RESEARCH ARTICLE OPEN ACCESS

SECURING WMN USING HYBRID HONEYPOT SYSTEM

Dr.T.Geetha[1], R.Karthikeyan[2] , Kumar M [3], Kathiravan M [4]
[1,2] *Asst.Prof, Dept of MCA, Gnanamani college of Technolgy, Namakkal, INDIA.*
[3,4] *P.G.Scholar, Dept of MCA, Gnanamani college of Technolgy, Namakkal, INDIA.*

Abstract:

Nowadays, we are facing with network threats that cause enormous damage to the Internet community day by day. In this situation, more and more people try to prevent their network security using some traditional mechanisms including firewall, Intrusion Detection System, etc. Among them honeypot is a versatile tool for a security practitioner, of course, they are tools that are meant to be attacked or interacted with to more information about attackers, their motives and tools. In this paper, we will describe usefulness of low-interaction honeypot and high-interaction honeypot and comparison between them. And then we propose hybrid honeypot architecture that combines low and high - interaction honeypot to mitigate the drawback. In this architecture, low-interaction honeypot is used as a traffic filter. Activities like port scanning can be effectively detected by low-interaction honeypot and stop there. Traffic that cannot be handled by low-interaction honeypot is handed over to high-interaction honeypot. In this case, low-interaction honeypot is used as proxy whereas high-interaction honeypot offers the optimal level realism. To prevent the high-interaction honeypot from infections, containment environment (VMware) is used.

Keywords — ***Low-interaction honeypot, High-interaction honeypot, Wireless Mesh Network, Proxy.***

INTRODUCTION

Global communication is getting more significant every day. At the same time, computer crimes are growing rapidly. Counter measures are developed to detect or prevent attacks - most of these measures are based on known facts, known attack patterns. As in the military, it is important to know, who your enemy is, what kind of strategy and plan he uses, what tools he utilizes and what he is aiming for. Gathering this kind of information is arduous but important. By knowing attack strategies, countermeasures can be improved and anomalies can be fixed. To gather as much information as possible is one main target of honey pot. Generally, such information gathering should be done without the attacker's knowledge. All the gathered information provides an advantage to the defending side and can therefore be used on productive systems to prevent attacks.

I.DEFINITIONS OF HONEYPOT

A honeypot is a system that is built and set up in order to be hacked. Honeypot can be used in a different scenario as intrusion detection facility (burglar alarm), defense or response mechanism. Moreover, 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 to divert the attention of the attacker from the real network, in a way that the main information resources are not compromised.

- *To capture new viruses or worms for future study*

- *To build attacker profiles in order to identify their preferred attack methods, similar to criminal profiles used by law enforcement agencies in order to identify a criminal's modus operandi*

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

II.LEVEL INTERACTION OF HONEYPOT

The level of interaction is defined as the range of attack possibilities that a honeypot allow an attacker to have, where as it can be classified as high- interaction honeypot and lowinteraction honeypot.

A. High- Interaction Honeypot

In high- interaction honeypot, attacker interaction with real operating systems, services and programs and it can be used to observe the attackers behavior, their tools, motivation and explored vulnerabilities. This kind of honeypot must have a robust containment mechanism in order to prevent, once compromised, its use to attack other networks. One goal of a hacker is to gain root and to have access to a machine, which is connected to the internet 24/7. A high-interaction honeypot does offer such an environment. To facilitate the deployment of machines, automatic installation through images retrieved from system from Quattor can be used. Tools like Sebek can help high-interaction honeypot to instrument to log and/or system calls.A high- interaction honeypot can be installed inside a virtual machine using virtualization software such as VMware, Qemu and Xen. Using virtualization software, the attacker may run specialized code to detect that his code is running inside a virtual machine environment or perform timing attacks to identify honeypots. And performance of applications running in the guest operating system is reduced. However, an effort is made in the architecture to reduce the load of high-interaction honeypots by preprocessing the traffic using low-interaction honeypots as much as possible. Example of high-interaction honeypot is honeynet. A honeynet is a network of multiple systems. Honeynet can collect in-depth information about attackers, such as their keystrokes when they compromise a system, their chat sessions with fellow black hats, or the tools they use to probe and exploit vulnerable systems. This data can provide incredible insight on the attacker themselves. The advantage with honeynet is that they collect information based on the attackers' actions in the wild.

