



Review Paper on HoneyPot Mechanism – the Autonomous Hybrid Solution for Enhancing

Miss.Swapnali Sundar Sadamate.

Computer System Security

Department of Computer Engineering

Padmabhooshan Vasantdada Patil Institute Of Technology

Bavdhan, Pune-21, India

Abstract- Achieving computer system security is one of the most popular and fastest Information Technology in organization. Protection of information availability, its access and data integrity are the basic security characteristics of information sources. Any disruption of these properties would result in system intrusion and the related security risk. Advanced decoy based technology called HoneyPot has a huge potential for the security community and can achieve several goals of other security technologies, which makes it almost universal. This topic is devoted to sophisticated hybrid HoneyPot with autonomous feature that allows to, based on the collected system parameters, adapt to the system of deployment. Also we are using Support Vector Machine(SVM) for intrusion detection.

Keywords- HoneyPot, IDS, Support Vector Machine(SVM), KNN, Client-Server architecture.

I. Introduction

Computer security is among one of the main areas of information technology. Over recent years mentioned area achieves the biggest progress because nobody wants that exactly his system will be attacked and intruder or anybody else will receive the stolen data. Whichever more experienced attacker can exploit weaknesses in the security system and penetrate through its defense mechanism to obtain sensitive data. It's necessary to put high priority to system security, minimize vulnerabilities and secure the computer system against intrusion. Today's standard of security is using specifically configured firewall in combination with the intrusion detection system (IDS). But using only IDS is not sufficient. We need to find out how attacker attacks actually so we will provide a security hole in system and will provide unimportant data in it. Attacker will attack in system, so that we can record all activities done by attacker that will help us to prevent actual data from these type of attackers, this technology is called as HoneyPot.

In this paper we are focusing more on identifying intrusion detection by using Support Vector Machine(SVM).

II. History

The Role of Intrusion Detection System is proposed in [1], where perspectives on intrusion like victims and attackers are focused. Today's standard of security is using specifically configured firewall in combination with the intrusion detection system (IDS).

The Simulating Networks with Honeyd is proposed in[2], in this paper Honeyd simulates virtual hosts on a network, and is actively used in HoneyNet research today. it's a thin daemon with lots of interesting features. It can assume the personality of any operating system, and can be configured to offer different TCP/IP "services" like HTTP, SMTP, SSH etc. Honeyd is used in HoneyNet research typically for setting up virtual honeypots to engage an attacker.

The Official Nmap Project Guide to Network Discovery and Security Scanning is proposed in [3], which says The Nmap Security Scanner is a free and open source utility used by millions of people for network discovery, administration, inventory, and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on a network, what services (application name and version) those hosts are offering, what operating systems they are running, what type of packet filters or firewalls are in use, and more.

Host-based Intrusion Detection System proposed in [4], it presents intrusion detection system which informs system administrator about potential intrusion incidence in a system. The designed architecture employs statistical method of data evaluation, that allows detection based on the knowledge of user activity deviation in the computer system from learned profile representing standard user behaviour.

The Definitions and Values of HoneyPots proposed in [5], it gives actual idea and some definitions of honeypots. For the purposes of this paper, they have define a honeypot as "a resource whose value is being in attacked or compromised". This means that whatever we designate as a honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited. HoneyPots are not a solution, they do not 'fix' anything.

Design of Network Security Projects Using HoneyPots is described in paper [6], here they advocate the use of honeypots as an effective educational tool to study issues in network security. They support this claim by demonstrating a set of projects that they have carried out in a network, which they have deployed specifically for running distributed computer

security projects. The design of their projects tackles the challenges in installing a honeypot in academic institution, by not intruding on the campus network while providing secure access to the Internet. In addition to a classification of honeypots, they present a framework for designing assignments/projects for network security courses. The three sample honeypot projects discussed in this paper are presented as examples of the framework. There are three Honeypots subtypes depending on the level of integration: Low-interaction, Medium-interaction, High-interaction.

Algorithms:

There are two algorithms which we are going to use in this paper to improve intrusion detection.

