Honeypots: Basic Concepts, Classification and Educational Use as Resources in Information Security Education and Courses

Miguel Hernández y López
General Systems Directorate – Government of the State of Tamaulipas, Ciudad Victoria, Tamaulipas, Mexico

Carlos Francisco Lerma Reséndez
General Systems Directorate – Universidad Autónoma de Tamaulipas, Ciudad Victoria, Tamaulipas, Mexico

miguel@honeynet.org.mx cflerma@uat.edu.mx

Abstract

This paper deals with the basic aspects of Honeypots, their use in modern computer networks and their implementation in educational environments. Initially, the implementation of Honeypots solves a common problem of Information Security and Forensics: The dissection of the elements that make up an attack against a computer system. Next, the paper explains the different types and functions of Honeypots once they are implemented in a network in order to make a distinction in terms of what is needed for the Honeypot to do. Finally, the use of Honeypots in educational environments, its benefits and the use of virtualization technologies is discussed.

Keywords: Honeypot, honeynet, intrusion detection system, computer forensics, virtualization.

Introduction

Traditionally, the nature of the field of Information Security has been purely defensive. Firewalls, intrusion detection systems and encryption are mechanisms used defensively to protect information resources (Dunsmore, Brown, & Cross, 2002). The strategic dogmas of Information Security consist in defending the information infrastructure as well as possible, detect possible failures in the defensive structure and promptly react to those failures, preferably in a proactive way (Roberti & Bonsembiante, 1995). The nature of the existence and operation of the "information enemy" is purely offensive, due to always being ready to attack.

Honeypots have demonstrated their value as a research tool in the field of Information Security and as a powerful educational tool in the modern classroom (The Honeynet Project, 2005). Many researchers and organizations, public and private, which are part of the Information Security Community, are currently using trap-style networks to learn the tactics, techniques and proce-

dures used by the hacker community to break into information vaults without authorization, which could contain potentially sensitive information. Additionally, Honeypots provide teachers and students with a tool that allows them to dissect security events thoroughly and in a modular way, which is a very desirable characteristic when it comes to teaching Information Security courses.

This paper analyzes the functions of Honeypots and their technology, which are becoming not just a key component in a layered system of protection against intruders but also a valuable simulation resource in the academic field.

Honeypots – What They Are and How They Work

Honeypots are a new technology with enormous potential for the Information Technology community. The first concepts regarding them were introduced by various icons in Information Security, such as those defined by Cliff Stoll in the book "The Cuckoo's Egg" (2002) and the works of Bill Cheswick, documented in the book "An Evening with Berferd" (1997). Since then, those concepts have been in continuous evolution, developing in an accelerated way and becoming a powerful security tool nowadays (Riebach, Rathgeb & Tödtmann, 2005).

Honeypots are, in their most basic form, fake information severs strategically-positioned in a test network, which are fed with false information disguised as files of classified nature. In turn, these servers are initially configured in a way that is difficult, but not impossible, to break into them by an attacker; exposing them deliberately and making them highly attractive for a hacker in search of a target (Spitzner, 2002). Finally, the server is loaded with monitoring and tracking tools so every step and trace of activity left by a hacker can be recorded in a log, indicating those traces of activity in a detailed way.

The main functions of a Honeypot are (Pouget & Holz, 2005):

- To divert the attention of the attacker from the real network, in a way that the main information resources are not compromised
- To capture new viruses or worms for future study
- To build attacker profiles in order to identify their preferred attack methods, similar to criminal profiles used by law enforcement agencies in order to identify a criminal's *modus operandi*
- To identify new vulnerabilities and risks of various operating systems, environments and programs which are not thoroughly identified at the moment

In a more advanced context, a group of Honeypots becomes a Honeynet, thus providing a tool that spans a wide group of possible threats which gives a systems administrator more information for study. Moreover, it makes the attack more fascinating for the attacker due to the fact that Honeypots can increase the possibilities, targets and methods of attack.

Classification of Honeypots

Honeypots can be classified according to two criteria: Implementation Environment and According to their Level of Interaction. This classification criterion makes it easier to understand their operation and uses when it comes to planning an implementation of one of them inside a network. (See Figures 1 and 2.)

