Framework for the Establishment of a Honeynet

N. Veerasamy and Prof. J. P. Eloff

Information and Computer Security Architectures Research Group
Department of Computer Science
University of Pretoria
South Africa
nveerasamy@csir.co.za

Abstract—Honeypots are decoy machines that are placed on the network to attract attackers, whilst also distracting them from more important targets. Honeypots are thus an ideal medium for collecting data that can later be studied to analyse attackers' actions and motives. As a decoy and data collection tool honeypots have become a useful security resource. A Honeynet consist of a number of honeypots and can thus be used to compare attack data, experiment with different setups and gather more information. However to successfully set up a Honeynet a number of design, architectural and implementation considerations need to be taken. The aim of this paper is to provide a framework to guide the establishment of a Honeynet.

Index Terms—honeypot, honeynet, framework.

I. INTRODUCTION

ORGANISATIONS around the world are faced with the daunting task of securing their communication processes and infrastructure. Computer and information system networks form a vital part of the communication backbone, and it is imperative that sufficient security mechanisms be deployed.

Intruders are keen to exploit vulnerabilities present on systems. Implemented security mechanisms are only as strong as the weakest link, and new vulnerabilities are continuously being discovered. Hackers thrive on taking advantage and utilising any opportunity to wreak havoc on systems or steal and abuse resources. Hackers are too inventive and persistent to be ignored.

Despite ongoing research in the computer security field, it is still not possible to measure or completely secure computer systems. Vulnerabilities, as soon as it is discovered, allow intruders to exploit and compromise computer systems. As in any society, not everyone has good intentions and motives. The public domain of the Internet is quite the same. The overwhelming success and the rapid growth of the Internet has made networked computer systems a ubiquitous resource [1]. The interconnectivity facilitated by the Internet leaves little room for anonymity and allows for great transparency of one another. Internet history is filled with examples of exploitation, and this scenario continues to increase at an alarming rate. It is believed that the more interesting the target, the faster the attack will occur. Honeypots are a means of creating an inviting target to lure attackers. If a system is configured and deployed as a honeypot, one can study attackers' strategies when the vulnerabilities are exploited.

Very little is known about attackers [2]. There are many questions surrounding them - who are they, what is the reason/purpose of the attack, how was it done? Ordinary computer security often does not provide answers to these questions. Defence organisations have directed their security focus on collecting information on the enemy to understand and defend against threats. Security is a field that requires continual monitoring and reactive responses. To improve security, you should know your attackers and study and understand them. However, with standard computer security practices not possessing such attack information, the task of understanding an attack and providing better security is quite challenging. Honeypots are one example of such technology that can be employed to understand how attackers work.

Honeypots are a means of attracting and recording attacks as it occurs, which can then be traced in real time or analysed at a later stage. The definition of a honeypot, according to Spitzner, is that it is an information security resource of which the value lies in being probed, attacked or compromised [3]. Why would anyone want to build a system and allow it to be attacked? The reason is that in an attempt to fight back against intruders, what better way is there than to learn from them and follow their actions? A honeypot records their footsteps as they move through a system. This data can be studied and analysed to understand their behaviour. Through observing and learning from hackers, an attempt is made to shed light on attacks: how it is performed, its purpose and how it can be prevented or repaired. Patterns can be identified, for example specific attacks originating from certain countries or the infection of certain worms and viruses.

Honeypots can be used to set up a honeynet. A honeynet consists of a high-interaction honeypot [2]. Honeynets were meant to replicate production systems so that attackers could interact with real operating systems and applications and not just emulated services (as in some honeypots) [3]. A honeynet also refers to a network of honeypots that have been deployed [4]. In this way, honeypot data can be compared and more complex honeypot systems can be exposed to open networks and thus gain an understanding of higher-level attacks.

Honeynets can be implemented in any number of ways- with different operating systems, networking, applications, configurations and logging mechanisms. This paper seeks to guide the setting up of a honeynet by explaining the various design and architectural decisions and implementations. The paper thus proposes a high-level three stage framework that will be used to methodically carry out the establishment of a honeynet.

The paper is therefore structured as follows: the next section will provide a brief introduction to previous documented honeynet development, Section III will discuss the proposed framework and Section IV will provide the concluding comments and close the paper.

