International Journal of Power Control Signal and Computation(IJPCSC)
Vol3. No1. Jan-Mar 2012 ISSN: 0976-268X
www.ijcns.com

A Context Based Honeypot Deployment Strategy

Sivachandiran.S, Assistant Professor,
Dept of Computer Science,
Saradha Gangadharan College, Puducherry.
sivachandiran.s@gmail.com

Rajeshkumar.M, Assistant Professor,
Dept of Computer Application,
Mailam Engineering College, Mailam
Raj_win123@yahoo.com

Abstract

An entrepreneur is always looking for simple and effective ways to make their company networks more secure and resilient from attackers. It is a great thing to proactively address problems before they become mountains. Honeypots are a versatile tool for a security practitioner. Of course, they are tools that are meant to be attacked or interacted with to gain more information about attackers. One way to 'sweeten' a private network is with a honeypot that sends alerts when worm and virus infected machines are crawling the network; those nastier are looking for another vulnerable box to infect. The contribution of this paper is that it presents an organizational framework for analyzing a variety of deployment strategies and for further studying deployment of honeypots in the enterprise networks.

Keywords: **Honeypots, Honeyd, IDS, Honeynet**

1. INTRODUCTION

In computer security, a honeypot is a tool used to lure attackers and analyze their behavior in the Internet. It seems a contradiction, as the ordinary function of security tools is precisely the opposite: to keep attackers away and prevent their attacks. However, since a few years ago, honeypots are used to draw attackers into a controlled environment and attempt to know more details about how they carry out their attacks, and even to find out new vulnerabilities. A honeypot is a deception trap, designed to entice an attacker into attempting to compromise the information systems in an organization. If deployed correctly, a honeypot can serve as an early-warning and advanced security surveillance tool, minimizing the risks from attacks on IT systems and networks. Honeypots can also analyze the ways in which attackers try to compromise an information system, providing valuable insight into potential system loopholes.

2. WHY HONEYPOTS ARE NEEDED?

A honeypot is a decoy, put out on a network as bait to lure attackers. Honeypots are typically virtual machines, designed to emulate real machines, feigning or creating the appearance of running full services and applications, with open ports that might be found on a typical system or server on a network. A honeypot works by fooling attackers into believing it is a legitimate system; they attack the system without knowing that they are being observed covertly. When an attacker attempts to compromise a honeypot, attack-related information, such as the IP address of the attacker, will be collected. This activity done by the attacker provides valuable information and analysis on attacking techniques, allowing system administrators to "trace back" to the source of attack if required.

3. RELATED WORK

3.1. What is an IDS?

Intrusion detection is a set of techniques and methods that are used to detect suspicious activity both at the network and host level. Intrusion detection systems fall into two basic categories: signature-based intrusion detection systems and anomaly detection systems. Intruders have signatures, like computer viruses, that can be detected using software. You try to find data packets that contain any known intrusion-related signatures or anomalies related to Internet protocols. Based upon a set of signatures and rules, the detection system is able to find and log suspicious activity and generate alerts. Anomaly-based intrusion detection usually depends on packet anomalies present in protocol header parts. In some cases these methods produce better results compared to signature-based IDS. Usually an intrusion detection system captures data from the network and applies its rules to that data or detects anomalies in it. Snort is primarily a rule-based IDS, however input plug-ins are present to detect Anomalies in protocol headers.

3.2. Why are Honeypots important?

Honey pots are systems used to lure hackers by exposing known vulnerabilities deliberately. Once a hacker finds a honey pot, it

is more likely that the hacker will stick around for some time. During this time you can log hacker activities to find out his/her actions and techniques. Once you know these techniques, you can use this information later on to harden security on your actual servers. There are different ways to build and place honey pots. The honey pot should have common services running on it. These common services include Telnet server (port 23), Hyper Text Transfer Protocol (HTTP) server (port 80), File Transfer Protocol (FTP) server (port 21) and so on. You should place the honey pot somewhere close to your production server so that the hackers can easily take it for a real server. For example, if your production servers have Internet Protocol (IP) addresses 192.168.10.21 and 192.168.10.23, you can assign an IP address of 192.168.10.22 to the honey pot. You can also configure your firewall and/or router to redirect traffic on some ports to a honey pot where the intruder thinks that he/she is connecting to a real server. You should be careful in creating an alert mechanism so that when your honey pot is compromised, you are notified immediately. It is a good idea to keep log files on some other machine so that when the honey pot is compromised, the hacker does not have the ability to delete these files.

