrid honeypot system to improve the exposure aspect of intrusion deception systems, specifically research server honeypots is proposed. The proposed system integrates active modules of client honeypots into server honeypots. The active modules interact with webservers, publicising the honeypot's IP address, and increasing exposure of server honeypot's services. Additional details on intrusion deception systems are given in the following section.

2. Intrusion Deception System

Traditionally, information security has been defined and implemented based on the concept of protection, detection, and reaction [1]. Protection implies defending the information against unauthorised access. Detection is an on-going process of monitoring activities for any failure in the defence structure and discovering intrusion, using intrusion detection systems. Reaction involves the actions taken to recover from a failure in the protection and detection systems. Modern approaches in information security employ deception as an added layer to create a more robust defence strategy. Deception is frequently used by hackers to conceal their identities and activities [19]. For instance, IP spoofing uses deception techniques to conceal the identity of the real attacker behind an authentic host. Concealment of identity and activities allows attackers to bypass detection tools and increase the chances of a successful compromise. Deception techniques are used in protection against attacks in the form of concealment of valuable information and hosts, by placing decoy systems, which are targeted by hackers. This decoy system is an intrusion deception system. Honeypots and honeynets are the main tools in intrusion deception systems which utilise deception techniques in the form of decoy systems, services, and false information. Decoy systems divert attention away from real targets and offer appealing environments for attackers to work without being aware that their actions are being recorded. Production honeypots are decoy systems that are responsible for diverting attention away from high-value production servers and hosts [18]. This is achieved by placing decoy systems that mirror production servers. Deployment of a large number of decoy systems within a production network can significantly increase the time required for scanning the targets, thus, providing additional time for administrators to detect attacks. Production honeypots can also be used to mitigate DOS attacks and slow down network scanners and propagation of worms, by placing sticky honeypots such as Labrea Tarpit, among production hosts.

Data gathering on attacks and monitoring attack behaviour, pattern and tools, is typically achieved through deployment of research honeypots in large numbers within the networks of universities, corporations and research organisations [8]. Deploying large number of honeypots in different geographical locations maximises exposure of honeypots to attacks and increases the probability of detecting new attacks by capturing large amount of data. Large amount of data collected from different honeypots located in different regions, also provides an overview of the types and frequency of cyber attacks in different regions around the globe. Leurre.com, Honeynet Alliance, and Brazilian Honeynet project, are examples of such research organisations dedicated to monitoring internet threats. Research server honeypots implemented by such research groups, involve the use of single or distributed networks of server honeypots. Server honeypots are passive decoy systems waiting to be attacked. The passiveness of such honeypots is a major factor in the simplicity of honeypot deployment and analysis of captured data, given that no legitimate traffic exists, and hence, any traffic and activity within the honeypot is considered to be malicious. However, lack of exposure of honeypot services to attackers is a direct outcome of the passiveness property of server honeypots. A detailed explanation of server and client honeypots, including the properties, advantages and drawbacks is provided in the following section.

3. Server and Client Honeypots

Traditional honeypots were designed and deployed to protect production environments by mirroring production servers and services. These honeypots were commonly called server honeypots. A wide range of systems to combat and capture different attacks such as DDos, Spam and worm mitigation were designed and introduced. However, the number of online threats, targeted at client applications' vulnerabilities, has increased significantly in the past few years. The shift from server side to client side attacks is due to the increased popularity of web browsers for accessing online resources and the improved security of servers [10]. As attack behaviour changed and shifted to client side applications, a new classification of honeypots to identify attacks on client applications, discover vulnerabilities of client web browser, and detect malicious webservers, was introduced. The new class of honeypots focuses on client attacks, hence, the name "client honeypots" or "Honeyclients", was adopted. These two generalisations of honeypots differ in purpose of deployment, as well as the approaches used to identify and capture attacks.