II. Background

A study of honeypot literature reveals descriptions of the uses, functionality and implementation of honeypots. This Section seeks to explain the context of honeynet development and previous prescriptions of honeypot/honeynet establishments.

Firstly the distinction between a honeypot and a honeynet will be drawn. A honeypot takes the appearance of an attractive service, set of services, an entire operating system or even an entire network, but is in reality a tightly sealed compartment built to lure and contain an attacker [5]. Several honeypots can be assembled into networks of honeypots called honeynets, and because of the wealth of data collected through it, honeynets are considered a useful tool to learn more about attack patterns and attacker behaviour in communication networks [4]. From the definitions it is evident that honeypots can be implemented in various forms, levels and configurations. A honeynet is therefore built up by setting, linking and collecting data/monitoring a number of honeypots. Various software products, logging tools and architectures have been used as part of honeypot/honeynet development which will briefly be explored next.

Many commercial honeypots have been developed that merely need to be installed and deployed. These consist of software simulation honeypots which are deception programs that emulate system software (SS) and services [3]. A few examples include Back Officer Friendly, CyberCop Sting and Specter. In such instances, the software is installed and the logging mechanisms monitored.

A few architectural implementations have also been described in literature. For example, Anagnostakisy et al. explained their setup of a shadowed honeypot to detect attacks. The architecture consisted of a number of anomaly detectors (monitoring traffic entering network), and a shadowed honeypot (instance of production system to which suspicious traffic is sent for analyses) [6].

Another architecture, proposed by Spitzner, consisted of a production environment, gateway and honeynet connected to the open network (Internet). The critical element is the Honeywall gateway [in a honeynet], a layer two bridging device that controls and captures all of the attacker's inbound and outbound activity [7]. Spitzner proposes that the traffic in a honeynet should go though a gateway, so as to both capture and control the activity [7].

A honeynet architecture, Gen II, was explored by d'Orey et al. Gen II honeynets consist of an isolated network segment where a honeywall machine mediates the network traffic going in and out of the honeypot [8]. The packet capturer TCPdump and the intrusion detection mechanism Snort were used for the data capture and analysis in this scheme.

Key to any honeypot is a sensing device (typically the honeypots itself and can be either low-interaction (simulates certain services or vulnerabilities) or high-interaction honeypots (complete operating system with more realistic capabilities and functionality)) and a logging mechanism (intelligent software like Sebek, an Intrusion Detection System (IDS) like Snort, a packer capturer like Ethereal or even a layered combination of these technologies) [9]. These critical components form the basis of a honeypot. In a honeynet the sensing and logging could take place at a central point or be distributed across the honeypots.

The discussion in this section is indicative that many authors have described critical elements and specific implementations of honeynets. However, a useful guide explaining how a honeynet should be set up is proposed by the author so as to offer support to the successful implementation of a honeynet.

Having carried out a few experiments to establish a honeynet, various problems were encountered. These experiments consisted of setting up a honeynet of virtual machines, installing, networking and configuring each honeypot and the system overall. The various problems encountered were helpful in understanding many architectural, design and analysis concepts. Through the experimentation process many lessons were learnt. As a result, a framework for the establishment of honeypots was developed. According to the Whatis online Computer dictionary and encyclopedia a framework serves to support and guide the development or building of a real or conceptual structure [12]. In this context, the framework proposed by the author discusses the various considerations and decisions that should be made before and during the implementation of a honeynet.

Honeypots and honeynets can be set up in various ways with numerous architectural, design and implementation decisions to be made. The goal of this paper is to describe a framework that will guide the establishment of a honeynet. Thus in the next section a basic strategy as well as implementation guidelines will be provided to facilitate the setting up of a honeynet.

III. Framework

In the previous sections an overview and introduction to honeypot/honeynets were given, as well as a brief description of a few examples of honeypot/honeynet establishments. The requirement for a framework to guide the establishment of a honeynet was identified. Various considerations should be made and the suggested framework aims to facilitate the process.