KNN (K-nearest neighbor)

Method proposed in this project for attack type detection uses KNN. This is used to classify program behaviour as normal and intrusive. Short sequences of system calls are used to characterise program's normal behaviour. Here frequencies of system calls are used to describe program's behaviour. In KNN text categorization is the process of grouping text documents into one or more predefined categories based on their content. Here number of statistical classification and machine learning techniques have been applied to text categorization, including regression models, decision trees, nearest neighbour classifiers, neural networks and support machine vector(SVM). The first step in text categorization is to transform documents, which typically strings of characters, into a representation suitable for learning algorithm and the classification task. The most commonly used document representation is so called vector space model. In this model, documents are represented by vectors of words.

For each training example $\langle x, f(x) \rangle$, add the example to the list of training examples.

Given a query instance x_q to be classified, Let x_1, x_2, \dots, x_k denote the k instances from Training examples that are nearest to x_q . Return the class that represents the maximum of the k instances.

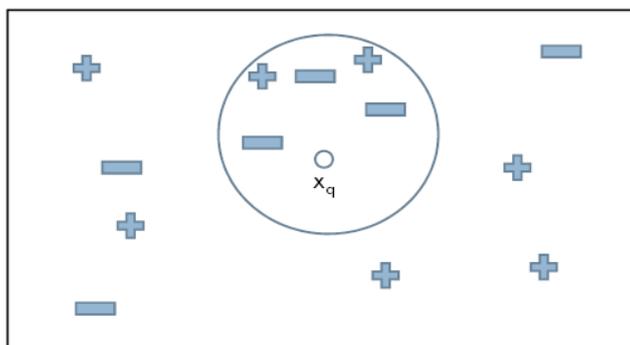


Fig. 1 KNN process-instance x_q and its nearest instances

If $K = 5$, then in this case query instance x_q will be classified as negative since three of its nearest neighbors are classified as negative.

Support Machine Vector (SVM)

Support Vector Machines (SVM) are the classifiers which were originally designed for binary classification. The classification applications can solve multi-class problems. Decision-tree-based support vector machine which combines support vector machines and decision tree can be an effective way for solving multi-class problems. This method can decrease the training and testing time, increasing the efficiency of the system. The different ways to construct the binary trees divides the data set into two subsets from root to the leaf until every subset consists of only one class. The construction order of binary tree has great influence on the classification performance. In this paper we are using an algorithm, Tree structured multiclass SVM, which has been used for classifying data. This work proposes the decision tree based algorithm to construct multiclass intrusion detection system.

If binary SVMs are combined with decision trees, we can have multiclass SVMs, which can classify the four types of attacks, Probing, DoS, U2R, R2L attacks and Normal data, and can prepare five classes for anomaly detection.

This paper's aim is to improve the training time, testing time and accuracy of IDS using the hybrid approach.

III. Overview

Sophisticated Hybrid Honeypot In IDS Security Architecture

The proposed architecture uses a sophisticated hybrid Honeypot with an autonomous feature as an IDS detection mechanism. Solution for minimizing failures in the detection process and collection of important data based on Honeypot consists of a combination of security tools: Snort IDS, Sebek and Dionaea. Tools were selected based on their properties analyzed above.

The detection mechanism based on a sophisticated hybrid Honeypot integrated in the client-server architecture consisting of centralized main server and multiple client stations. Client workstations serve to capture suspicious activity or directly record the malicious code which is then send to server for processing. Server analyzes received data, decides to issue or not to issue a security warning and displays cumulative information through a web interface. This proposal aims to provide a solution of early warning against any attack on the computer system.

Server Architecture

Due to the centralization of collected data is main server at the same time connected to multiple clients and is set to receive all incoming messages which are then stored in the knowledge database. Cohesion of individual reports indicated attackers' intention to attack aimed computer system areas with widespread attacks or full-range scanning. The proposed server architecture (Fig. 2) consists of three main parts, which data are normalized before storing to the database:

- Sebek server – at the same time receives and filters several data sources representing instructions or a connection to incoming data storing process.
- Dionaea server – accepts patterns of malicious code that sends the dionaea client part.
- Verification process – a modular scheme of hybrid open-source system for intrusion detection. It's using standard communication format. It can be adapted to the needs of an extensive system from any point of deployment, receives the amount of data from clients and integrates diversified data formats. Web-server interface displays all information about captured attack.