3.1. Server Honeypots

Server honeypots operate by exposing vulnerable services and waiting for attackers to exploit the services [5, 4, 13] as shown in Figure 1. In the case of production honeypots aimed at protecting other hosts by engaging attackers with non production services and resources, passiveness is not a setback as fewer attacks on a honeypot mean less risk to the operational hosts. However, research honeypots merely depend on the captured data to discover new vulnerabilities and threats. A passive server honeypot with no mechanism to expose its presence is less likely to be attacked than a honeypot with an exposed IP address. A study by Bloomfield *et al.* [3] shows that honeypots deployed in large corporations' networks whose IP addresses are more public, captured a higher number of attacks than honeypots in small enterprises.

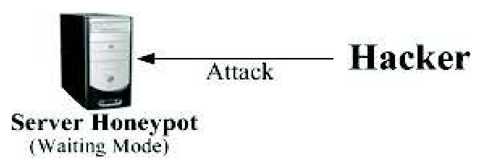

Figure 1. Server honeypot passively waits to be attacked.

Increasing exposure is applied in spam honeypot systems where several email addresses redirecting spam emails to a spam honeypot are manually propagated across the internet in chat rooms, forums and bulletins [14]. The propagation of email addresses increases the probability of email addresses being located by spam Bots and used as a target for spam emails. The passiveness of server honeypots, however, remains a property of server honeypots, but is addressed in client honeypots.

3.2. Client Honeypots

Client honeypots take an active approach to identify attacks on the systems. Client honeypots expose the client side services and actively crawl into web servers, visit websites, and monitor activities to determine if a webserver is malicious [17] as shown in Figure 2. This is done using client browsers or crawlers to visit websites placed on different webservers while every process, file read/write and registry entry, within the client honeypot is monitored. If any changes occur following each visit, the website is classified as malicious. Monitoring the traffic and examining the data for potential malwares accessed by a honeyclient, is achieved by utilising antivirus software or intrusion detection signatures. Capture-HPC is an example of a high interaction client honeypot. Table 1 summarises the key properties of client and server honeypots.

Active interaction and classification of attacks are the main properties of all client honeypots. A system, which takes advantage of active property of client honeypots, straightforward attack classification, and exposure of vulnerable services of server honeypots, is able to significantly improve the efficiency of research server honeypots in capturing attacks.

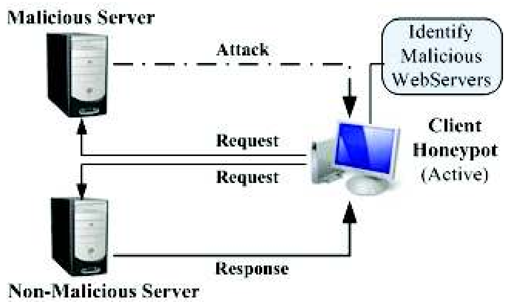

Figure 2. Honeyclients actively identify malicious webservers.

Table 1. Comparison between server and client honeypots.

Server Honeypots	Client Honeypots
Provide services for clients to use	Do not provide any services to others
Lure attackers into attacking exposed services	Do not lure attackers as no service is provided
Passively wait for an attacker	Actively crawl the web to identify malicious web servers
Very low or no false alarms	Generate false alarms
Every access and traffic is considered a threat	Has to decide which traffic is malicious

4. Hybrid Honeypot System

We propose a hybrid honeypot design to offset the weaknesses of client and server honeypots. The proposed hybrid honeypot has been designed to manipulate and combine the advantages of the client and the server honeypots, which will help to enhance exposure by active interaction with potential attackers.

Active interaction is an effective mechanism for finding malicious web servers across the internet and is used by all client honeypots [13]. The proposed honeypot system integrates the active interaction of client honeypots into server honeypots to form a hybrid honeypot system. The proposed hybrid honeypot system should have the following properties:

- *Exposed Vulnerable Services:* Exposure of vulnerable services, either simulated or real, is the basic feature of server honeypots to lure attackers toward the honeypot system. Production servers and hosts are constantly probed for open ports and vulnerable services to be exploited. A honeypot system that does not have any exposed services or open ports, will not be able to attract attackers and will simply be bypassed. Our proposed hybrid honeypot system utilises decoy and simulated vulnerable services to lure attackers.
- *Interaction:* The proposed system should take advantage of the active property of client honeypots to overcome the passiveness of server honeypots. The active component within the proposed system interacts with outside networks to advertise the presence of vulnerable services and generate traffic to increase deception.
- *Generating Small Set of Data for Analysis:* Unlike other technologies such as IDS, which produces false alarms and collects a very large amount of data for analysis, server honeypots produce no false alarms and capture comparatively very small amount of data. This is because server honeypots are not part of the production network, and conceptually, are not supposed to have any traffic and activity. Therefore, any activity observed in the honeypot system and the ports can be deemed to be malicious. Our proposed hybrid system can monitor and log activities and traffic on exposed services and open ports of a server honeypot. The active module, which is integrated into the proposed system, utilises minimal resources in terms of port usage to perform interaction, thus, activities on other ports can easily be monitored without creating unnecessary traffic.
- *Simplicity:* The module responsible for interaction were designed to be simple to install and to be compatible with different types of honeypot, and different operating systems.

The proposed hybrid honeypot addresses the passiveness of server honeypots, and specifically, research honeypots by integrating the active modules, which interact with outside networks as shown in Figure 3. The aim is to incorporate the concept of publicising a honeypot's presence in spam honeypots into server honeypots by advertising the honeypot's IP address through interaction. Interaction with webservers and publicising honeypot's IP address, however, do not expose the existence of a decoy system but increase the exposure of its services. Moreover, an active honeypot is less likely to resemble a conventional passive honeypot, hence, the deception is more likely to increase.

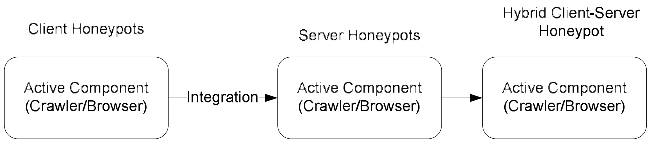

Figure 3. Hybrid honeypot integrates modules of client honeypots into server honeypots.

The hybrid system utilises a customised web browser and a crawler to perform interaction with suspected malicious webservers. The crawler initiates connection with a webserver, downloads the source codes of a website and extracts the URLs embedded within the source code. Extracted URLs are added to a list of websites to be visited by the browser. The links are then browsed by the browser in a timely fashion, as set by an attribute within the browser code as shown in Figure 4. It must be noted that the purpose of this crawler/browser module is not to be exploited like client honeypots, but to act as an active module to interact with webservers. Meanwhile, all services and ports are monitored for any activity through a server honeypot.

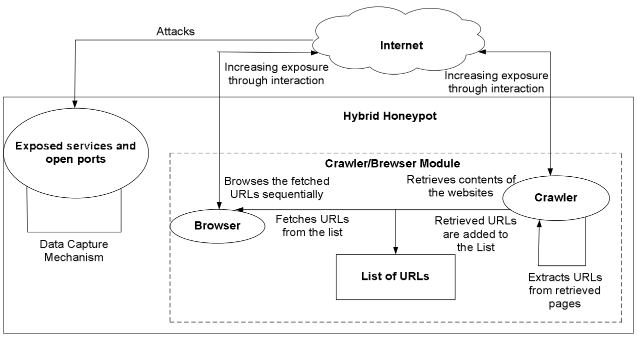

Figure 4. Block diagram of the functions of the proposed system.

Traffic generation is a direct outcome of interaction performed by the active modules of the proposed system. Lack of legitimate traffic is one of the basic attributes of honeypots, which deem any traffic and activity in and out of honeypot as malicious. Lack of traffic, on the other hand contributes to the detection of honeypots. Mukkamala *et al*. [9] states that the only true way to detect an ideal honeypot at the network layer is to monitor the local traffic. The proposed system overcomes this drawback by generating simulated traffic which is achieved by interaction with the webservers. If the proposed system is deployed in a honeypot, the browser generates traffic by interacting with webservers. This causes the source IP address of the honeypot to be available in the HTTP packet headers being transmitted through the network. Anyone sniffing the traffic using the packet sniffing utilities would notice the information exchange and activity originating from or toward the honeypot. This creates an illusion that the honeypot is an operational host/server.

