# EXTENDED HONEYPOT FRAMEWORK TO DETECT OLD/NEW CYBER ATTACKS

**Article** · March 2011

Source: DOAJ



**5 authors**, including:

Hemraj Saini
Jaypee University of Information Technology
**87** PUBLICATIONS **145** CITATIONS
SEE PROFILE

Bimal Kumar Mishra
Birla Institute of Technology, Mesra
**123** PUBLICATIONS **928** CITATIONS
SEE PROFILE

# EXTENDED HONEYPOT FRAMEWORK TO DETECT OLD/NEW CYBER ATTACKS

HEMRAJ SAINI

Department of Computer Science & Engineering
Orissa Engineering College, Bhubaneswar-752050

BIMAL KUMAR MISHRA

Department of Applied Mathematics
Birla Institute of Technology, MESRA, Ranchi- 835 215

H.N.PRATIHARI

Department Electronics & Telecommunication engineering
Orissa Engineering College, Bhubaneswar-752050

T.C.PANDA

Department Applied Mathematics
Orissa Engineering College, Bhubaneswar-752050

**Abstract:**

**In cyber space, a hot problem is to detect the newly emerged malicious objects. There are significant methodologies to detect the early detected malicious objects but not for newly emerged malicious objects. Generally, the widely applicable approach to detect the early detected malicious objects is signature-based detection. But for new malicious objects no signature is existed in the history, therefore, they can not detected by using this approach. New malicious objects can only be detected by using signature-based approach after a significant loss of the assets, as the signature is generated in between the duration.**

**The paper deals to propose an approach to detect the new malicious objects with an optimal cost. Honeypots are generally used to detect the new malicious objects. The available honeypot frameworks are too costly to be afforded by an average organization. Therefore, we are proposing a low cost honeypot framework to detect malicious objects named extended honeypot. The approach is not only cost effective but also better than other approaches in some situations such as in the Intranet which is having more than one LANs and every LAN is having double honeypot.**

***Keywords: Honeypot, Double Honeypot; Malicious Objects; Shared Double Honeypot; LAN; Extended Honeypot framework.***

## 1. Introduction

In present era many of the enterprises or individuals are dealing with the online valuable information for their day to day work. This valuable online information may prone to threats such as malicious attacks by virus, worms, Phishing or Trojans. These malicious attacks may be responsible for major destructions related to stores or consumers. Hence, these attacks must be detected before unwanted destructions. To detect the cyber attack we can use the signature-based intrusion detectors [Zhou, J. *et al.*(2007), Sommer, R. *et al.* (2003), Pouzol, J. P. *et al.* (2003)], because, each attack is having a signature. But in this approach the signature must be already known to the user. Hence, we cannot detect the newly emerged attacks by the signature-based approach.

Another way to detect the attacks is anomaly-based intrusion detectors [Kruegel, C. *et al.* (2003), Ghosh, A. K. *et al.*(1998), Kruegel, C. *et al.* (2002), Swapna, S. *et al.* (2005)]. In this approach, the state of resources is to be computed at an instance and compared to the base profile i.e. threshold values of the resources. By this approach we can detect any attack whether it is newly emerged or older one. But this approach is suffered by large false positives.

The spread of a malicious worm is often an Internet-wide event. The fundamental difficulty in detecting a previously unknown worm is due to two reasons [Thottan, M. *et al.* (2003)]. First, the Internet consists of a large number of autonomous systems that are managed independently, which means a coordinated defense system covering the whole Internet is extremely difficult to realize. Second, it is hard to distinguish the worm activities from the normal activities, especially during the initial spreading phase. Although the worm activities become apparent after a significant number of hosts are infected, it will be too late at that time due to the exponential growth rate of a typical worm.

After seeing the drawbacks of the above well known approaches, we are proposing a different approach to detect the known and unknown attacks based on honeypots. This approach uses the concept of double honeypot and suggesting the sharing of information among the double honeypot to reduce the cost of the system and increase the efficiency of detection and prevention of attacks. Most importantly, the system is able to detect new worms that are not seen before. To understand the honeypot, let us to give the exploration of honeypot.

