International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue 2nd International Online Conference on Advent Trends in Engineering, Science and Technology "ICATEST 2016", 03 April 2016

Dynamic Honeypot Security with Deceptive Virtual Host

Miss Aarti G Mantri
Department of Computer Science and Engineering
Prof. Ram Meghe College of Engineering and Management, Amravati, India
mantriaarti21@gmail.com , 8600736938

Abstract: A compromised control system could have security, public safety, industrial or economical consequences. In most contexts, they are rarely capable of providing perfect intrusion detection. Deceptive systems called Honeypot that emulate critical network entities have been deployed. [1]. Self-configuring honeypots that passively examines control system network traffic. A collaborative use of dynamic virtual honeypots in control system network is introduced [2]. Aspects of effective tools identifying network host characteristics are examined. The idea was based on automatically configuring a honeypot by gathering information gleaned from network traffic [4]. One of the most common approaches is anomaly detection.

Keywords: Honeypot, network security, intrusion detection

1. INTRODUCTION

Modern complex control systems are interconnected via Ethernet networks. These networks, found deployed in areas such as chemical facilities or energy production, are utilized to deliver status and control information which is vital to the operation of physical systems. In this system, the collaborative use of dynamic virtual honeypots in a control system network is introduced [2]. The presented algorithm focuses on automatically managing the complexity of self-configurable dynamic virtual hosts (DVH) by adapting to an active network environment.

Honeypot is a non-production system, used for exploiting the attacker and notice the attacking techniques and actions. In network security, honeypots are used to detect the attackers and learn from their attacks and then modify and develop the system accordingly for security. In Static honeypots are unable to represent the dynamic nature of today's networks dynamic honeypots are different numbers of hardware devices and hosts running various operating systems are online at a particular time and frequently join and leave a network. A single static server honeypot presents a particular operating system with unique address. Due to static nature of this honeypots cyber devices are attack this system, so dynamic honeypots are always beneficial to prevent from attack. Dynamic honeypots overcome the static nature of server honeypots by automatically adjusting the number of hosts, operating systems and running services of honeypots deployed in a network environment, based on the topology of the production network. Honeypot system takes the captured packet as input to the system and processes this packet to obtain the details of the particular system from where the packet is originated [4].

2. PROPOSED APPROACHES

In the proposed systems, described the software tools evaluation and implementation logic of the solution. The key functional areas:

1) Network entity identification (NEI);

2) DVH configuration;

3) Virtual host instantiation (VHI).

These act in a continuous cycle of processing and updating information represented by the dotted line box. A pseudo code of the algorithm implementation details for procedures in italics found afterward in each section

1) Network Entity Identification:-

Component of NEI monitors network traffic from which it get the source, destination, and port activity Information from the NEI is delivered to an implementation of the logic tasked with creating a DHP configuration. An evaluation was conducted on six passive network information gathering open source tools. The critically required capabilities examined were OS identification, port or service identification per host, and the capture of media access control addresses with a resolution to the appropriate vendor. A pseudo code of the algorithm is shown in Fig.

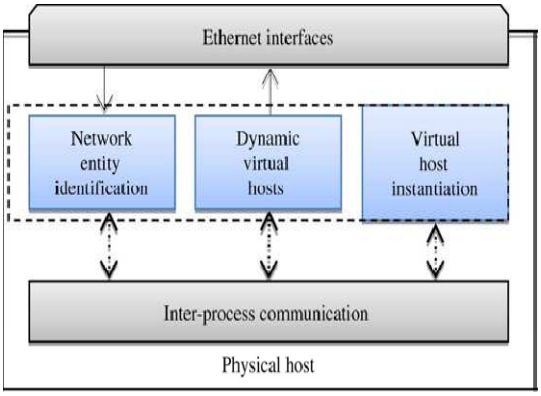

Fig. 1 : (Conceptual Design)

2) Dynamic Virtual Hosts:-In DVH hosts emulate the network signature of actual systems on a physical network. Honeyd is a popular open source solution for virtual honeypots that provides a flexible and feature rich configuration capability. Ultimate goal is the automatic configuration and dynamic update of a variable length list of virtual hosts based on information gathered from actual hosts using Ettercap. Following are areas of DVH

