Data Security Using Honeypot

Rohit Upadhayay, Tushar Kanti Mandal, Sumit Joshi, Manish Kala
Student, CSE, Dronacharya Group of Institution, Greater Noida, Uttar Pradesh, India

***Abstract*— With an increase in the use of the internet, there has been a rise in the number of attacks on servers. These attacks can be successfully defended against using security technologies such as firewalls, IDS and anti-virus software, so attackers have developed new methods to spread their malicious code by using web pages, which can affect many more victims than the traditional approach. Honeypot is an new technology with enormous potential for security communities. Honey pot offers a wealth of features that can assist with intelligence data gathering, incident response for a better understanding of who the attacker is, what method the attacker used to gain access and the results of the attacker's unauthorized attack for possible prosecution measures. This paper is based upon the introduction to honeypots, their importance in network security, types of honeypots, their advantages disadvantages and legal issues related with them. Finally we shall conclude by looking at what the future holds for honeypots.**

***Index Terms*- Honeypot, Types of Honeypot, Advantage, Disadvantage, Proposed System, Honeynet, Conclusion**

I. INTRODUCTION

Honeypots are closely monitored decoys that are employed in a network to study the trail of hackers and to alert network administrators of a possible intrusion. Honey Pots are fake computer systems, setup as a "decoy", that are used to collect data on intruders. This "decoy" appears to contain operating system vulnerabilities that make it an attractive target for hackers. A Honey Pot, loaded with fake information, appears to the hacker to be a legitimate machine. While it appears vulnerable to attack, it actually prevents access to valuable data, administrative controls and other computers. Deception defenses can add an unrecognizable layer of protection. As long as the hacker is not scared away, system administrators can now collect data on the identity, access, and compromise methods used by the intruder. The Honey Pot must mimic real systems or the intruder will quickly discover the 'decoy'. Honey Pots are set up to monitor the intruder without risk to production systems or data. If the Honey Pot works as intended, how the intruder probes and exploits the system can now be assessed without detection. The concept of a Honey Pot is to learn from the intruder's actions. This knowledge can now be used to prevent attacks on the "real", or production systems, as well as diverting the resources of the attacker to a the 'decoy' system.

A honeypot is a system that is built and set up in order to be hacked.Honeypot can be deployed in order to consume the resources of the attacker or distract him from the valuable targets and slow him down that wastes his time on the honeypot instead of attacking production systems.

Main functions of a honeypot are:

1. To divert the attention of the attacker from the real network, in a way that the main information resources are not compromised .

2. To capture new viruses or worms for future study

3.To build attacker profiles in order to identify their preferred attack methods.

4. To identify new vulnerabilities and risks of various operating systems, environments and programs which are not thoroughly identified at the moment.

Features of a honeypot are:

1) It is based on real web application environments, constantly keeping surveillance on the data entry and taking instant reaction on detecting the intruder.

2) It can get complete attack sequence.

3) It captures the unknown attacks just by watching/monitoring the activities of the intruder.

4) It makes the network safer for the future.

II. TYPES OF HONEYPOT

Honeypots classified based on their deployment :

1.Production honeypots 2.Research honeypots

- *Production honeypots*

Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, which are easier to deploy. They give less information about the attacks or attackers than research honeypots.

- *Research honeypots*

Research honeypots are run to gather information about the motives and tactics of the Black hat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats that organizations face and to learn how to better protect against those threats.Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations.

Honeypots on the basis of level of interaction :

1. Low interaction 2. High interaction

- *Low Interaction*

In low interaction Honeypots there is no operating system that an attacker can operate on. Instead operating system emulators are installed which interacts with the attacker. It offers limited interaction level to the attackers. It will be used to scan the port and generates attack signatures.

- *High Interaction*

High interaction Honeypots have actual operating system and has tools which motivates the attacker to attack so that their attack strategies can be recorded and later analysed. As high interaction Honeypot offers 24/7 internet connectivity, it attracts the attackers and to reduce the load of these high interaction Honeypots, only traffic filtered by low interaction. Honeypots is passed to them. So high interaction Honeypots basically process the packets sent only by malicious users.

III. ADVANTAGE OF HONEYPOT

1. Small data sets : Any connection made with the honeypot is considered as malicious. So the thousands of alerts logged by organizations can be reduced to hundreds of entries.

2 Reduce false positives and false negatives.

3. Catching False negatives: Catching false negatives with the help of honeypots is quiet easy.

4. Encryption: Honeypots have the capability to capture the malicious activity if it is in encrypted form.

5. Flexible : Honeypots are extremely adaptable in variety of environments.

6. Minimal Resources: Honeypot require minimal resources.

7. Simplicity – Honeypots are simple.For their functionality they do not require complicated algorithm or operations.

IV. DISADVANTAGE OF HONEYPOT

1. Single Data Point: One huge drawback is generally faced by honeypots that they are worthless if no one attacks them. Obviously, they can accomplish wonderful things but if the attacker doesn't send any packet to honeypots then it would blissfully unaware of any unauthorized activity.