### 1.1. Honeypot

Before presenting the architecture of our double-honeypot system, we give a brief introduction of honeypot. Developed in recent years, honeypot is a monitored system on the Internet serving the purpose of attracting and trapping attackers who attempt to penetrate the protected servers on a network [Lee, W. *et al.* (2001)]. Honeypot is an unsecured computer, with a goal to let the attackers in, so that, we can observe attackers and protect our network by diverting attacks to honeypots. Unlike other defense measures, it does not try to prevent any attack. But it loves attacks, especially new ones or we can say that honeypot acts like a canary in the mine. The goal of honeypot is simply to learn about attackers.

Honeypots are a highly flexible security tool with multiple uses [Mahoney, M. *et al.* (2004)], such as prevention, detection, or information gathering. Honeypots all share the same concept, a security resource that should not have any production or authorized activity. In other words, deployment of honeypots in a network should not affect critical network services and applications. A honeypot is a security resource whose value lies in being probed, attacked, or compromised [Yamada, Y. *et al.* (2007)]. This means that whatever we designate as a honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited. A honeypot also is a detection and response tool, rather than prevention which it has a little value in [Spitzner, L. (2003)]. Honeypots are increasingly used to provide early warning of potential intruders, identify flaws in security strategies, and improve an organization's overall security awareness. Honeypots can simulate a variety of internal and external devices, including Web servers, mail servers, database servers, application servers, and even firewalls [Edwards, M. J. (2005)].

A better way to think of a honeypot is as an Internet-attached server that acts as a decoy, luring in potential hackers in order to study their activities and monitor how they are able to break into a system. Honeypots are designed to mimic systems that an intruder would like to break into but limit the intruder from having access to an entire network. If a honeypot is successful, the intruder will have no idea that s/he is being tricked and monitored. Most honeypots are installed inside firewalls so that they can better be controlled, though it is possible to install them outside of firewalls. A firewall in a honeypot works in the opposite way that a normal firewall works: instead of restricting what comes into a system from the Internet, the honeypot firewall allows all traffic to come in from the Internet and restricts what the system sends back out. By luring a hacker into a system, a honeypot serves several purposes: The administrator can watch the hacker exploit the vulnerabilities of the system, thereby learning where the system has weaknesses that need to be redesigned. The hacker can be caught and stopped while trying to obtain root access to the system. By studying the activities of hackers, designers can better create more secure systems that are potentially invulnerable to future hackers [Spitzner, L. (2003)].

Here is one simple example to understand the principle of honeypot. Let there is an e-mail server which is responsible to send and receive the e-mails of the persons in an organization. As, all of us know that e-mail is a big carrier of malicious attacks. To detect e-mails having such attacks, we can have a tricky way. We create a dummy e-mail account having no restrictions to receive e-mails. We do not forward this account to any one for the communication. As the e-mails having malicious (worm) attacks are forwarded to all the e-mail accounts at this e-mail server, so, for dummy account also. Therefore, to detect worm attacks we simply check the dummy account first because e-mails received in this account will be having maximum probability of worm attack. To save the network form such worm attacks we delete all the e-mails from all the e-mail accounts at this server. This is what the simple honeypot example.

### 1.2. Types of Honeypot

There are two factors to divide the honeypots. First is the purpose of the honeypot and second is the level of interaction with attacker. Following table-1 summarize the categorization on the basis of these two categorization factors.

On the basis of purpose of honeypots we can categorize honeypots in to two categories– production honeypots and research honeypots.