This section proposes a high-level three phase framework for the establishment of a honeynet. Each of these stages will be elaborated. The tasks in each category serve as an outline of the actions that need to be taken. By stating three broad categories, the framework can be adapted as required when new considerations are discovered. The broad stages are show in Figure 1. The framework also suggests that the stages can be cyclic with the process being repeated for different contexts.

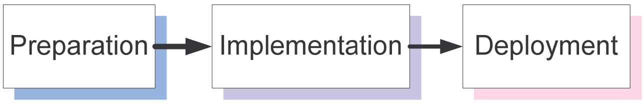

Fig 1. Stages in framework of honeynet establishment

A. *Preparation stage*

The preparation stage consists of the activities/considerations shown in the Fig2.

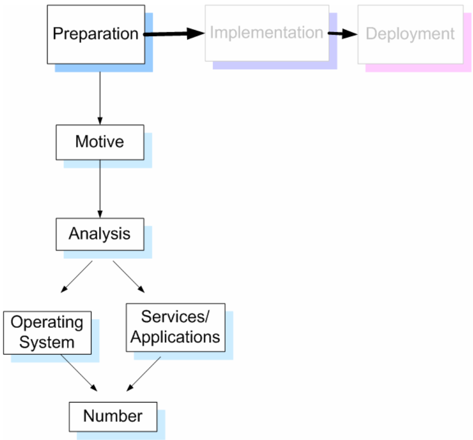

Fig 2. Preparation stage

Motive

The motive or aim behind deploying a honeypot must be established. This is necessary as the aim will provide insight into how the honeypot should be set up. The author's overall impression from studying available literature is that some motives for using honeypots' are:

1) Data capturing- if the approach is statistical in approach [3]

2) Individual investigation to track attackers and look at their behaviour

3) Production system defence to improve the security of actual production system by studying the data on a honeypot configured as a production system [3]

4) *IDS* a honeypot as a component of an IDS or for IDS development (identification of signatures) [2]

5*) Determine trends and patterns* by studying the data

6) *Formulate attack study methods* by devising different ways of examining the attack data

7) *Serve as a decoy* [2]

To further elaborate on the above-mentioned motives one purpose would lie in publishing statistical findings. By formulating statistics much insight into trends and patterns could be gained. Another objective would be to correlate released bugs, worms, viruses or other exploits with activity on a honeypot. In individual investigations, the motive would be understand particular attackers- to investigate who they are, where are they from, what were they interested and what did they do. Organisations are keen to know who the "bad guys" are. Another intention would be to improve the security of a network by implementing mirror copies of the actual production system and studying the attacks, adding security features and deploying on the network to test out its functionality. Setting up a honeypot to prevent attackers from attacking critical systems is another motive. In this way, the attacker spends time exploring the honeypot instead of invading the actual system. The honeypot distracts and keeps the attacker occupied.

The motive behind setting up a honeypot can also be an organization or directive from higher management. It may be the case that a group of people are interested in honeypots as a research tool. Under their directive a honeypot should be set up and studied. The technical team will be responsible for deciding how the analysis and implementation will be carried out. The considerations for establishing a honeynet follow. These issues form the foundation of the honeynet and should therefore be carefully thought out.

Analyses

Decide the type of analysis that will be carried out at the beginning of the project. This is necessary as the analysis scheme drives the entire system: it determines the logging mechanisms, the expected format, required output and actually is linked to the overall aim of the system. The analysis scheme is integral as the analysis requirements need to be built into the system design and implementation. The analysis scheme provides direction for the system. Implementing a honeynet and deciding that the analysis should have been carried out in a different way can seriously impact the system. New applications would have to be installed, additional configuring of the system would have to be carried out, the different components would have to be integrated again, etc. By understanding the motive for the system, the types of analysis can be identified. Thereafter the specific logging programs, applications or tools should be decided on. Identifying the programs, tools and applications, will result in a specification of the type of data that will be captured.

The raw network traffic can be captured and analysed. The logs generated from auditing programs can be studied. Specialised logging programs can be installed. Scripts/Programs can be written to filter/summarise logs and traffic capture and thus report of the attack behavior. The logs/traffic capture could also be manually studied to detect attack actions and patterns. IDS entries can be correlated with data capturing logs and individuals investigated.

Each motive for establishing a honeypot has considerations regarding the analysis methods.