2. Risk: Once compromised , honeypots can introduce risk to organisation's environment. Different kind of honeypots possess different levels of risk. Low interaction honeypots

3. Risk of takeover – after gaining control over the Honeypot attacker can retrieve all the collected data.

4. Disclosure of identity – Honeypot has expected characteristics and behaviour. Experienced
attacker can detect presence of incorrectly configured decoy in system.

V. PROPOSED SYSTEM

The proposed system is based on the concept of a ticketing authority; The main idea of a ticketing authority is the use of issued tickets to allow clients to access network resources.The proposed model utilizes this idea for assigning permissions to an authenticated client. The back-end server will compare the requested operation with the client's permissions to determine whether the requested operation is allowed. If the back-end server finds a discrepancy between permissions and requested operations, the back-end server will transfer the packet to the deployed

In this paper we take an overview of what several of these problems are, and look at possible approaches on how to solve them. By identifying these problems now, we can hope to make honeypots a stronger technology for the future. The three problems we discuss below are identifying honeypots, exploiting honeypots, and attacker clientele honey pot for filtration.

- *Identifying honeypots*

As we have seen in the past months, there are many types of different honeypots (both low and high interaction) that can achieve many different things (tarpitting, detection, countering spam, information gathering, etc). Most of these honeypots share a common trait -- their value diminishes upon detection. Once detected, an attacker now knows which systems to avoid (your honeypot) or potentially even worse, can now feed your honeypot false or bogus information. This is why in most cases you want your honeypot to avoid detection. Some exceptions do exist, such as in deterrence. Organizations may want to be known for using honeypots or have some of them detected, as it could deter attacker from probing their networks. Or in the case of sticky honeypots stopping worms, there is little threat (at least at the moment) of the worm using honeypot detection routines as worms are too busy scanning to care about honeypots.

In most cases you want your honeypots to avoid detection. Honeypots are growing in use and we have already begun to see tools and techniques released to counter and detect them. One of the more unique examples is the commercial tool Honeypot Hunter, used by the Spamming industry to identify honeypots. Here is a tool developed and released for the sole purpose of identifying Spam-catching honeypots. Other tools have been developed to identify virtual honeypots, and papers have been published that identify potential issues.

So, what can be done? First, realize that no matter what type of honeypot you are dealing with, from the most basic Back Officier Friendly, to the most advanced Honeynet, any honeypot can eventually be detected. While the goal is to have a honeypot that is never detected, if you have an adversary that has the necessary skills or the proper tools and they are looking for honeypots, then its only a matter of time. In many ways, just like most other technologies such as IDS sensors, honeypots are in an arms race. As new honeypots are released, or newly updated versions appear, attackers can identify ways to detect and identify them. As these new detection methods are developed, counter detection measures can be built into the honeypots. Attackers can then counter these new measures, and the cycle continues. For example, to remotely identify older versions of the Honeyd honeypot, you merely had to send a SYN packet, as the honeypot would respond with a SYN/ACK packet that had no options. However, if you were to use Nmap to profile the same honeypot, then it would respond to SYN packets with options (this has now been corrected in Honeyd ver 0.7a).

As I see it, there are two steps you can take now to address this problem. First, decide at what point does detection diminish the value of your honeypot. If your honeypot can derive value before it is detected, then it has potentially still done its job. For example, lets say you deploy honeypots on your internal network to detect unauthorized activity (such as someone scanning for open file shares). The purpose of your honeypot is to act as a burglar alarm (as Marcus Ranum likes to call honeypots). Let's say that an attacker is on your internal network, and while probing for vulnerable systems he probes (and detects) your honeypot. In this case, even though the honeypot was identified it still has potentially done its job, detecting and alerting you to a threat. Even if it was detected minutes after being probed, the honeypot is letting you know there is a threat on your internal network, and the threat is actively looking for open file shares. For other honeypots, the story is different. For example, if you are using Honeynets to gather information, detection compromises your ability to collect accurate data. In this case, you want your honeypot to go for days, if not weeks or even months, without detection. This can be much harder to achieve. So, the first step is for you to decide just how important detection is to you, and how long your honeypot needs to remain undetected.