Traffic generation is particularly useful if it is done in conjunction with high interaction honeypots. While this system can help to reduce the possibility of detection of low and medium interaction honeypots, since their interaction level is a major detection factor, detection through a lack of traffic is the least concern on such honeypots. On a high interaction honeypot, however, services are not simulated and an actual operating system is provided. Therefore, the lack of traffic is one of the major factors that can blow the cover of a honeypot. This system can increase the deception capabilities of honeypots and overcome a major disadvantage.

The honeypot system used in our implementation is HoneyBot, which is a Windows-based medium interaction honeypot capable of simulating vulnerable services and monitoring ports that are used by those services. HoneyBot provides the functionalities to: capture the payload of connections; download the binaries of several malwares; and offer enhanced interaction for some services such as SSH, FTP and Telnet. The latter services allow the attacker to login into false accounts while every username and password is logged. HoneyBot is used in the implementation of several honeynet groups such as the Taiwanese Honeynet Chapter. The HoneyBot system used in this research can distinguish between connection attempts initiated by interaction of active module and connection attempts initiated from outside toward the used ports. In the former case, the activities are not logged, whereas in the latter case, all activities and connection attempts are logged.

The crawler/browser module was developed using Java programming language as it is compatible with a wide range of operating systems. Both HoneyBot and crawler/browser modules were installed on a newly installed Windows XP Professional system containing the latest updates, and deployed to perform data capture process.

5. Data Capture

Data capture is the process of monitoring and logging every activity and data going in and out of a honeypot. The data capture function is generally embedded within the honeypot software in low and medium interaction honeypots, while high interaction honeypots utilise data capture tools such as Sebek, intrusion detection system, and firewall logs to monitor activities in a honeypot. The HoneyBot system utilises a built-in data capture mechanism, which is capable of capturing connection attempts, connection payloads, binary codes of worms and attempted usernames and passwords. The metric for comparison of increased exposure of a honeypot system is based on the number of captured attacks. In order to determine the effectiveness of an active hybrid honeypot system compared to a regular server based honeypot in capturing attacks, a comparison was made based on the number of logged attacks in two separate phases of data capture:

- Pre-deployment phase (phase 1).
- Post deployment phase (phase 2).

5.1. Pre-Deployment Phase (Phase 1)

Phase 1 (pre-deployment phase) involves the deployment of HoneyBot without any interaction with webservers, representing a regular server honeypot. In order to determine the number of attacks on a passive server based honeypot system, HoneyBot was deployed using a public IP address for a period of one week, during which, the honeypot was never taken offline. In this phase of data collection, 1311 sockets, corresponding to widely-used protocols and applications, were created and the activities on each port were monitored and logged. Each log file, belonging to each day, was automatically saved in the local hard disk as Comma Separated Values or .csv files. A main characteristic of server honeypots is the lack of any legitimate activity and traffic within the honeypot system. Therefore, to mimic a regular passive server honeypot, no user activities were performed, no websites were visited and no outbound connections were initiated on the honeypot. This mimics a standard server based honeypot where legitimate user activity and traffic must not exist.

5.2. Post-Deployment Phase (Phase 2)

The second phase (post-deployment phase) of data capture involved interaction with webservers through surfing of websites by the crawler and browser module. Upon completion of the first phase of data collection, a new unfiltered public IP was obtained, and an initial 1000 URLs of suspicious websites, mostly comprising of Warez and adult websites, were loaded into the browser. The system was then left running for a period of one week and additional websites were occasionally crawled. Similar to the first phase of data collection, 1311 sockets were created and monitored. The reason for obtaining a new IP address in phase 2 was to lower the probability of higher attack rates due to the longer duration for which the honeypot had been online, and to obtain a more accurate result.

6. Results

Data captured through the deployment of a regular server honeypot and a hybrid honeypot system over different durations were respectively analysed. Attacks were analysed based on the number of attacks captured each week to determine the effectiveness of the hybrid honeypot system as compared to a regular server based honeypot. Analyses were also made to determine the origins of the attacks as well as the behaviour of the attackers, in both phases.