B. Low- Interaction Honeypot

On low- interaction honeypot, there is no operating system that an attacker can operate on. Tools are installed in order to emulate operating systems and services. And they interact with the attackers and malicious code. This will minimize the risk significantly. This kind of honeypot has a small chance of being compromised. It is production honeypot. Typical use of low-interaction honeypot includes; port scans identification, generation of attack signatures, trend analysis and malware collection.On the other hand, this is also a disadvantage. It is not possible to watch an attacker interacting with the operating system, which could be really interacting. Example of lowinteraction honeypot is honeyd. Honeyd is an open source low-interactivity honeypot system that creats virtual hosts that can be configured to run arbitrary services and their personality can be adapted so that they appear to be running certain operating systems. Honeyed , enables a single host to claim multiple addresses. Honeyed improves cyber security by providing mechanism for threat detection and assessment. It also deters adversaries by hiding systems in the middle of virtual systems.It is possible to ping the virtual machines or to trace out them. Any type of service on the virtual machine can be simulated according to a simple configuration file. Instead of simulating service, it is also possible to proxy it to another machine. A

complete picture of how honeyd work is shown in following.

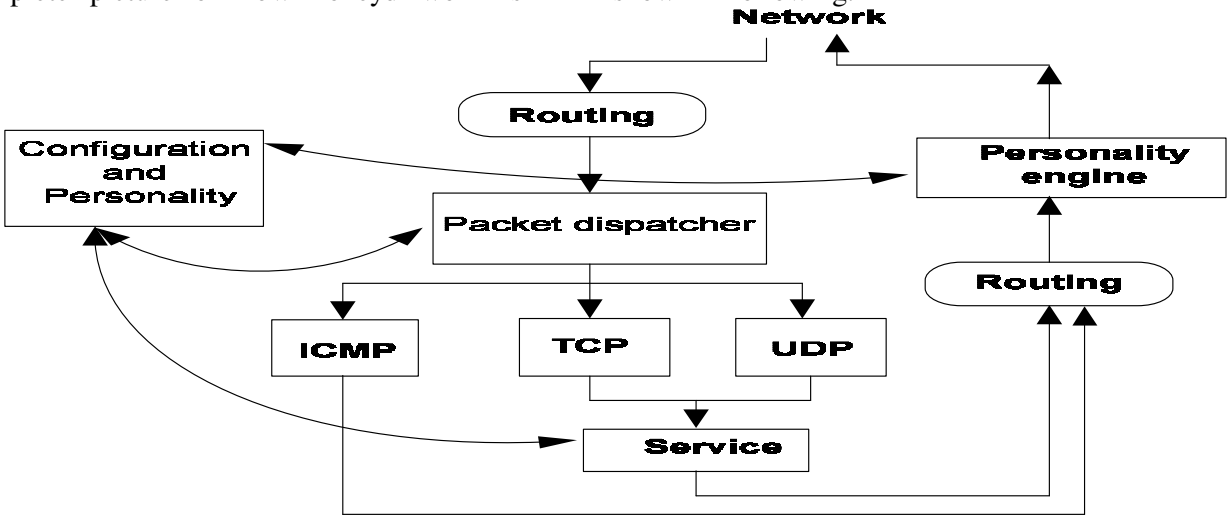


Fig: A simplified view of the honeyd architecture

C.Comparison between low- interaction honeypot and high- interaction honeypot

Each level has advantages and disadvantage as mention below;

TABLE : SUMMARIZES OF LOW AND HIGH-INTERACTION HONEYPOT

Name of experiments	Low-Interaction Honeypot	High-Interaction Honeypot
Degree of Involvement	Low	High
Real Operating System	No	Yes
Risk	Low	High
Information Gathering	Connections	All
Compromised Wished	No	Yes
Knowledge to Run	Low	High
Knowledge to Develop	Low	Mid-High
Maintenance Time	Low	Very High