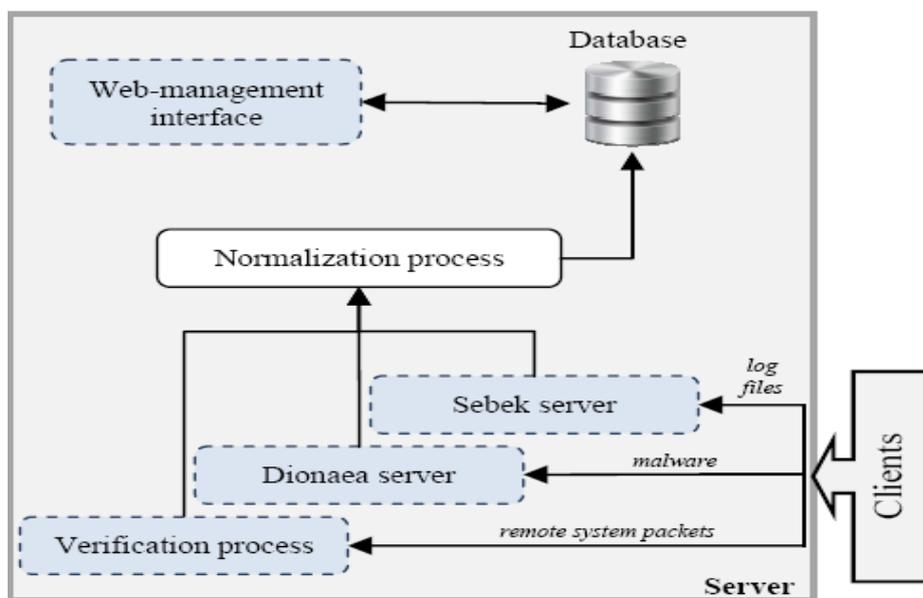


Fig. 2 Server architecture

Client Architecture

Because of gathering data about attacker activities during an attack are installed clients placed in the same domain. Various parts of the system are independently activated for collecting set of data depending on attack type. Obtained data are subsequently backward delivered to a server to facilitate further analysis and for the subsequent updating system security. Client architecture (Fig. 3) consists of three components/tools:

- Sebek client – records attacker behaviour during interaction with the Honeypots in log files.
- Dionaea client – attracts attackers and captures the patterns of malware by simulating basic system services and vulnerabilities.
- Snort – monitors and filters packets during detecting intrusions, Identifies patterns of separate attacks, information and warning messages.

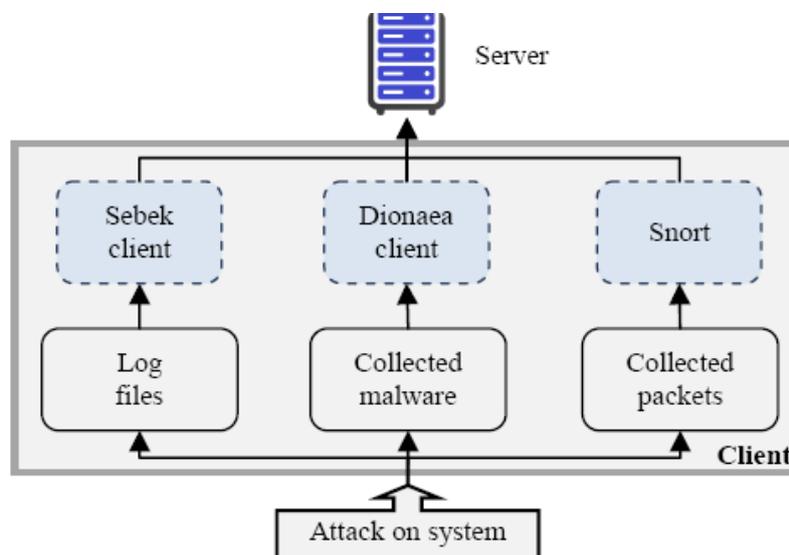


Fig. 3 Client architecture

Sophisticated Hybrid Honeypot

Proposed Honeypot is deployed as a separate device physically connected to the computer system. Using the passive fingerprinting method determines the system of deployment individual parameters – number and type of OS, running services, hosts communications. Right after obtaining all the necessary information can begin deploying of Honeyspots (Fig. 4), which are designed to system mirroring – created Honeyspots merge with the environment, making them more difficult to identify and reveal. Proposed sophisticated Honeypot significantly facilitates the configuration, administration and maintenance process.