Table-1: Categorization of Honeypots

| Categorization factor | Categories of Honeypot | Brief description |
|---|---|---|
| Purpose of Honeypot | Production Honeypot | A Production honeypot is one used within an organization and help to mitigate risk. |
| | Research Honeypot | A Research honeypot is used to gain the information about the hacker's or attacker's community and does not add any direct value to the organization. |
| Level of Interaction With Attacker | Low-Interaction Honeypot | The low-interaction honeypots are the easiest to implement. Basic services such as Telnet and FTP are emulated on low interaction honeypots. |
| | Medium-Interaction Honeypot | In terms of interaction, this is a little more advanced than low-interaction honeypots, but a little less advanced than high-interaction honeypots. |
| | High-Interaction Honeypot | High-Interaction honeypots are time-consuming to design, manage and maintain. These are generally used to gather the attacker's information for analysis. Information and evidence gathered for analysis are bountiful. The goal of a high interaction honeypot is to give the attacker access to a real operating system where nothing is emulated or restricted. |

#### 1.2.1. Production honeypots

Production honeypot is the most applicable type of honeypot in the general thinking of people. A production honeypot is generally used within an organization's environment to protect the information assets of an organization and help mitigate risk [Karthik, S. *et al.* (2005)]. It has direct value because it provides immediate security to a site's production resources. Since they require less functionality then a research honeypot, they are typically easier to build and deploy. But they give less information about the attackers than research honeypots. Production honeypots make the mirror the production network of the organization, inviting attackers to interact with them in order to expose current vulnerabilities of the network. These types of honeypots add more weightage in the field of detection of the attacks rather than the prevention of attacks. Better prevention can be done with the combination of intrusion detectors, firewalls and production honeypots. Example of production honeypot is Nepenthes.

#### 1.2.2. Research Honeypot

A Research honeypot is used to gain the information about the hacker's or attacker's community and does not add any direct value to the organization [Karthik, S. *et al.* (2005)].

Their general purpose is to gather information on the general threats which an organization may face. Its primary function is to study the way of attacker's progress, understand their motives, behavior. Research honeypots are complex to both deploy and maintain but are used to capture extensive amounts of data. They can be very time consuming. They are better to learn about the attackers but have very little contribution in the direct security of an organization. They are typically used by organizations such as universities, governments, the military or large corporations interested in learning more about threats research. Examples of such research honeypots are honeynets.

Another categorization is based on the level of interaction with the attacker. There are three types of honeypots on the basis of level of interaction with the attacker.

### 1.2.3. Low-interaction honeypots

Low-interaction honeypots can be easily installed on the system and configured to any of the services such as TELNET, FTP, MESSAGING, etc. This low-interaction honeypot is both easy to deploy and maintain. But to prevent the system from being fully exploited by hackers, the administrator needs to ensure patch management on the host system and to carefully monitor the alert mechanisms. Low-interaction honeypots have the lowest level of risk. The honeypot cannot be used as a launch pad to attack other systems as there is no legitimate operating system for the hacker to interact with. The low-interaction honeypot is only good at capturing known attack patterns, but is worthless at interacting or discovering unknown attack signatures [Yamada, Y. *et al.* (2007)]. The main objective of low-interaction honeypot is only to detect, such as unauthorized probes or login attempts. Examples of Low-interaction honeypot are honeyd, mwcollect, MultiPot etc.

### 1.2.4. Medium-Interaction Honeypots

In terms of interaction, this is a little more advanced than low-interaction honeypots, but a little less advanced than high-interaction honeypots. Medium-Interaction honeypots still do not have a real operating system, but the bogus services provided are more sophisticated technically [Yamada, Y. *et al.* (2007)]. Example of Medium-interaction honeypot is Nepenthes.

### 1.2.5. High-interaction honeypots

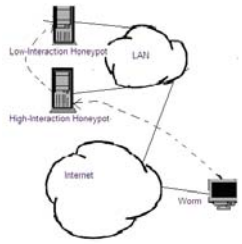


Fig. 1. Double honeypot System

These kinds of honeypots are time-consuming to design, manage and maintain. Among the three types of honeypots, this honeypot possess a huge risk. But, the information and evidence gathered for analysis are numerous. The goal of a high interaction honeypot is to give the attacker access to a real operating system where nothing is emulated or restricted [Yamada, Y. *et al.* (2007)]. An example of High-interaction honeypot is honeynets.