3.3. Success of Honeypot implementation

The first success lies on honeypot is Data Capture. It concerns information gathering. All information that enters or leaves the Honeynet must be collected for analysis. This data must be collected without the knowledge of the individuals who are conducting malicious activity against the network that is to be protected. This is to prevent the hacker from bypassing the Honeynet network. The data that is collected must be stored in a location different from the Honeynet. This is done so that if the hacker compromises a Honeynet system, the data cannot be destroyed or altered. The goal is to be able to capture data on the hacker without the hacker knowing that this data is being collected.

The Second success of honeypot lies in Data Control. It concerns protecting other Networks from being attacked and compromised by computers on the Honeynet. If a hacker compromises a Honeynet system, then this hacker must be prevented from using this system to attack and compromise production systems on other networks. The process of Data Control must be automated to prevent the hacker From getting suspicious. We do not want the hacker to become aware of the fact that the system he has compromised is on a Honeynet.

3.4. Where to deploy Honeynet in the Enterprise

There are three options of where one can place their Honeynet in the network with respect to other networking devices and these are either externally facing the Internet, Internally behind the firewall or in the DMZ zone

- External Placement
- Internal Placement
- DMZ Placement

Each of these locations has their own advantages and disadvantages and the choice depends on factors like amount of network resources the company has, the objectives of deploying a honeynet e.g. a research based institution that wants to capture as much hacker details as possible will likely put their honeynet externally, whereas an organization that wants to have up-to-date details of the possible exploits to their production systems will likely place their honeynet internally. The expertise of the administrators can also be a factor as some deployments are more complex than others. Also note that if a Honeynet Project honeywall is placed in front of the honeypot in any of these situations it can improve both the data control and data capture.

4. DEPLOYMENT STRATEGIES

Honeypot can be deployed in to two major categories:

- ✓ Production
- ✓ Research.

A. Production Honeypot are low–interaction honeypots which have little or no
Interaction with the attacker or intruder in context. Also, they have less value to security of production resources. They try to create as less a realistic environment as possible i.e. when they are deployed they not necessarily emulate the whole system as a whole but try to emulate as much as possible within certain time and value. Once deployed they serve very little purpose they capture data. In essence they just act as a basic event log, with a potential difference that they are not meant to be interacted with. For example, if you want to monitor web-based attacks, you just emulate a basic web server like Apache and listen to port 80(usually HTTP)

connections. Once this is done, all the connections that scan the honeypots for HTTP vulnerabilities will be logged. Production honeypots are made for mainly this reason; they capture data and send it to administrators. How they utilize this data and what precautions they take is left on to them. This has so many advantages compared to competing technologies like Intrusion detection systems and firewalls. A honeypot has no production value i.e. they do not act as servers and so are not meant to be interacted with. If any probe or access comes on it, it is most likely a malicious activity, unless there has been a Misconfiguration by the administrator or someone has mistakenly accessed the wrong system.

Also, production honeypots often are used to deceive people as legitimate servers. An intruder might think he/she is interacting with the real system while they are just attacking a honeypot. A recent example of this was cited in one of the large security firms. – Internet Security Systems (ISS). According to the firm, one of the web-server that suffered a breach and got Defaced was just a honeypot and was meant to get hacked. However, the article said the "X-force Internet Watch" was then properly monitored and the malware was removed.

B. Research Honeypot are more complex than production honeypots and are kept in more secure environment since they do not comparatively have much valuable assets to protect in the backbone. However, they simulate the whole operating system and thus present the intruder with a known set of Vulnerabilities within the system. For example, for web attacks a default installation of Linux 3.1 with Apache 1.1 can be installed and the results observed.
Since, research honeypots are a step ahead than production ones, they naturally get the backward compatibility and advantages. Thus, all the advantages of production honeypots are present in research ones. Also, they are more stringent in their deployment and can serve response tasks like trace-back. Security firms might be interested in finding new attack tools and Trends and thus keep their eye on research honeypots. However, law enforcement agencies and government look more for early warnings and prediction from the analysis of research honeypots. These are just a few examples to cite for the significance of research honeypots to security circles. For example, a recent result from the Honeynet Project revealed a vast increase in organized credit card fraud. According to it a vast majority of stolen credit cards are used across relay channels thus increasing the illicit use of credit cards, performing identity thefts, compromising merchant sites and exchanging of these numbers.