6.1. Attack Statistics

Based on the analysis of the attacks, the first attacks took place less than 15 minutes after going online and were repeated approximately every 2 minutes and 20 seconds during phase 1. Overall, the rate of attacks has been quite consistent throughout the first phase, with the exception of 17 May 2009 which was a Sunday. This could be due to the fact that fewer infected hosts, particularly corporate hosts, were online during the weekend. Compared to the average number of attacks on other days of the week, Sunday represents a 30% drop in the number of attacks given in Table 2.

Table 2. Summary of attacks during phase 1.

Phase 1	
Date	**Number of Attacks**
11/5/2009 – Monday	718
12/5/2009 – Tuesday	725
13/5/2009 – Wednesday	648
14/5/2009 – Thursday	666
15/5/2009 – Friday	656
16/5/2009 – Saturday	660
17/5/2009 – Sunday	475
Total Number of Attacks	4584
Average Attacks Per Day	654

The HoneyBot captured a total of 7568 attacks during phase 2, which is an average of 1 attack every 1 minute and 33 seconds. The total number represents a significant increase of 65% in attack ratio. As in the first phase, attacks logged for Sunday, 24 May 2009 was relatively lower, by 16% given in Table 3.

Table 3. Summary of attacks during phase 2.

Phase 2	
Date	**Number of Attacks**
18/5/2009 – Monday	1118
19/5/2009 – Tuesday	1175
20/5/2009 – Wednesday	987
21/5/2009 – Thursday	1011
22/5/2009 – Friday	1177
23/5/2009 – Saturday	1171
24/5/2009 – Sunday	929
Total Number of Attacks	7568
Average Attacks Per Day	1081

6.2. Origin of Attacks

Table 4 shows the top countries where the attacks originated. The findings from a single deployed honeypot in this implementation show that China, Russia, Japan, and the United States top the list of countries from where the attacks originated. These countries are also among the top 10 countries with the highest number of internet users.

Table 4. Top attacking countries.

Country	Country Code	Percent
Russian Federation	RU	11.48%
China	CN	9.30%
Japan	JP	7.90%
United States	US	7.48%

Identifying the most persistent IP addresses can help administrators to setup rules and share their information with other organisations so that preventive measures can be taken to avoid attacks from these IP addresses. Another measure would be to inform the relevant authorities such as relevant ISPs about the malicious activities originating from their domain, so that appropriate actions can be taken. A list of the most persistent IP addresses of the attackers, originating country of the attacks, and the number of attacks attributed to each IP address, is shown in Table 5.

Table 5. Most persistent IP addresses.

IP Address	Country	Number of Attacks
125.65.112.204	China	159
220.194.138.156	China	95
61.160.217.10	China	48
121.14.148.200	China	43
202.97.184.67	China	38

6.3. Attack Behaviour

HoneyBot is considered a medium interaction honeypot, and as such, it does not provide full interaction for every protocol. This makes it difficult to differentiate between Worms and human-based attacks. Based on the similarity of payloads and the vast number of connections, however, most of the attacks were basic scanning, and worm propagation. The SYN/FIN packets made up the highest number of received packets captured by HoneyBot. On three occasions, however, human-based activities were detected on port 21 (FTP), and the attempted usernames and passwords were logged.

Captured usernames targeted by attackers included "administrator" and "guest" accounts, and the combination of passwords that were tried were among the most common passwords [11]. A research by Ramsbrock *et al.* [12], based on SSH server honeypot, identified these passwords as the top attempted passwords used in dictionary attacks based on common passwords. Generally, attackers were not very keen to gain access as no dictionary-based or brute-force attack was used and each attacker usually tested only a few passwords. The passwords, in descending order of number of usage, include: abc123, password, passwd, 123456, none and guest123.

Port 25 logged several attempts at relaying email addresses, most probably attributed to Bots scanning for open relays. This port is constantly scanned by spammers' automated software to find open relays to send spam. Based on the captured content and the subject of the emails, a test email address is sent to the spammer's email address containing the IP address of the target. If it succeeds, the IP address of the target is added to the list of open relays. Figure 5 shows an example of relaying attempt from 123.205.233.208 directed at the honeypot (121.121.5.77) on port 25. A test email from cool.boy@ms.hinet.net is sent to bibio@gmail.com with the IP address of the honeypot as the subject of the email.