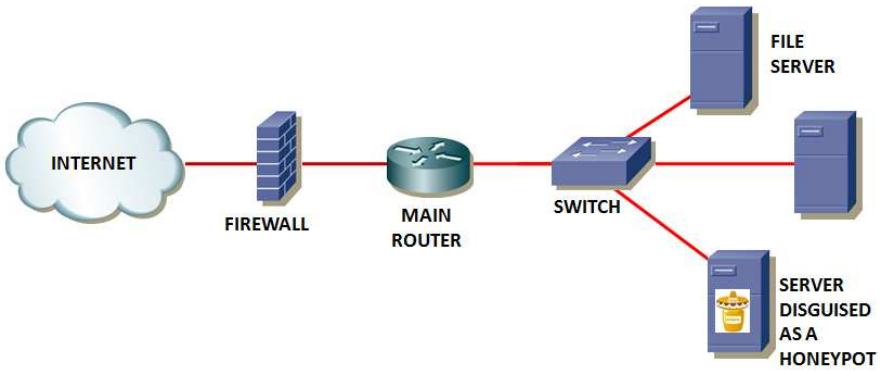


Figure1. A Honeypot placed inside the network

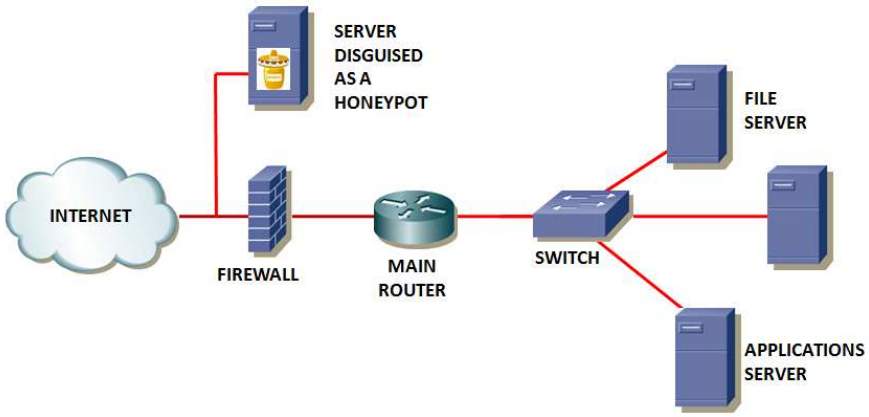


Figure 2. A Honeypot placed in a perimeter network

Honeypots According to their Implementation Environment

Under this category, we can define two types of Honeypots: Production Honeypots and Research Honeypots.

Production Honeypots: Those used to protect organizations in real production operating environments. They are implemented parallel to data networks or IT Infrastructures and are subject to constant attacks 24/7. These honeypots are constantly gathering more importance due to the detection tools they provide and because of the way they can complement network and host protection.

Research Honeypots: These Honeypots are not implemented with the objective of protecting networks. They represent educational resources of demonstrative and research nature whose objective is centered towards studying all sorts of attack patterns and threats. A great deal of current attention is focused on Research Honeypots, which are used to gather information about the intruders' actions. The Honeynet Project, for example, is a non-profit research organization focused in voluntary security using Honeypots to gather information about threats in cyberspace.

Honeypots According to their Level of Interaction

Within this classification criterion, the term "Level of Interaction" defines the range of attack possibilities that a Honeypot allows an attacker to have. These categories help us understand not

just the type of Honeypot which a person works with, but also help define the array of options in relation to the vulnerabilities intended for the attacker to exploit. Those are the most important traits when it comes to starting the construction of an attacker's profile.

Low Interaction Honeypots: Normally, Low Interaction Honeypots work exclusively emulating operating systems and services. The attacker's activities are limited to the Honeypot's level and quality of emulation. The advantage of a Low Level Honeypot lies upon its simplicity, due to the fact that they tend to be easy to use and maintain, with minimum risks. For example: An emulated FTP service, listening on port 21, is probably emulating an FTP login or will possibly support additional FTP commands but it does not represent a target of critical importance due to the fact that it is not possibly linked to a FTP server containing sensitive information.

Generally, the implementation process of a Low Interaction Honeypot consists of installing any kind of operating system emulation software (i.e. VMWare Workstation or Server), choosing the operating system and services to be emulated, establishing a monitoring strategy and let the software operate by itself in a normal manner. This "plug-and-play" type of process makes it extremely easy to use a Low Interaction Honeypot. The emulated services mitigate the risk of penetration, containing the intruder's activities so he/she never gains access to a real operating system that could be used to attack or damage other systems.