## 2. Double honeypot System

Yong Tang and Shigang Chen [Tang ,Y. *et al.* (2005)] gave the system architecture of double honeypot. They used the double honeypot system for detecting the new worm's attacks. They prefer to use double honeypot system to detect the worm's attack due to the following reasons-

Used honeypots are local to the intranet

System architecture is simple

Double honeypot system is having two honeypots; one is inbound or high-interaction honeypot to attract the worm's attacks, and second is outbound or low-interaction honeypot to gather the attack information. In this

architecture the principle is simple; the inbound honeypot is not authorized to establish the outbound connection. But when an attack comes to the inbound honeypot, it tries to establish the outbound connection because worms are having the self-replication property. As soon as the inbound honeypot tries to establish the outbound connection the malicious traffic is forwarded to the inbound honeypot, so that it can gather the malicious traffic. Figuer-1 shows the double honeypot system.

In this architecture the inbound honeypot can establishes the connections to the machines requested from outside (non-local to LAN). Hence it is implemented as high–interaction honeypot. But the inbound honeypot is having authorization to establish the connections for the inside requests. Hence due to the system security it is made of low-interaction type honeypot.

## 3. Extended Honeypot Framework

An Intranet of an organization may have more than one local LANs. These distinct LANs may have different vulnerable systems. These vulnerabilities of systems may be different. Hence, distinct nature worm attacks may be their on them. In this situation if we make the honeypot for whole of the Intranet then it will be too complex to implement and costly to maintain also. To reduce the complexity of the honeypot and make it simple to implement, we make the local honeypots for all LANs. In this architecture we can not tell the gathered information of one local honeypot to the other local honeypot. To resolve this shortcoming we need some means to share the information among the all local honeypots. Also, in the simple double honeypot system the efficacy

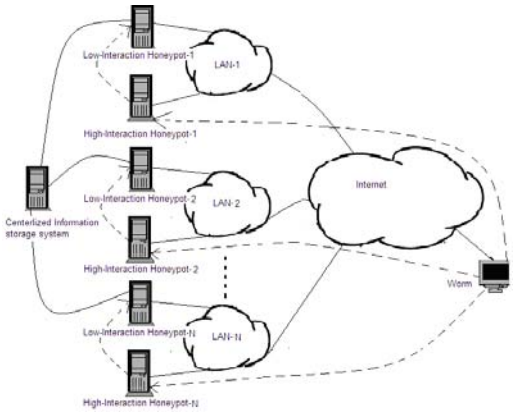


Fig. 2. Extended Honeypot Framework

of the system can be increased by sharing the information among various LANs of an Intranet. This sharing of information can be achieved by including a centralized storage system which can interact with all the low-interaction honeypots and update all the most recent attack information. Following figure-2 shows this Extended Honeypot Framework architecture for an Intranet.

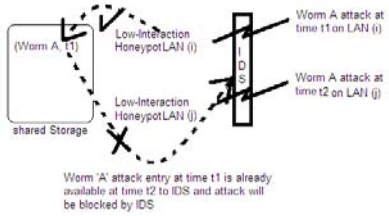


Fig. 3. Prevention Efficacy of Extended Honeypot Framework

### 3.1. Prevention Efficacy of Extended Honeypot Framework

If we consider the prevention efficacy of double honeypot and Extended Honeypot Framework, we find that the prevention efficacy of Extended Honeypot Framework is more if there are more than one LANs in an Intranet.

It can be explained by the help of figure-3 as follows- Let we assume worm 'A' attack is carried out on LAN (i) at time t1. This attack is detected by the low-interaction honeypot of LAN (i) and entry is made into the shared

storage as it is not available previously. At time t2 same worm 'A' attack is carried out on LAN (j), where t2>t1. At time t2 all the previous entries of shared storage is available to intrusion detection system (IDS) and hence, this time no need to process the functioning of honeypot. This type of situation is more helpful where more than one LANs are there in an Intranet.

### 3.2. Detection by Extended Honeypot Framework

Extended Honeypot Framework is able to detect the old as well new malicious attacks. For detecting worm attacks it uses the self-replication properties of worms. Due to self-replication property the worm tries to make an outbound connection from inbound honeypot which is not permitted to the inbound honeypot. Hence, old as well new, both types of attack can be detected from the internet traffic. Shared honeypot simply increases the speed and gives proper combination to use the well known and widely used signature-based approach to block the worm attacks.