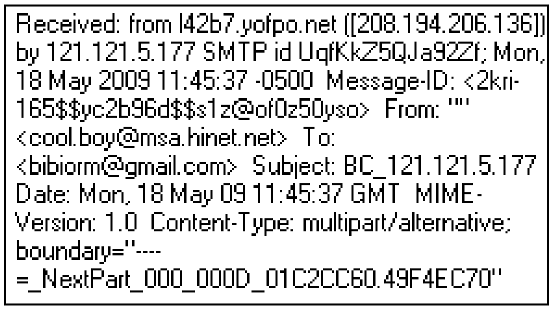
Received: from l42b7.yofpo.net ([208.194.206.136]) by 121.121.5.177 SMTP id UqfKkZ5QJa9ZZf; Mon, 18 May 2009 11:45:37 -0500 Message-ID: <2kri-165$$yc2b96d$$s1z@of0z50yso> From: "" <cool.boy@msa.hinet.net> To: <bibiorm@gmail.com> Subject: BC_121.121.5.177 Date: Mon, 18 May 09 11:45:37 GMT MIME-Version: 1.0 Content-Type: multipart/alternative; boundary="----=_NextPart_000_000D_01C2CC60.49F4EC70"

Figure 5. An attempt to relay email captured by HoneyBot.

The test email is what makes spam honeypots such as HoneySpam [2] not practical as the HoneySpam does not allow forwarding of spam emails, and unless the test email is not delivered, no spam is forwarded to the honeypot. Jackpot honeypot, an SMTP relay honeypot written in Java programming language, is capable of forwarding the test email sent by an attacker. The rest of the spam is not delivered but saved in a single database shared with several honeypots.

Overall, attacks on port 445 (SMB) contributed to 66% of all attacks against the HoneyBot followed by port 1433 (Microsoft SQL Server) contributing 11%, port 135 (DCE-RPC) contributing 7%, and port 139 (NetBIOS) contributing 5%. Table 6 shows the top most attacked ports and the total number of attacks on each port, during the two data capture periods.

Table 6. Top 10 most attacked ports.

Port	Total Attacks	Percentage
445	7613	66 %
1433	1286	11 %
135	846	7 %
139	628	5 %
1080	332	3 %
2967	303	3 %
137	176	2 %
22	153	1 %
4899	148	1 %
110	75	1 %

7. Discussion

Analysis of the captured data produced some interesting results as follow:

- *Improved Exposure and Increased Number of Attacks:* Findings demonstrate that browsing websites and interaction with webservers, can significantly increase exposure and the likelihood of attacks on server based honeypot services. During deployment of a hybrid honeypot system in phase 2 of data capture, 65% more attacks were captured when compared to the deployment of a single passive server honeypot in phase 1 of data capture as shown in Figure 6.

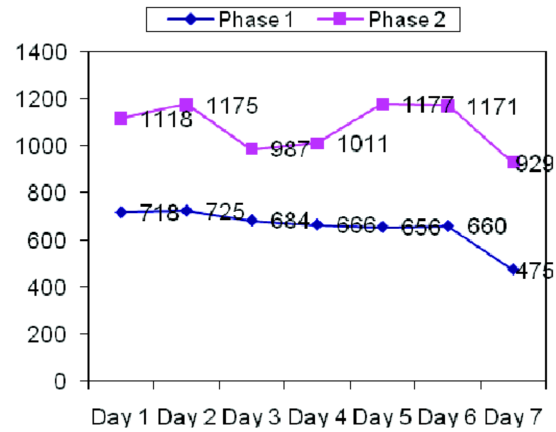

Figure 6. Graph representing the number of attacks in the two phases of data capture.