The main advantage of Low Interaction Honeypots lies in the fact that they only record limited information since they are designed to capture predetermined activity. Due to the fact that emulated services can only go as far as certain operational thresholds, this feature limits the array of options that can be advertised towards a potential intruder. Likewise, it is relatively simple for an attacker to detect a Low Interaction Honeypot due to the fact that a skilled intruder can detect how good the emulation capabilities are as long as he/she has enough time to verify this.

Effective examples of Low Interaction Honeypots are: Specter, Honeyd and KFSensor (HoneyD, 2007).

High Interaction Honeypots: These Honeypots constitute a complex solution because they involve the utilization of operating systems and real applications implemented in real hardware, without using emulation software, running in a normal way; many times directly related to services such as databases and shared folders. For example, if a Honeypot needs to be implemented on a real Linux system running a FTP server, a real Linux system needs to be built on a real computer and a real FTP server will need to be configured.

The aforementioned solution offers two advantages: Initially, there is the possibility of capturing large amounts of information about the *modus operandi* of attackers because intruders are interacting with a real system. This way, a systems administrator is in a position to study the full extent of the attacker's activities: anything ranging from new rootkits, zero-days up to international IRC sessions. Finally, High Interaction Honeypots do not assume anything about the possible behavior the attacker will display since they only provide an open environment which captures every one of the attacker's moves but they still offer a wide scope of services, applications and information vaults posing as potential targets related to those services which we specifically want to compromise. This allows high interaction solutions to come in contact with unexpected behaviors.

However, the latter capability also increases the risk of attackers using those operating systems as a launch pad for attacks directed at internal systems which are not part of a Honeypot, turning bait into a weapon. As a result of this, there is a need to implement additional technologies which prevent the attacker from damaging non-Honeypot systems that deprives the compromised system of its capabilities of becoming a platform to launch potential attacks.

Currently, the best example of a High Interaction Honeypot is represented by Honeynets (The Honeynet Project, 2007a, 2007b).

Advantages and Disadvantages

Honeypots are incredibly simple concepts that offer powerful advantages:

- New Tools and Tactics: They are designed to capture anything that interacts with them, including tools or tactics never seen before, better known as “zero-days” (Leita, Dacier, & Massicotte, 2006).
- Minimal Resources: This means that resources can be minimal and still enough to operate a powerful platform to operate at full scale. For example: A computer running with a Pentium Processor with 128 Mb of RAM can easily handle an entire B-class network.
- IPv6 Encryption: Unlike most security technologies, Honeypots also work in IPv6 environments. The Honeypot will detect an IPv6-based attack the same way it does with an IPv4 attack (Man, 2003).
- Information: Honeypots can gather detailed information, unlike other security incident analysis tools.
- Simplicity: Because of their architecture, Honeypots are conceptually simple. There is not a reason why new algorithms, tables or signatures must be developed or maintained.

Just like any other technology, Honeypots also have significant weaknesses inherent to their design and functioning. This is because Honeypots do not replace current technologies, but instead work along with other existing technologies:

- Limited Vision: They can only scan and capture activity destined to interact directly with them. They do not capture information related to attacks destined towards neighboring systems, unless the attacker or the threat interacts with the Honeypot at the same time.
- Risk: Inherently, the use of any security technology implies a potential risk. Honeypots are no different because they are also subject to risks, specifically being hijacked and controlled by the intruder and used as a launch pad for subsequent attacks.

Educational Application of Honeypots

The field of Information Security has become a new challenge to educators when it comes to developing quality material that allows students not just to understand basic concepts (Lerma, 2007), but also to manipulate tools that allow them to dissect the strategies, exploits, tools and methods used by attackers. Moreover, the field of Information Security is based in establishing a set of security guidelines and frameworks that are often tailored according to specific situations and organizations.

In the educational arena of Information Security, Honeypots provide a safe and manageable environment that can be deployed in a controlled fashion (lab) and can also be implemented in a live production setting (actual network). This capability is also enhanced by the use of virtualization technologies, which allow a Honeypot to be implemented in a matter of minutes and to be stored with particular settings, according to the vulnerability and exploits that will be subjects of study in a particular lesson (Collins, 2006).