Deployment of multiple physical Honeyspots constitutes considerable workload and associated costs. Simpler solution is to use virtual Honeyspots that counteract with their characteristics to physical decoys.

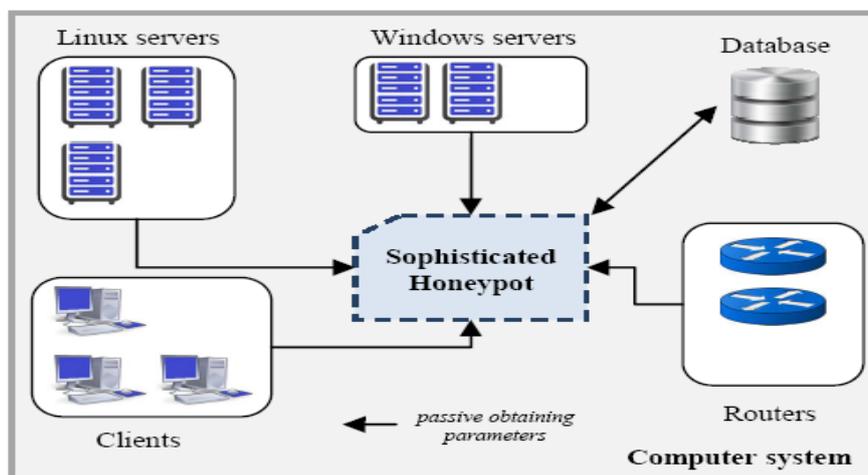


Fig. 4 Obtaining system parameters during determination process of deployment via passive method

They can also monitor unused IP address space as the system itself. Proposed sophisticated Honeypot (Fig. 4) is responsible for creating virtual Honeyspots in computer system. The ability to deploy and set on virtual Honeyspots in the whole system of deployment is implemented using existing open-source solution called Honeyd.

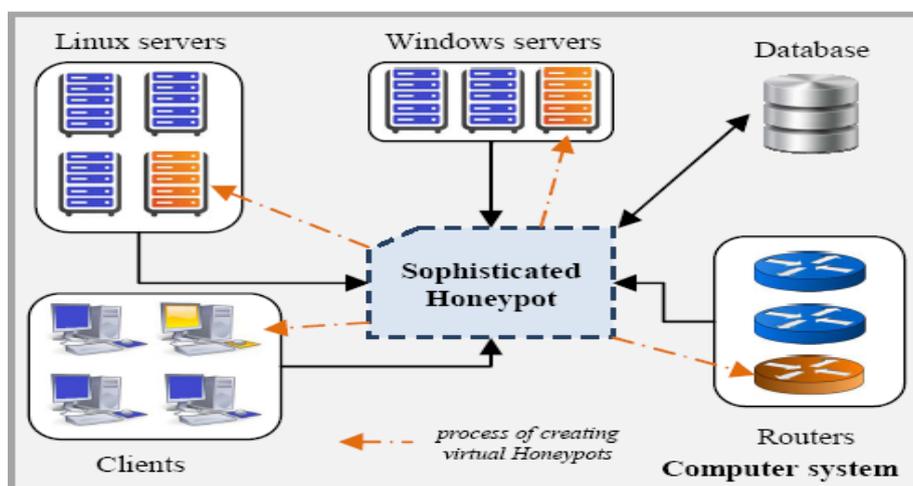


Fig. 5 Honeypot deployment process upon acquired parameters

Advantages

Minimal resources– captures only malicious activity in the system. For its functionality is enough an equipment with low end system parameters.