- *Identification of the Origins of Attacks:* Attack behaviour, connection payloads, and number of connections from a wide range of hosts, and a few persistent IP addresses, show that most attacks were initiated by worms scanning random hosts, and attempting to spread by infecting vulnerable services of target hosts.
- *Identification of Most Targeted Operating Systems:* Based on the captured data in both phase 1 and phase 2, Windows native services recorded the highest number of connection attempts, and contributed to 84% of total attacks against the honeypot. It is, therefore, advisable for organisations running Windows operating systems to be vigilant in monitoring these services and ports for illegal activities, and to institute protective measures for the services using firewalls, intrusion prevention systems or patches of the latest updates from vendors.
- *Placement of Honeypots:* Analysis of the captured data by honeypots helps system administrators to get a clear understanding of the risk associated with every running service and open port in their system. Honeypots deployed behind a firewall within an organisation can detect infected hosts that are used to propagate worms, and help with insider threats. While most attacks are blocked by firewall, open ports that might have been missed out to be filtered by firewall can be identified, and illegitimate connection attempts, either internal or external, can be detected [6]. On the other hand, a honeypot deployed in front of a firewall is constantly being probed and attacked, and gives users an indication of the severity and the number of attacks against the ports and services.

8. Conclusions

This study focuses on the limitations of research server honeypots that in use. It is aimed at enhancing the design of the honeypot to address the limitations. The main outcomes of the study are:

- The current approach used by server honeypots to attract attackers is passive in nature because server honeypots do not initiate connection; have no interaction; and take no measure to publicise their vulnerable services to lure attackers.
- Client honeypots contain active modules (i.e., Crawler, Browser), which interact actively with webservers, examine websites, and classify them based on the maliciousness of the content.
- The significance of this research yields is evident with the introduction of a hybrid client server honeypot. Previous definition of hybrid honeypots involved integration of honeypots with different interaction capabilities in the same design; utilising low and medium interaction honeypots to log less significant attacks, while more sophisticated attacks were directed to high interaction honeypots. In this paper, a hybrid honeypot system that bridges the gap between honeypots with entirely different functionalities was introduced. It consists of active components of the honeyclients, and the exposed services of server honeypots. This design was intended to enhance the data capture capabilities of server based honeypots.
- Analysis of the captured data from a regular server honeypot as well as the proposed hybrid honeypot, demonstrates the effectiveness of active interaction of hybrid honeypot system in improving exposure of server side services. A considerably higher number of attacks was noticed immediately after the deployment of the hybrid honeypot system.

Active interaction has shown an effective mechanism of client honeypots to detect malicious webservers, and also an excellent technique to achieve higher attack rates on research server honeypots.

Acknowledgment

We would like to express our appreciation to all those who helped us to understand the importance of knowledge and showed us the best way to gain it.

References

[1] Anderson P., "Deception: A Healthy Part of Any Defense in-Depth Strategy," available at: http://www.sans.org/reading_room/whitepapers/policyissues, last visited 2001.

[2] Andreolini M., Bulgarelli A., Colajanni M., and Mazzoni F., "Honeyspam: Honeypots Fighting Spam at the Source," *in Proceedings of the International Workshop on Steps to Reducing Unwanted Traffic on the Internet*, USA, pp. 77-83, 2005.

[3] Bloomfield R., Gashi I., Povyakalo A., and Stankovic V., "Comparison of Empirical Data from Two Honeynets and a Distributed Honeypot Network," *in Proceedings of 19th International Symposium on Software Reliability Engineering*, USA, pp. 219-228, 2008.

[4] Hecker C., Nance K., and Hay B., "Dynamic Honeypot Construction," *in Proceedings of the 10th Colloquium for Information Systems Security Education*, US, pp. 95-102, 2006.

[5] Jianga X., Xua D., and Wang Y., "Collapsar: AVM-Based Honeyfarm and Reverse Honeyfarm Architecture for Network Attack Capture and Detention," *Journal of Parallel and Distributed Computing*, vol. 66, no. 9, pp. 1165-1180, 2006.

[6] Maheswari V. and Sankaranarayanan P., "Honeypots: Deployment and Data Forensic Analysis," *in Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, USA, vol. 4, pp. 129-131, 2007.

[7] McGrew R. and Vaughn R., "Experiences with Honeypot Systems: Development, Deployment, and Analysis," *in Proceedings of the 39th Hawaii International Conference on System Sciences*, USA, vol. 9, pp. 220a, 2006.