According to Wiley (2000), Honeypots fall into the category of Generative-instructional learning objects which are defined as a “combination of objects providing advanced visual and auditive

capabilities with advanced interactive features, allowing a high level of hands-on experience". Leaving beside the visual and auditive capabilities, Honeypots (and specially those mounted on virtualized platforms (Provos, 2004)) allow a high level of interaction between students and the machine. Users can manipulate important elements in Information Security Forensics lessons such as:

- Hardware and software settings
- Services installed in a server
- Operating system logs (especially security and event logs)
- Network settings (including logical network placement of the Honeypot)
- Installed applications (and their respective settings and roles played inside the Honeypot)
- Users and user groups (including group membership and capabilities)
- "Dummy" information inside the server (and its nature)

As stated before, the use of virtualization is a powerful tool when implementing a Honeypot due to the features this technology provides (Border, 2006). Not only do they provide a virtual environment that can replicate a real one in full, but once a Honeypot has been compromised beyond its normal limits or damaged beyond actual recognition, it can be taken offline and brought back to its original settings in a matter of minutes.

Virtualized Honeypots offer a very good feature when it comes to its "level of vulnerability": it can easily be managed and set according to a particular topic or lesson. Due to the modern feature of automatic updating found in most operating systems, vulnerabilities are easily patched and blocked, preventing attackers from exploiting them. When teaching specific topics and subjects related to Information Security and Computer Forensics, those vulnerabilities are the true matter of the course and they cannot be blocked or patched because that eliminates the subject of study. In this case, virtualized Honeypots can be customized with specific levels of vulnerabilities and once they have been patched or eliminated as a part of a specific laboratory, the original virtual image of the Honeypot can be restored with the needed level of vulnerability needed.

Practical Applications

When used with productive purposes, Honeypots provide protection to an organization through prevention, detection and response to an attack. When used with research purposes, they gather information related to the context in which the Honeypot was implemented. Some organizations study the tendencies displayed by intrusive actions, while others shift their interest towards prediction and anticipated prevention.

Honeypots can help prevent attacks in various forms:

- Defense against automated attacks: These attacks are based on tools which randomly scan entire networks, searching for vulnerable systems. If a vulnerable system is located, these automated tools will attack and take over the sytem (with worms which replicate inside the victim). One of the methods to protect a system from the aforementioned attacks is to reduce the speed of their scanning activities in order to stop them later on. Known as "Sticky Honeypots", these solutions monitor unutilized IP space. When systems are analyzed, Sticky Honeypots interact with those systems and reduce the speed of the attack. This is attained by using a variety of TCP tricks, such as setting the Window Size to zero or constantly putting the attacker on hold. This technique is excellent to reduce the speed or prevent the dissemination of worms which have penetrated the internal network.

- Protection against human intruders: This concept is known as conning or dissuasion. The idea behind this countermeasure is to confuse the attacker and make him waste time and resources while he is interacting with the Honeypot. As the process takes place, the attacker's activity can be detected and there is enough time to react and stop the attack.
- Surgical Detection Methods: Traditionally, detection has been an extremely difficult task to carry out. Technologies like Intruder Detection Systems and Logging Systems have been deficient for many reasons: They generate excessive amounts of information, inflated percentages of false positives and do not possess the ability to detect new attacks, work in encrypted mode or work in IPv6 environments. Honeypots excel in the field of intrusion detection by solving many of the problems of classic detection. They reduce false positives, capture small amounts of data of crucial importance like unknown attacks and new methods to exploit vulnerabilities (zero-days), as well as operating in IPv6 environments.
- Cyber-Forensics: Once a network administrator determines that one of his/her servers was illegally compromised, it is necessary to immediately conduct a forensic analysis in the compromised system in order to produce an assessment of the damages caused by the attacker. However, there are two problems affecting incident response:
    - Frequently, compromised systems cannot be disconnected from the network in order to be analyzed and;
    - The amount of generated information is considerably large, in such a way that it is very difficult to determine what the attacker really did inside the system.

Honeypots help solve both problems due to being excellent incident analysis tools, which can be quickly and easily taken offline to conduct a thorough forensic analysis without impacting daily enterprise operations. The only activity traces stored by a Honeypot are those related to the attacker, because they are not generated by any other user but the attacker. The importance of Honeypots in this setting is the quick delivery of previously-analyzed information in order to respond quickly and efficiently to an incident.

References

Border, C. (2007). The development and deployment of a multi-user, remote access virtualization system for networking, security, and system administration classes. *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, Covington, Kentucky, USA.