1) Data capturing- If the approach is statistical in nature the traffic capturing/logs can be analysed and be reported back in the form of statistical findings of the attack behaviour. For example, peak attack periods or top attackers can be identified.

2) Individual investigation to track attackers and look at their behaviour. Scripts/ programs can automate the processing of huge amounts of log/traffic data and thus summarise attack behavior. The analysis can also be carried out without the use of automation, but through a manual study of the data captured- similar to a forensic analysis in which the identification and investigation of suspicious activities is carried out. A description of attackers' actions can be compiled by studying the systems logs/traffic capture. It is often insightful to identify and list the actions carried out by an attacker.

3) Production system defence to improve the security of actual production system by studying the data on a honeypot configured as a production system. The honeypot will need to be configured as the production system and deployed on the network. If the honeypot is compromised the honeypot data can be studied to improve the security mechanisms on the production system itself. The honeypot with improved security can once again be deployed to test out these measures. The honeypot can also be studied to identify what actions the attacker took and determine the nature/motive of the attacks. This provides insight into the purpose of the attacks.

4) Determine trends and patterns. Statistically analyzing the data will help detect patterns. Attack activity could also be correlated with specific security events/exploits. New trends and types of behaviour can be found to occur.

5) Formulate attack study methods by devising different ways of examining the attack data. This could involve program/scripts to process attack data and produce reports, use of data-mining techniques to detect patterns and even individual investigations of attackers.

6) Comparison of results. In this case different honeypots can be set up with varying levels of security. After the honeypots are deployed and the system attacked the honeypots can be studied to determine what attack activity occurred. Various comparison scenarios are possible: attacks on various deployments with different application and security installations, virtual machines and real machines, a prescribed deployment versus an open system, varying degrees of hardening, etc.

Many open source and Windows based tools are available. These range from honeypot specific tools to logging and traffic capturing applications. A decision regarding the route the analysis will follow needs to be taken to determine the applications and tools that need to be installed.

Number

Formulate the number of honeypots that will be installed. A number of honeypots with different operating systems and applications can be set up to collect data. This forms a good basis for study and comparison of attack data.

As the name implies, a honeynet consists of a number of honeypots and a decision should be made as to how many honeypots will be set up. This step forms the basis of the next two steps: Selecting the Operating System and Services. The honeypots can be both real, virtual or combination of the two.

If the analysis will be comparison based, the number of honeypots will be determined by the comparison scheme. For example if attacks on different web servers are to be compared, a decision to install four web servers, IIS on Windows 2000,Apache on a Windows 2000 machine, IIS on a hardened Windows 2000 machine and Apache on a Suse machine. In this way, the decision regarding the number of honeypots will be determined based on how the systems will be compared according to application, operating system and security.

Operating System

Select the operating systems to install on the different honeypots that will be forming the Honeynet. The choice of operating system is often determined by the selection of services and applications. Certain applications can be run on a multiple operating systems whilst others are specific to certain operating systems. For example the logging program Ethereal can be run on both Windows and Linux whilst the web server IIS is specific to Windows.

One comparison scheme is to compare attacks on different operation systems. If different security implementations want to be investigated, one operating system will be chosen with different levels of security on each operating system. Another example is to test different levels of security on one operating system, for example a base installation of XP, XP with service pack one or service pack two or even fire walled and installed with anti-virus software.

Services and Applications

This step involves deciding on the configuration settings for the various honeypots. Decisions regarding the services/applications, vulnerabilities, open ports, etc will have to made. This step serves to determine what will be installed on the machines and how they will be configured.

Decisions regarding how the honeypot will be hardened should be taken. If base installations are to be studied, no additional applications will be installed. In the case of comparing escalating levels of security, honeypots will have different service packs, patches and security applications installed. In prior setups, several security features (firewall and anti-virus software) were installed in an effort to keep the system operational. However, the analysis scheme will dictate the degree of security that should be employed on the system.

If a production system is being mirrored, the honeypot will have to be set up to emulate a production system machine. The honeypot will therefore have to installed and configured with all the services and applications that the production system has.

In general decisions regarding which: ports will be left open, web/email/file servers to install (Apache, IIS, etc), patches, logging and security mechanisms, services and other applications to install should be made.