In this, all the worm attacks carried over all the LANs of the Intranet are stored at one place and can be shared by all one IDS of the Intranet which is having low cost of detection.

## 4. Conclusion

The paper provides a brief overview of honeypots and their usage. Different types of honeypots such as production honeypots, research honeypots, Low-Interaction honeypots, medium-Interaction honeypot and high-Interaction honeypot are discussed with examples. It also proposes an extended double honeypot system to secure the Intranet having more than one LANs with less cost and more efficiency. The positive effects of extended double honeypots in the detection and prevention of malicious attacks are also discussed.

Finally, the honeypots is relatively a new technology and having good scope for future works. Honeypot can be used with other well established security tools such that IDS or Firewalls to make them more effective. Digital forensic is another field where honeypots can play an important role because honeypots technology is a good means to gather the information about malicious attacks.

## Reference List


[1] Zhou, J.; Heckman, M.; Reynolds, B.; Carlson, A.; Bishop, M. (2007). Modeling network intrusion detection alerts for correlation. ACM Transactions on Information and System Security (TISSEC), Volume 10, Issue 1, pp.-1-31.

[2] Sommer, R.; Paxson, V. (2003). Enhancing byte-level network intrusion detection signatures with context. In: Proceedings of the 10th ACM conference on Computer and Communications Security, ACM, pp. 262-271.

[3] Pouzol, J. P.; Ducass, E. M. (2002). Formal specifications of intrusion signatures and detection rules. In Proceedings of the Computer Security Foundation Workshop.

[4] Kruegel, C.; Vigna, G. (2003). Anomaly detection of web-based attacks. Proceedings of the 10th ACM conference on Computer and communications security, ACM Press, New York, NY, USA, Pages: 251 – 261.

[5] Ghosh, A. K.; Wanken, J.; Charron, F. (1998). Detecting Anomalous and Unknown Intrusions against Programs. In Proceedings of the Annual Computer Security Applications Conference (ACSAC'98), pp.-259-267, Scottsdale, AZ.

[6] Kruegel, C.; Toth, T.; Kirda, E. (2002). Service Specification Anomaly Detection for Network Intrusion Detection. In Symposium on Applied Computing (SAC). ACM Scientific Press.

[7] Swapna, S.; Gokhale; Lu, J. (2005). Anomaly Detection Based on Performance Data. Proceedings of the IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, pp.-444-445.

[8] Thottan, M.; Ji, C. (2003). Anomaly Detection in IP Networks. IEEE Trans. Signal Processing (Special issue of Signal Processing in Networking), pp. - 2191–2204.

[9] Lee, W.; Xiang, D. (2001). Information-Theoretic Measures for Anomaly Detection. In IEEE Symposium on Security and Privacy, Oakland, CA.

[10] Mahoney, M.; Chan, P. K. (2004). PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic. Florida Tech. technical report 2001-04. Available at: http://cs.fit.edu/~tr/

[11] Yamada, Y.; Katoh, T.; Bita, B. B.; Takota, T. (2007). A New Approach to Early Detection of an Unknown Worm. 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), pp. 194-198.

[12] Spitzner, L. (2003). Honeytokens: The Other Honeypot, Security Focus, 2003. Available at: http://www.securitydocs.com/library/2692 Retrieved on: 20th June, 2010.

[13] Edwards, M. J. (2005). Honeypots That Collect Malware. WindowsItpro, 2005. Available at: http://www.windowsitpro.com/Article/ArticleID/47561/47561.html Retrieved on: 15th June, 2010.

[14] Karthik, S.; Samudrala, B.; Yang, A. T. (2005). Design of Network Security Projects Using Honeypots. Journal of Computing Sciences in Colleges, Volume-20, Issue-4, pp.-282-293.

[15] Tang, Y.; Chen, S. (2005). Defending Against Internet Worms: A Signature-Based Approach. In Proceedings of IEEE INFOCOM'2005, Miami, Florida, USA, pp.1-11.