Cheswick, W. (1997). An evening with Berferd. In D. E. Denning & P. J. Denning (Eds.), *Internet besieged: Countering internet scofflaws* (pp. 103-117). New York, NY: ACM Press/Addison-Wesley.

Collins, D. (2006). Using VMWare and live CD's to configure a secure, flexible, easy to manage computer lab environment. *Journal of Computing in Small Colleges, 21*(4), 273-277.

Dunsmore, B., Brown, J., & Cross M. (2002). *Mission critical!: Internet security.* Syngress.

HoneyD. (2007). Retrieved 9 October 2007 from http://www.honeyd.org

Honeynet Project, The. (2005). *Know your enemy: Learning about security threats* (2nd ed.). Addison-Wesley Professional.

Honeynet Project, The. (2007a). *Know your enemy: Honeynets*. Retrieved on 7 October 2007 from http://www.honeynet.org/papers/honeynet/index.html

Honeynet Project, The. (2007b). Know Your Enemy: GenII Honeynets. Retrieved 2 September 2007 from http://www.honeynet.org/papers/gen2/

Leita, C., Dacier, M., & Massicotte, F. (2006). Automatic handling of protocol dependencies and reaction to 0-day attacks with ScriptGen based honeypots. *RAID 2006*, LNCS 4219, 185–205. Berlin, Heidelberg: Springer-Verlag.

Lerma, C. F. (2007).Creating Learning Objects. *Proceedings of the InSITE 2007 Conference.* Ljubljana, Slovenia. June 22-25, 2007. Retrieved from http://proceedings.informingscience.org/InSITE2007/InSITE07p113-126Lerm283.pdf

Man, Y. R. (2003). *Internet security: Cryptographic principles, algorithms and protocols*. Wiley.

Philippine Honeynet Project, The. (n.d.). Honeynets learning. Retrieved 27 August 2007 from http://www.philippinehoneynet.org/docs/honeynetlearning.pdf

Pouget, F., & Holz, T. (2005). A pointillist approach for comparing honeypots". In K. Julisch & C. Kruegel (Eds.), *Intrusion and malware detection and vulnerability assessment*. Berlin / Heidelberg: Springer.

Provos, N. (2004). A virtual honeypot framework. In *Proceedings of the 12th USENIX Security Symposium*, 2004.

Riebach, S., Rathgeb, E. P., & Tödtmann, B. (2005). Efficient deployment of honeynets for statistical and forensic analysis of attacks from the Internet. In *Proceedings from IFIP-TC6 Networking Conference 2005*. Waterloo, Ontario, Canada.

Roberti, R., & Bonsembiante, F. (1995). *Llaneros Solitarios Hackers, Guerrilla* (1st ed.). Espasa Calpa.

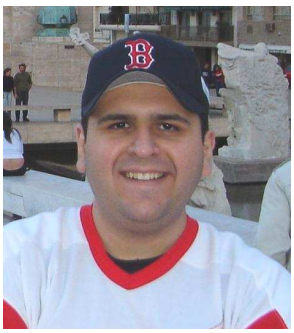
Spitzner, L. (2002). *Honeypots: Tracking hackers.* Boston: Addison-Welsey.

Stoll, C. (2002). *The cuckoo's egg: Tracking a spy through the maze of computer espionage* (1st ed.). Pocket.

Wiley, D. A. (2005). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The instructional use of learning objects*. Retrieved February 17, 2005 from: http://reusability.org/read/chapters/wiley.doc

Biographies

Miguel José Hernández y López is a Service Engineer at the Directorate of Systems of the Government of the State of Tamaulipas. He holds a Bachelor's of Management Information Systems from the School of Business (Unidad Académica Multidisciplinaria de Comercio y Administración – Victoria) at the Universidad Autónoma de Tamaulipas, in Ciudad Victoria, Mexico. Founding member of the Mexican Honeynet Project, he has been a keynote speaker in several information security and open source software events in Mexico and abroad, including a notable participation in the 6th Convention on Open Source Software of the Universidad de Mendoza in Mendoza, Argentina.



Carlos Francisco Lerma Reséndez, MSc is a Service Engineer heading the area of IT Monitoring at the Directorate of Information Technology and Telecommunications (Dirección de Informática y Telecomunicaciones) at Universidad Autónoma de Tamaulipas in Ciudad Victoria, Mexico. He holds a Master of Science in Telecommunications and Network Management from Syracuse University in Syracuse, New York.