B. Implementation stage

The Implementation stage consists of the activities shown in Fig. 3.

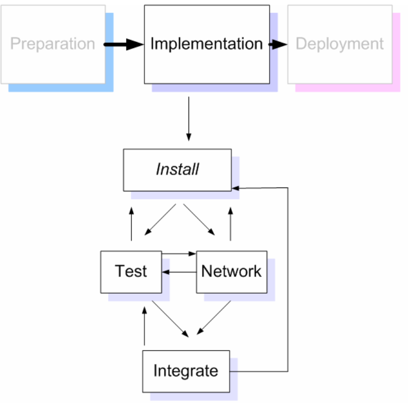

Fig3. Implementation stage

Installation

The installation process involves transforming the design in previous steps (number, operating system and services and applications) into an operational system. All the operating systems, services/applications and security measures (if required) should be installed. Each operating system and piece of software requires time to install as well as configure.

During the installation process it is recommended that backups be made. This ensures that in case of a system crash (quite common when working with honeypot as a result of attackers wreaking havoc on systems), installation does not have to commence from the start once again. The backups can be used to restore the images and in this way reduce the time and effort in getting the system operational once again. Backups can be made of the operating system installation with or without installing the applications.

Initially when installing a virtual machine, a complete operating system installation is required. Thereafter the initial virtual machine installation can be used as a base for other virtual machines. The virtual machine image is copied, renamed and started up. Additional services and applications can then be installed. This saves time and effort in installing new operating systems each time.

Networking

This aspect in the establishment of honeynets, involves connecting the honeypots to a network as well as ensuring transparency for data capture whilst offering sufficient coverage so as not to reveal the true nature of the system. Connections to the Internet, internal or external networks will have to be established.

Virtual machines set up as honeypots will have to be configured to be able to connect to the Internet/other networks for the attack data to be captured. Various networking options are available for virtual machines.

Testing

Testing ensures that the system is operational. This ensures that the applications/services are running properly, and also that attacks can reach the system and are recorded by the logging services running. The connections to the Internet/other network will also have to be tested. Simulated attacks are often a means of testing the system. For example simulate an attack by connecting through telnet or ftp connection, try retrieving files and check the logs for attack capture. Simulated attacks can help test whether the applications, networking and logging is successful.

Another aspect of testing is to ensure that the backups are operational. It is often the case that the backups can become corrupted in the copying process. After making the backups it is essential to test that the operating system can be restored from the hard drive, partition or DVD.

Integration

Integration comes into play when the various honeypots are to be deployed together, especially in the case of virtual machines. If the honeypots are virtual and running on one machine, the security of the host should be set up carefully. In addition, the running of all the honeypots should be tested together to ensure functionality of the system as a whole. The testing and integration steps can be tricky. Individually the different machines may be operational. However functioning at the same time, in harmony, can require some experimentation.

Overview

The initial stages involve the planning and design of the system. The upfront design will form the basis of the system. Much experimentation is required in the installation, networking testing and integration process. It can often become a cyclic process whereby an application is installed, tested and integrated before another piece of software is installed. An alternative is to install many applications, network, test and integrate before going through the processes again with another batch of software if required.

C. *Deployment*

The deployment stage consists of the activities shown in Fig. 4. :

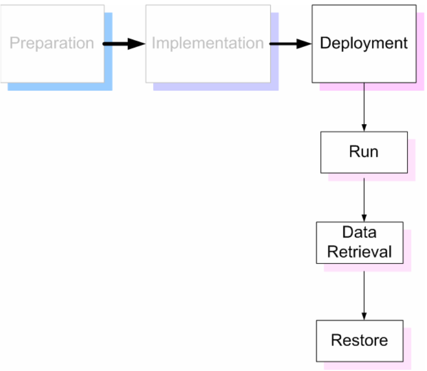

Fig4. Deployment stage

Run

Once installed, networked and tested to be operational the Honeynet can be deployed. This involves opening a connection the Internet or deploying on the network and the logging of data. The system can be left open for a set period of time and analysed. Another option is run the system continuously and carry out statistical analyses.