If in step one you determine that avoiding detection is important to you, then you want to consider customization. There are many different types of honeypot solutions you can download and work with. Just as you can easily download the solutions, so to can attackers. They can download evaluation copies or source code of anything publicly available, analyze it, and identify signatures. In many ways, its similar to how Fyodor's powerful Nmap can remotely fingerprint operating systems, as each IP stack has its own unique idiosyncrasies. To counter this, you need to customize your honeypot, change its behavior or appearance so it does not look like every other honeypot on the Internet. The more you modify the default behavior, the potentially more difficult it will become for attackers to identify it. For example, if you are using the Honeyed Toolkits for Linux, you do not want to use the default honeyd.conf files for production environments. Instead, modify the behavior of the templates to adapt to your environment. More advanced users can potentially modify the source code, so as to change how packets are created. Regardless, attackers may be looking for a specific type of known honeypot behavior. You can help minimize the chance of detection if your honeypot is behaving or reacting in ways attackers do not expect.

- Exploting honeypots

Anything coded by humans can be compromised. For years this has been true for various applications such as firewalls, webservers, or browsers. Whenever a new application has been released, we can expect bug or vulnerability reports. Honeypots are no different. We have to assume for that every honeypot released, there are known (and unknown) vulnerabilities in those systems. As with any other security technology, steps should be taken to protect against unknown attacks. With low interaction honeypots, the risk is somewhat limited as there are only emulated services for attackers to interact with, they are not given real applications to exploit, nor real operating systems to gain access to. However, we should assume that an attacker can bypass the controlled environments of emulated services, and as such everything should be done to secure the honeypot application. For Win32 low-interaction honeypots (such as KFSensor), you want to build a secure base OS with the latest patches, disabling all services. Perhaps even install a host based firewall, one that allows inbound connections to any port the honeypot is monitoring but blocks all other inbound connections. Even more importantly, have the firewall block (and alert) any outbound-initiated connections, to help protect against the threat of the honeypot when compromised. For Unix low-interaction honeypots we can take greater measures. Chroot() is one usual way to improve containment against attacked processes on a Unix system. Jail() (under FreeBSD) proposed a real way to restrict what could be seen by processes. Low level kernel patches like Systrace (Process based Discretionary Access Control) or Grsecurity (Process based Mandatory Access Control, Address Space Protection, etc) or others such as SE Linux should be used in low-interaction honeypots to help protect against known and unknown attacks.

For high-interaction honeypots, the problem is more challenging. These solutions provide real operating system and applications for attackers to interact with, as a result they have greater risk. It is expected for attackers to gain privileged control of the honeypots. This means external Data Control measures have to be put in place, such as an IPS (Intrusion Prevention System) or bandwidth limiting. In these cases (such as Honeynets) there are two steps you can take. First, use several layers of control. This prevents having the risk of a single point of failure. The second is human intervention. High-interaction honeypots should be closely monitored. Any time there is anomalous activity on your honeypot (outbound connections, uploaded files, increased system activity, new processes, system logins, etc) a human should then be monitoring everything that happens on the system. Anytime an attacker's action exceeds your organizations threshold for risk (such as attempting an outbound attack) you can terminate the attacker's connection, drop packets, redirect connections, etc. The advantage to real time monitoring is you can potentially identify activity that automated mechanisms may miss. This also gives you far greater control over what the attacker does, and how your honeypot responds.

- Attacker clientele

For us, this has been the toughest nut to crack, in part because it's just not a technical issue. One of the biggest challenges of honeypots is how can they be deployed to detect, identify, and capture the activity of specific threats, both internal and external to a company. Think of deploying honeypots as similar to fishing. Traditionally, most honeypot deployments have not been focused on a specific target, instead they have been common systems deployed on external networks. This is similar to going fishing to any local lake, throwing out a line with an ordinary worm on it, and you're happy with whatever you catch. In most cases, these 'fish' have been attackers that focus on targets of opportunity, probing and breaking into as many systems they can find, often using automated tools. These threats are relatively easy to capture with honeypots, as they are highly active, will attack anything with an IP stack, and most often don't spend the time checking to see if they are interacting with a honeypot.

VI. HONEYNETS

Two or more honeypots on a network form a honeynet.Typically, a honeynet is used for monitoring a larger and more diverse network in which one honeypot may not be sufficient. Honeynets and honeypots are usually implemented as parts of larger network intrusion detection systems. A honeyfarm is a centralized collection of honeypots and analysis tools.The concept of the honeynet first began in 1999 when Lance Spitzner, founder of the Honeynet Project, published the paper "To Build a Honeypot"."A honeynet is a network of high interaction honeypots that simulates a production network and configured such that all activity is monitored, recorded and in a degree, discreetly regulated."

VII. CONCLUSIONS

In this paper we have provided overview of what honeypots are, and what they are useful for.We have discussed the different types of honeypots such as production honeypots,research honeypots.Honeypots are an emerging technology, with extensive potential. They have tremendous advantages that can be applied to a variety of different environments. They dramatically reduce false positives, while providing an extremely flexible tool that is easy to customize for different environments and threats.Traditionally, honeypots have been applied against external threats or common internal threats.However, by combining the capabilities of Honeynets, honeypots contribute to the early indication and confirmation of advanced insider threats. The research in this area is still in the early stages, with the intent of greater testing and development in the future.