5. EXISTING WORK

In this section we survey some notable deployment strategies developed for or used in security settings. For each of the system, if applicable, we identify and analyze the underlying various context. The existing deployment strategies can be classified into five categories: based on their circumstance approach.

Sacrificial Lamb

These systems are just placed on the network so that they can be compromised. They have no connections to the production network and just act as perfect dummy services. The idea behind this strategy is to quench the thirst of the attackers. In simple terms, give the attackers what they want, and let them play with it. These techniques necessarily developed from Clifford Stoll's publication of his encounter with a German hacker. Although his idea was just to stall the hacker so that he can track him to his root. These systems just sit there on entry points and serve with no Production value. Even data gathered within it may not be used by administrators to prevent future attacks.

Deception ports on production systems

These honeypots first 'observe' the operating system they reside on and then portray these services according to that. Honeyd is a common example of these sorts of honeypot. Also, specter is a feature-rich edition to this kind of honeypots strategy. The basic idea is deception so that the adversaries are just 'stuck-up' in solving the deception while they can either be knocked down from the network or suitable measures like trace-back, forensics can be taken. Also, various homemade honeypots use this technique, as this seems to be the most common and less-liability-shared strategy to adopt.

Proximity Decoys

These honeypot is part of the same Subnet the main servers are included in it becomes part of your own network and you are

allowed to monitor activities pertaining to your network. Also, once they are in proximity to other production systems you have ease in either re-routing traffic once some malicious attack is detected on the production systems, or trapping that attack. This helps in non-proliferation of worms, viruses as well.

Redirection shield

In this deployment by using port redirection or re-routing the traffic, honeypots can be said as acting in place of production systems. More precisely, it can be said that honeypots are just on the network to protect the production servers in case of attack. Thus, it can be legally argued that honeypots are just a layer of defense in order to protect the production systems. Also commercially, if rerouting switches are installed on client sites, honeypots while sitting at any remote system across the world can serve as services instead of just a device. Once this is done the client can be charged either based on attacks – which in any open huge corporate network would be enormous, or based on time length contract basis.

Minefield

These types of honeypots are placed just at the perimeter so that any scans or vulnerability detectors can just exploit the contents of honeypots, sparing the production servers. Also, once attacks or scans are recognized suitable alerts can be raised in order to mitigate them. Thus, honeypots just act as third layer of defense in these types of deployment. Also, this does not mean singular honeypots but even multiple honeypots if deployed can serve as means to trap, deceive, trace, tear down, or tar pit the attackers.

A. Virtual Network

VMware Workstation 8.0 software was installed on the honeypot. VMware offers the possibility to install so called virtual networks, a flexible way to interconnect virtual machines, host computers and network devices on host computers. Each of the configured virtual machines on the honeypot could then be connected to one of those virtual networks. Virtual machines not connected to the same virtual network have no possibility to "see" traffic for other virtual machines. The virtual networks are totally transparent to the virtual machines, from their viewpoint they are directly connected to a physical network.

B. Snort Configuration

Running Snort seemed a good and especially cheap solution. At the time of the development, version 2.9 was the latest available Snort development. Snort checks the network traffic on all inbound interfaces, which are eth0 for incoming traffic and eth1 and eth2 for the forwarded traffic from the IP tunnel for each interface, a Snort instance is started with its own configuration file. Snort also logs all connections as well as every network packet to the file system. Through this attempt, the possibility to check the network traffic at a later time is given.

C. Connection Logging

IPTables that are running on the network bridge store all the active connection details. Even IPTables lists the connections after they have been closed for about 60 seconds, which makes it easy to catch them for inserting into the database. Perl was used as the scripting language as it is very easy to access a database. This script is called and polls the IPTables connection tracking file once every 30 seconds for new or changed connections. All newly found connections are inserted into a data structure which resides on the local memory as well as into the database. All found connections are compared to the locally stored connections. As soon as a connection is marked as closed or doesn't exist anymore, the end time is written into the database and the local data structure gets deleted. Through this attempt, database access is minimized but logging of all connections is ensured even in a crash.