- Simplicity – Honeyspots are simple and flexible. For their functionality they do not require complicated algorithms or other complex operations.
- Discovering new tools & tactics – capture and record everything that interacts with them.
- Small data sets – produces small amount of data, but it can be a high quality.
- Reduce false positives and false negatives.

Disadvantages

- Risk of takeover – after gaining control over the Honeypot attacker can retrieve all the collected data.
- Disclosure of identity – Honeypot has expected characteristics and behaviour. Experienced attacker can detect presence of incorrectly configured decoy in system.

IV. Conclusion

Honeypots becoming highly-flexible solution, Not only their deployment and management become more cost-effective, but also provide a much better integration into the system, thereby minimizing the risk of human error during manual configuration. Merger with the surrounding system in addition minimizes the risk of identification by attackers. Just as all new technology, the decoys also have some shortcomings that need to be overcome and eliminated. Honeypot is excellent security tool but it is not a panacea for securing the whole system. The apart of this work is improving the IDS detection mechanism and minimizing the number of generated false positives and also false negatives using advanced technology called Honeypot. The work includes proposal of an autonomous special safety feature by using KNN algorithm for detecting attack type and by using SVM for intrusion detection for enhancing security of distributed computer systems. Unique proposal combines a variety of security tools, to order to minimize their disadvantages and maximize the security capabilities in the process of intrusion detection.

References

- [1] J. McHugh, A. Christie, J. Allen, "Defending Yourself: The Role of Intrusion Detection System," IEEE Software, IEEE Computer Society, pp. 42-51, October 2000.
- [2] R. Chandran, S. Pakala, "Simulating Network with Honeyd,"[online], Technical Paper, Paladion Networks, December 2003. Available on: <http://www.paladion.net/papers/simulating_networks_with_honeyd.pdf>.
- [3] F. G. Lyon, "Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning," [online], Nmap Project, USA, ISBN 978-0979958717, January 2009. Available on: <<http://nmap.org/book>>.
- [4] L. Vokorokos, A. Baláz, "Host-based Intrusion Detection System," IEEE 14th International Conference on Intelligent Engineering Systems, Budapest, pp. 43-47, ISBN 978-1-4244-7651-0, 2010.
- [5] L. Spitzner, "The Value of Honeypots, Part One: Definitions and Values of Honeypots," Security Focus, 2001.
- [6] L. Spitzner, "Honeypots: Tracking Hackers," Boston, USA: Addison-Wesley, Parson Education, ISBN 0 321-10895-7, 2003.
- [7] S. Karthik, B. Samudrala, A. T. Yang, "Design of Network Security Projects Using Honeypots," Journal of Computing Sciences in Colleges, 2004.
- [8] R. Baumann, C. Plattner, "White Paper: Honeypots," Swiss Federal Institute of Technology, Zurich, 2002.
- [9] N. Provos, "Developments of the Honeyd Virtual Honeypot,"[online]. Available on: <<http://www.honeyd.org>>.
- [10] A. Baláz, N. Ádám, "Intrusion Detection System Using Multilayer Perceptron," 6th PhD Student Conference and Scientific and Technical Competition of Students of FEI Technical University of Košice, pp. 13-14, ISBN 8080860351, 2006.
- [11] Snort [online]. Available on: <<http://www.snort.org>>.
- [12] Sebek [online]. Available on: <<http://www.honeynet.org/tools/sebek/>>.
- [13] Dionaea catches bug [online]. Available on: <<http://dionaea.carnivore.it/>>.
- [14] E. Danková et al., "An Anomaly-Based Intrusion Detection System," Electrical Engineering and Informatics 2, Košice, ISBN 978-80-553-0611-7, 2011.
- [15] Securing WMN using hybrid honeypot system author Rawat, Paramjeet; Goel, Sakshi; Agarwal, Megha; Singh, Ruby PUB. DATE May 2012 SOURCE International Journal of Distributed & Parallel Systems; May 2012, Vol. 3 Issue 3, p29.
- [16] distributed web honeypots, Ryan Barnett, Christian Bockermann (<http://www.jwall.org>), Lukasz Juszczak (CERT Polska), Josh Zlatin (Pure Hacking), Tony Carter (iTrustlabs), Steeve Barbeau. 2013.