III.HYBRID HONEYPOT ARCHITECTURE

In this system, low- interaction honeypot act as lightweight proxy. We want high-interaction honeypot to process all traffic destined to black IP address space. We need to offload them as front end to high-interaction honeypot because it is instrumented machines. Honeyd has the appropriate properties to play the role of the front end and acts as a filtering

component. The lightweight proxy responds only to TCP/SYN requests to ports that are open. For any other ports, it just absorbs and records the packets received. When the three-way handshake has completed properly between the attacker and the low- interaction honeypot, the connection must be handoff to the appropriate high-interaction honeypot . At this point, also referred as zero point, the low-interaction honeypot set as a connection with the high- interaction honeypot. The lowinteraction honeypot sets as like relay agent. Any application level data coming from attacker is forwarded to the highinteraction honeypot and vice versa, until the connection is terminated. This behavior is embedded to the honeyd implementation, know as proxy mode. The proxy mode is instrumented to record the message exchanges, for further analysis purposes. Hand-off is useful in case of port scanning, where low-interaction honeypot will absorb all incoming connections without disturbing high-interaction honeypot. In the following illustration, initially, the attacker sends a TCP/ SYN packet to the low-interaction honeypot. If the honeypot is configured to listen to the port, then it sends a SYN/ACK packet and waits to receive the next packet. If the packet is not an ACK then the low-interaction honeypot assumes that it was a port scan and the connection is dropped. If the third packet received is ACK then it is a valid TCP connection and the zero point is reached. Thus the lowinteraction honeypot connects with the high-interaction honeypot running the requested service. Then after the connection establishment the low-interaction honeypot continues to work as a proxy. As low and high- interaction honeypots belong to the same local network, no additional delay will be perceived by the attacker.

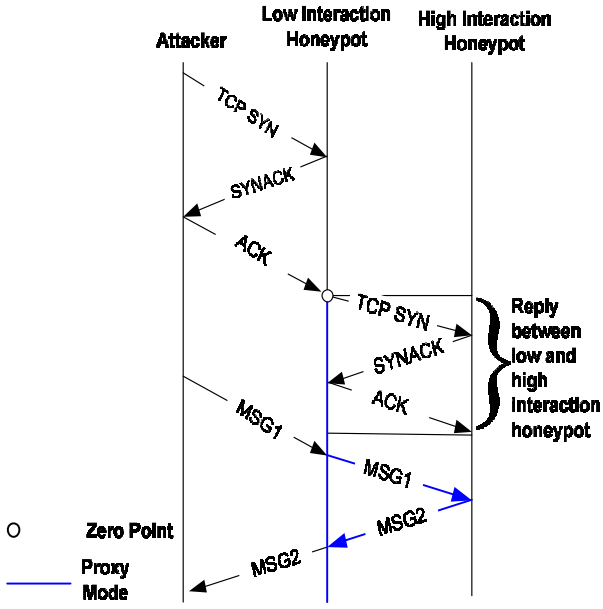


Figure 2: An example of

A.Application of the hybrid framework

In this section, we discuss the applicability of this framework that included distributed lightweight proxy deployment, a centralized VMware system, and the connection handoff to solving some of the standard problems in Internet thread detection and resolution.

B.Detection

One unique application of the hybrid framework is to the area of worm detection. Recall that the backend component of actual hosts serving as honeypots. Our novel worm propagation detection mechanism actually watches for the propagation of worms. Worms often use the same propagation method from host to host; we can apply the same content check summing algorithm to packet out of the backend honeypot and match them to the MD5 of the inbound connection. A matching outbound signature that matches an inbound handoff signature is even a higher indicator of self propagation code. Here, we give criteria that characterize the methodology that was applied to evaluation attack detection quality . The following criteria characterize the methodology used to evaluate attack detection:

Measured Parameter. It can be either:

- FP: False Positives

- FN: False Negatives
- TP: True Positives
- TN: True Negatives
- Probe Size: The number of different attacks (true positives/false negatives) or the number of benign interactions (true negatives/false positives) if available.
- Establishment of Ground Truth. Parameters can be either: – manual analysis: We did a manual analysis to find the actual number of attacks.