D. Logging inside the Virtual Machines at the Kernel Level

Virtual machines on the honeypot were all equipped with Linux operating systems. A Linux kernel patch was developed. This patch allows the logging of all data which passes through so-called pseudo- terminals. Pseudo-terminals (pty's) are allocated whenever a daemon starts an interactive session with a remote user. The logging of pty's has the advantage that no user mode programs (*bash*, for instance) have to be modified. Also, even not-yet-developed remote shell applications will be covered by this patch. It does not matter which encryption scheme the attacker uses for the

connection, since logging is done at the pty level after the decryption, directly in the kernel. The mechanism can only be circumvented by installing a new, unpatched kernel on the honeypot.

E. Ethernet Tunneling

Using TAP (4) software we can achieve the Ethernet tunneling. This TAP (4) can act as server or client. The server listens on a freely configurable port for an incoming TCP connection of a client program. After the successful establishment of the connection, no difference between the server and client end of the tunnel can be determined. The TCP connection between the client and server is encrypted with the RC4 crypto algorithm, and the client and server side do have to authenticate themselves during the connection setup. The authentication step and the encryption are not implemented in a totally bullet proof manner their purpose was only to not let others see what is flowing through the tunnel. Implementing perfect tunneling software was not a target.

6. RECENT TRENDS AND FUTURE DIRECTIONS

In this paper we evaluated the major works in the field of honeypot deployment, especially focusing on the context based deployment in the enterprise. A distant view in this area might be to employ advanced security for the further studying honeypot based security principles.

7. CONCLUSION

Network Security has span diverse disciplines from business to research. Currently there are quite number or research has been conducted based on honeypot security. Honeypot are new technology which is to be using for to give the network security from threats for organizations This paper gives an organizational framework for analyzing various honeypot deployment strategies and for further deploying honeypots.

REFERENCES

[1]. Edward G. Amoroso *Intrusion detection: An introduction to Internet surveillance, correlation, trace-back, traps and response.* Addison – Wiley Publications

[2]. Bakos George and Beale Jay (2001) *Honeypot advantages and disadvantages* Honeypot best practices Seminar at Dartmouth College, Hanover, New Hampshire.

[3]. Benett Jeremy (2002) *Deploying Deception* – seminar on cybersecurity, Feltham UK Recourse Technologies (now acquired by Symantec).

[4]. Gubbels Kecia (2002) *Hands in the Honeypot* SANS research paper on Honeyd – the virtual honeypot. Link: http://www.sans.org/rr/papers/30/365.pdf.

[5]. Haig Leigh (2002) *LaBrea – a new approach top securing our networks* http://www.sans.org/rr/paper.php?id=36

[6]. Homepage for Deception Toolkit (DTK) The first homemade honeypot in context by Fred Cohen
Link: http://www.all.net/dtk/dtk.html

[7]. Honeyd homepage Neils Provos the maker of honeyd - A virtual honeynet for gathering information by re-routing malicious traffic
Link:
http://www.citi.umich.edu/u/provos/honeyd/

[8]. Honeynet Project Lance Spitzner
Know Your Enemy series I, II and III – revealing the security tools, tactics and motives of the blackhat community Addison –Wesley 2000

[9]. Hydan – Rakan Al-Khalil (2003)
A groundbreaking program that hides messages within executables. The researcher is a student at Columbia university NY, USALink: http://www.crazyboy.com/hydan/

[10]. Mantrap or Symantec Decoy Server honeypage A high interaction commercial honeypot by Symantec
Link:
http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=157

[11]. Liston Tom *LaBrea – The 'sticky honeypot' and IDS* Honeypot that tar-pits hackers for indefinite time. Link: http://labrea.sourceforge.net/labrea-info.html

[12]. Lipson Howard F. (2002) *Tracking and tracing cyber attacks: technical challenges and global policy issues* – CERT report on traceback Link: http://www.cert.org/archive/pdf/02sr009.pdf

[13]. Magalhaes Ricky M. (2003)
Understanding Virtual honeynets – an article in Windowsecurity.com
Link:
http://www.windowsecurity.com/articles/Understanding_Virtual_Honeynets.html