[8] Mokube I. and Adams M., "Honeypots: Concepts, Approaches and Challenges," *in Proceedings of the 45th Annual Southeast Regional Conference*, USA, pp. 321-326, 2007.

[9] Mukkamala S., Yendrapalli K., Basnet R., Shankarapani K., and Sung H., "Detection of Virtual Environments and Low Interaction Honeypots," *in Proceedings of 8th IEEE Information Assurance Workshop*, New Mexico, pp. 92-98, 2007.

[10] Nazario, J., "PhoneyC: A Virtual Client Honeypot," *in Proceedings of 2nd USENIX Workshop on Large Scale Exploits and Emergent Threats*, USA, pp. 1-8, 2009.

[11] Owens J. and Matthews J., "A Study of Passwords and Methods Used in Brute-Force SSH Attacks," *Technical Report*, Department of Computer Science, Clarkson University, 2008.

[12] Ramsbrock D., Berthier R., and Cukier M., "Profiling Attacker Behavior Following SSH Compromises," *in Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, USA, pp. 119-124, 2007.

[13] Riden J., "Server Honeypots Vs. Client Honeypot," available at: http://www.honeynet.org/node/158, last visited 2008.

[14] Schryen G., "An E-mail Honeypot Addressing Spammers' Behavior in Collecting and Applying Addresses," *in Proceedings of the 6th Annual IEEE for Information Assurance Workshop*, NY, pp. 37-41, 2005.

[15] Spitzner L., "Honeypots: Definitions and Value of Honeypots," available at: http://www.tracking-hackers.com/papers/honeypots.html, last visited 2003.

[16] Spitzner L., "Honeypots: Simple, Cost Effective Solution," available at: http://www.securityfocus.com/infocus/1690, last visited 2003.

[17] Sun X., Wang Y., Ren J., Zhu Y., and Liu S., "Collecting Internet Malware Based on Client-Side Honeypot," *in Proceedings of the 9th International Conference for Young Computer Scientists*, China, pp. 1493-1498, 2008.

[18] Watson D., "Honeynets: A Tool for Counterintelligence in Online Security," *Journal of Network Security*, vol. 2007, no. 1, pp. 4-8, 2007.

[19] Wicherski G., "Medium Interaction Honeypots," *Technical Report*, Rheinisch-Westfaelische Technische Hochschule Aachen, 2006.

Masood Mansoori holds a MSc degree in data communication and networking from University of Malaya, Malaysia, specializing in network security. He is currently in the process of pursuing his PhD studies. His research interests are client honeypots and intrusion detection systems.

Omar Zakaria completed his undergraduate degree in computer science at the Computer Centre, University of Malaya in 1994. He obtained his MSc and PhD in information systems security management from the Royal Holloway, University of London, United Kingdom, in 1996 and 2007, respectively. He joined the University of Malaya as a tutor in 1995 in the Pusat Asasi Sains. He was appointed as Lecturer in December 1996, and transferred to Faculty of Computer Science and Information Technology in April 1997. Subsequently, he was promoted to senior lecturer in April 2006. Ever since July 2010, he is now an associate professor at the Department of Computer Science, National Defence

University of Malaysia. His research interests are in information systems security management and human impact management. He has published more than 60 conference and journal papers, both locally and internationally.

Abdullah Gani is a senior lecturer at the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. He holds a PhD degree in computer science from University of Sheffield, UK. His MSc in Information Management and B.Phil degrees were obtained from University of Hull, UK in 1987, 1990 respectively. He is a qualified trained teacher. He has more than 28 years of experience in teaching and learning in a number of subjects of computer science and holding several management posts including head of department. He has designed several academic programs such as Diploma in information, Bachelor of information technology and Master of networking for the faculty. He teaches courses at the Bachelor and Master levels. He has a number of Master and PhD students working on network related domains including Darknet, Overlay networks, and graphical password. His interests include self-computing, network security, wireless networking and e-learning. He has published more than 80 academic papers locally and internationally. He is the principal researcher in a number of grants.