Honeypots have tremendous potential for the security community, and they can accomplish goals few other technologies can. Like any new technology, they have some challenges to overcome. Most likely none of these problems will ever be completely solved or eliminated. However, expect to see in the next 12 to 18 months many new developments that help address these, and other issues.Honeypots have the potential to capture bigger fish. Organizations may not be concerned about automated or common attacks, they may be more concerned about advanced attackers targeting their critical systems, or employees who are stealing and selling their confidential information. For honeypots to capture such threats, the honeypots have to be tuned for each individual threat, we need the proper bait and location. When you are fishing for 150 pound Tarpons, you don't simply throw a hook and worm in the local pond. Instead, you travel to the Florida Keys during the spring and summer. The same analogy holds true for more advanced attackers. Your honeypots have to be located in the proper location, at the proper time, and with the correct bait. For this, such honeypots have to be customized to your specific threat, a much more difficult job to do. For example, if you are concerned about organized crime breaking into your ecommerce site, throwing a default RedHat 7.3 honeypot on your external network is most likely not going to capture their activity. If you are out to catch the latest attacks or exploits, you need high value targets, such as a CVS honeypot, that will give attackers a high ROI (Return on Investment) for their new attack. For internal threats, you need honeypots that have value to that insider, such as honeypots that appear to be research and development databases. To go after a specific threat, your honeypot has to be tuned to that individual.

We tried to research errors within Honey pots. This topic is really interesting because many organizations create Honey traps to track hackers. This instrument is widely used in the financial industry where institution develop honeypots which carry fake numbers and records to track the hackers. After researching further on this topic and its loop holes, We found out that Honey pot is very volatile regarding risk factor. When companies create Honeypots they usually tries to mimic the original infrastructure of the company's system, so the hackers wouldn’t be able to detect the entrapment. This is one place where the errors of Honeypots exist. By mimicking the company’s infrastructure system to create the Honeypot, the company is literally allowing the hacker to analyze and read the construction of the system. In many cases of Honeypots, it is not easy to confront the hacker because he/she will be using many worms to hack others. Therefore, it may be not a great idea to use Honey pot to play with hackers. In addition, even after creating the best honey pot trap, who decides the legality of the system".

We posed the following questions.

- Is a honey pot considered entrapment if you want to press charges?
- Can you really claim “damage” if it’s a fake site that’s broken into?
- If you build the site with the intention of attracting a break-in, is it actually against the law if someone does just that?

We concludes

"Therefore, we findings question what additional benefits do Honeypots serve if the company can’t even press charge against the hackers. In most times the hacker wouldn’t be able to track by the company’s system. Finally, a honeypot is a great decision to safe guard the company’s record, but if it’s not devereal system."loped properly, then it could back fire towards company’s.

REFERENCES

[1] Wikipedia. http://en.wikipedia.org/wiki/Honeypot(computing)

[2] Srivathsa S Rao,Vinay Hegde, Boruthalupula Maneesh, Jyothi Prasad N M,Suhas Suresh,"Web based honeypots network",International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013

[3] Chao-Hsi Yeh and Chung-Huang Yang, “Design and Implementation of Honeypot Systems Based on Open-Source Software”, IEEE International Conference on Intelligence and Security Informatics (ISI), 265-266, 2008.

[4] Pushpa Rani, Yashpal Singh, S Niranjan,"A Review on Honeypot as an Intrusion Detection System for Wireless Network ", International Journal of Engineering Research and Development e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com Volume 7, Issue 4 (May 2013), PP. 71-74 71

[5]Matthew L. Bringer, Christopher A. Chelmecki, and Hiroshi Fujinoki,"A Survey: Recent Advances and Future Trends in Honeypot Research",I. J. Computer Network and Information Security, 2012, 10, 63-75

[6] A.Chandra,K. Lalitha,"Honeypots: A New Mechanism for Network Security ", Publications Of Problems & Application In Engineering Research - Paper http://ijpaper.com/ CSEA2012 ISSN: 2230-8547; e-ISSN: 2230-8555

[7] Lance Spitzner,"Honeypots: Catching the Insider Threat "

[8] Navneet Kambow, Lavleen Kaur Passi,"Honeypots: The Need of Network Security ",Navneet Kambow et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6098-6101

[9] Miss.Swapnali Sundar Sadamate,"Honeypot Mechanism – the Autonomous Hybrid Solution for Enhancing",International Journal of Advanced Research in Computer Science and Software Engineering.