C. Signature Quality

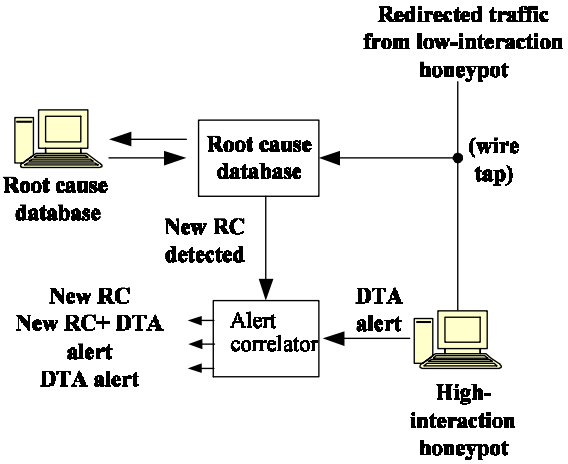
The following criteria characterize the methodology used to evaluate signature quality:

• Signature Generation Setup: The setup used to generate the set of signatures for the signature quality assessment. Parameters can be either:

- **Real-world deployment:** The system was deployed in a realworld network environment while generating the set of signatures.
- **Lab deployment, synthesized traffic:** The traffic used to generate the signature set was either hand-crafted or dynamically generated using a tested.
- **Lab deployment, replayed traffic traces:** The traffic used to generate the signature set was generated by replaying a set of (real-world) traffic traces.

D.Signature generation

Signature generation is the process of defining all the necessary characteristics of a new thread to be able to detect a new occurrence of the threat, identify existing infected hosts, and immunize against additional infections. This process is uniquely suited to the hybrid architecture as is required, sufficient number of mentioned hosts to catch the threat early in its growth phase and sufficient detailed behavioral analysis to identify previously unseen threats. We propose to use a combination of host- and networkbased attack detection algorithms, namely Dynamic Taint Analysis and Root Cause Analysis. The interaction between both systems is depicted in Figure 5. Traffic which is directed to the high interaction honeypot will be split and redirected to a root cause analysis engine. This engine monitors the attack activity on the network level, and generates an alert in case a new root cause is detected. For deciding whether a root cause is novel or not, a central database will be contacted. In a second step the alert correlator matches the DTA and RCA alerts. Alert correlation can be done based on simple timing information, i.e. when was an alert triggered. However, since the alert generation time can differ significantly for RCA and DTA, it would be better to correlate only alerts that were generated by the same network traffic. This is especially important in case a sensor is attacked very frequently.



E.True/ False Positive Ratio

True Positive Ratio (TPR) is a way showing how good the intrusion detection is at alerting on real attacks. In our setting we use this to

better performance. TPR is obtained by the following formula:

$$TPR = \frac{TP}{TP+FN}$$

Where, TP= The number of alerts on malicious traffic, FN= The number of missing alerts on malicious traffic. The total number of intrusion is given by TP+FN.. False Positive Ratio(FPR) shows the proportion of instances, which were not an attack but still were alerted on. FPR is result of the following formula:

$$FPR = \frac{FP}{FP+TN}$$

Where, FP=The number of alerts on benign traffic, TN= The number of correct decisions on benign traffic. The total number of no-intrusion is given by FP+TN. A perfect system would have TPR=1 and FPR=0. This would result in alerts only on malicious traffic, and no alerts on benign traffic. The confusion matrix in figure 6 illustrates what FP, FN, TP and TN mean.