Data Retrieval

The data retrieval process involves stopping the system, if necessary to retrieve the logs. Another possibility is to study the actual honeypot. In this way the logs or the honeypot itself can be examined.

Regular data retrieval should be instituted. If the honeypot will be running for a period of time and stopped, the logs can be taken off the system and saved onto another format/machine for analyses. The logs could also be saved without stopping the system by setting up the system to automatically do this.

In some cases the honeypot will actually be studied. The data retrieval is thus carried out by stopping the system and removing it from the Internet or network to be studied.

Restore

If data collection is to continue, the system should be started up again. In the case of stopping the system and retrieving the logs, the restore is achieved by a reconnection of the system to the Internet or network. In the case of the actual operating system being investigated, the backup copy of the operating system with program installations should be restored on the honeypot and deployed once again.

IV. Conclusion

The main aim of this paper is to define a framework for the establishment of a Honeynet. Various design, architectural and implementation considerations need to be made during the development of a honeynet. Practical implementations of establishing a honeynet, has taught that certain logic and steps need to be followed to properly set up a honeynet. This enables for sufficient preparation and design followed by a structured implementation and a planned deployment.

Similar tasks were grouped together. It is often the case that the decisions in each stage are interleaved with each other. The overall stages are Preparation, Implementation and Deployment. These stages can be elaborated with other steps as new requirements are identified.

References

[1] S. Krasser, J. Grizzard and L. Owen, “The Use of Honeynets to Increase Computer Network Security and User Awareness”, Georgia Institute of Technology, School of Electrical and Computer Engineering,. Available: http://www.ece.gatech.edu/research/labs/nsa/papers/use_of_honeynets.pdf

[2] L. Spitzner, “Honeypots: Definitions and value of honeypots”, tech. rep., Honeynet, 2003. Available http://www.tracking-hackers.com/papers/honeypots.html.

[3] L. Spitzner, “Honeypots: Tracking Hackers”, Addison-Weasley, December 2002, pp. 1-86.

[4] T. Holz and F. Raynal, “Detecting Honeypots and other suspicious environments”, IEEE Workshop on Information Assurance and Security, 2005, pp. 29-36.

[5] I. Kuwatly, M. Sraj, A. Masri and H. Artail, “A Dynamic Honeypot Design for Intrusion Detection”, IEEE/ACS International Conference on Pervasive Services, 2004, pp. 1-10.

[6] M. Dacier, F. Pouget and H. Debar, “Honeypots: Practical Means to Validate Malicious Fault Assumptions”, Pacific Rim International Symposium on Dependable Computing, 2004.

[7] L. Spitzner, “Honeypots: Catching the Inside Threat”, IEEE Computer Security Applications Conference, 2003, pp. 170-179.

[8] M. d’Orey, P. de Andrade Carbone and P. Licio de Geus, “A Mechanism for Automatic Digitial Evidence Collection on High-Interaction Honeypots”, IEEE Workshop on Information Assurance and Security, 2004, pp. 1-8.

[9] N. Garner, “Honeypots for Incident Handling Education”, Sans Institute, Available: http://www.giac.org/certified_professionals/practicals/gcih/0494.php.

[10] C. Kreibich and J. Crowcroft , “Honeycomb – Creating Intrusion Detection Signatures Using Honeypots”, ACM SIGCOMM Computer Communication Review, 2004, pp. 51-56.

[11] Whatis.com, “Framework”, IT Encyclopedia. Available: http://whatis.techtarget.com

[12] B. Scottberg, W. Yurcik and D. Doss,”Internet Honeypots: Protection of Entrapment?”, Symposium on Technology and Society (ISTAS), June 2002.

Namosha Veerasamy has obtained a B.Sc. IT Computer Science degree and a B.Sc. Computer Science (Hons) degree with distinction at the University of Pretoria. She is currently a researcher at the Council for Scientific and Industrial Research (CSIR) in Pretoria.

Jan Eloff received a PhD (Computer Science) from the Rand Afrikaans University, South Africa. Since October 2002 he is Head of Department and full professor at the Department of Computer Science, University of Pretoria.
He has published extensively in a wide spectrum of accredited international subject journals and organized various international and national conferences were. He has delivered papers at leading information security conferences on an international level.