2.1) OS Selection:-For any given host on a network, may not be able to identify the operating system. If this occurs, for an emulation target, then an OS must be chosen. It is desirable to provide an exact match in network behavior.

```
Create and update virtual hosts with following:
   Network Entity Identification.
   Write entities to XML.

   Read_data; from input files
   For each IP create a Dynamic Virtual Host
      Find_closest representative OS.
      Map_OS values to Honeyd names
      Create_MAC address for new hosts
      Create_Features for device specific behaviors
      Create_Config for virtual hosts
   End
```

Fig. 2 : (Pseudo code)

2.2) OS Name mapping:-The algorithm's initial pass compares the word tokens of the OS names, looking for case-insensitive string matches. The number of word matches were summed and stored. After iterating through each possible OS combination, the one with the largest count total is presented as a candidate. Finally, each OS name combination is written to a file for reference during creation of the configuration.

2.3) MAC Creation:-In order to create a new MAC address that appears to come from a specific vendor, these first three octets were used. In the Create_MAC function, the last three octets are randomly generated and appended to the end of the captured candidate vendor portion. This new MAC is then compared with all other MACs noted in the host list O. For instance, if port security is enabled on the network switch the possible MACs would have to be predefined.

2.4) Network Service Emulation:-The host entries in contain network ports, previously defined as, that were active during the capture session. Along with the port number, a port service name is available. This service name is a human readable text value that is defined in a configuration file.Services. Utilizing the service names contained in this file, a new configuration file called serv.conf was created. This file maps the service name to a service emulation script path.

3) VHI and Update:-The candidate emulation hosts are provided at startup as a list of IP addresses. An initial configuration file is created by Create_Host_Conf Changes to the configuration of the virtual hosts running under Honeyd are performed while the system is running. The resulting XML output is saved and compared to an existing output file. A companion Honeyd executable file, called Honeydctl, provides this functionality. A simple example Honeyd configuration file containing one virtual host configuration in below fig.3

```
create vh1
set vh1 personality "Linux 2.4.xx"
set vh1 default tcp action reset
set vh1 default udp action reset
set vh1 default icmp action reset
add vh1 tcp port 23 "/script/router-telnet.pl"
set vh1 ethernet "00:00:BC:A1:00:23"
bind 192.168.1.125 vh1
```

Fig. 3 : (Honeyd host Configuration)

3) FUTURE ASPECTS

This work has identified several areas of possible future research. The use of virtualized networks and devices derived from the automated system presented could subsequently be used as a standard test bed for a variety of IDS systems. Finally, service emulation scripts are created. Autonomously developing service behaviours that emulate observed network communications would further the goals of deception and IDS testing

4) CONCLUSION

An algorithm was proposed and demonstrated to automatically deploy deceptive

virtual network entities in a control system network. Six open source passive network-monitoring tools were evaluated and Ettercap was chosen for host identification. Finally, service emulation scripts are manually created.

Several deficiencies with both the monitoring tools and virtual honeypot implementation Honeyd were discovered and discussed. These problems are non-IP based traffic, OS identification database support, missing information, and well formatted program output.

REFERENCES

[1]. Carcano et al., "A multidimenisonal critical state analysis for detecting intrusions in SCADA systems," IEEE Trans. Ind. Informat., vol. 7, no. 2, pp. 179–186, May 2011

[2]. N. Provos and T. Holz, Virtual Honeypots. Reading, MA, USA: Addison-Wesley, 2007.

[3]. Hecker, K. L. Nance, and B. Hay, "Dynamic honeypot construction,"in Proc. 10th Coll. Inf. Syst. Secur. Educ., Adelphi, MD, USA, 2006,pp. 4880–4889.

[4]. X. Jiang and D. Xu.(2004). BAIT-TRAP: A Catering Honeypot Framework [Online]. Available: http: //citeseerx.ist.psu.edu/view doc/download doi=10.1.1.84.3

[5]. J. Hieb and J. H. Graham, "Anomaly-based intrusion detection for network monitoring using a dynamic honeypot," Intell Syst. Res. Lab., Univ.Louisville, Louisville, KY, TR-ISRL-04–03, Dec. 2004

[6]. M. A. McQueen and W. F. Boyer, "Deception used for cyber defense of control systems," in Proc. 2nd IEEE Conf. Human Syst. Interact. Catania, Italy, May 2009, pp. 624–631.