		Alerts? YES	Alerts? NO
Attacks?	YES	TP	FN
Attacks?	NO	FP	TN

High-interaction honeypot	Low-interaction honeypot
- Slow	+ Fast
+ Able to detect unknown attacks + 0 False positive	- Unable to detect unknown attacks
- Unable to deal with time bombs and user interaction	+ Able to deal with time bombs and user interaction
- Expensive	+ Cheap
- Difficult to setup and operate	+ Easy to setup and operate

CONCLUSION

Using hybrid honeypot, we achieve a number of goals. First, we need to maintain only a small number of high- interaction honeypots since the portion of the traffic will be routed to them is limited. All port- scan attempts or connection to port that is not open will be stopped by low-interaction honeypots. Second, the high-interaction honeypots will be placed in a monitored network. Thus if a honeypot gets infected, the infection rate will be contrololable either through limiting bandwidth or traffic reflection. Also, since honeyd can emulate different machines running in a network, we can map several machines which run the same operating system and similar services to a single high-interaction honeypot. Finally, the addition of new services to the high- interaction is facilitated we only should open appropriate port at the low-interaction honeypot and set up the mapping.Honeypots offer a unique perspective to defending networks by learning the habits and techniques of the blackhat at an additional cost of minimal network alert reporting and monitoring time.

Hybrid honeypot

+Fast

+Able to detect unknown attacks

+0 False positive

REFERENCES

1. Spitzner, L. Open Source Honeypots: Learning with honeyd, Security Focus, 2003.
2. Wikipedia. http://en.wikipedia.org/wiki/Honeypot_(computing).
3. Karthik, S., Samudrala, B. and Yang, A.T. Design of Network Security Projects Using Honeypots. Journal of Computing Sciences in Colleges, 2004.

4. R.Karthikeyan," Improved Apriori Algorithm for Mining Rules" in the International Journal of Advanced Research in biology Engineering science and Technology Volume 11, Issue 4, April 2016, Page No:71-77.
5. R.Karthikeyan,Dr.T.Geetha "Honeypots for Network Security", International journal for Research & Development in Technology.Volume 7.Issue 2 ,Jan 2017,Page No.:62-66 ISSN:2349-3585
6. R.Karthikeyan,"A Survey on Position Based Routing in Mobile Adhoc Networks" in the international journal of P2P Network Trends and Technology, Volume 3 Issue 7 2013, ISSN:2249-2615
7. R.Karthikeyan,"A Survey on Sensor Networks" in the International Journal for Research & Development in Technology Volume 7, Issue 1, Jan 2017, Page No:71-77
8. R.Karthikeyan,Dr.T.Geetha "Web Based Honeypots Network",in the International journal for Research & Development in Technology.Volume 7.Issue 2 ,Jan 2017,Page No.:67-73 ISSN:2349-3585.
9. R.Karthikeyan,Dr.T.Geetha,"A Simple Transmit Diversity Technique for Wireless Communication",in the International journal for Engineering and Techniques. Volume 3. Issue 1, Feb 2017, Page No.:56-61 ISSN:2395-1303.
10. C.Ganesh,B.Sathyabhama,Dr.T.Geetha " Fast Frequent Pattern Mining using Vertical Data Format for Knowledge Discovery "International Journal of Engineering Research in Management & Technology. Vol.5,Issue-5,Pages:141-149.
11. R.Karthikeyan,Dr.T.Geetha "Strategy of Trible – E on Solving Trojan Defense in Cyber Crime Cases", International journal for Research & Development in Technology.Volume 7.Issue 1 ,Jan 2017,Page No.:167-171
12. Know your enemy Honeynets, http://www.honeynet.org/papers/key.html SANS institute GIEC certification GSEC
13. R.Karthikeyan,Dr.T.Geetha"Advanced Honey Pot Architecture for Network Threats Quantification" in the international journal of Engineering and Techniques, Volume 3 Issue 2, March 2017, ISSN:2395-1303, PP No.:92-96.
14. Assignments#1.4:Honeypots Stretegic Considerations,2002.
15. R.Karthikeyan,Dr.T.Geetha"Estimating Driving Behavior by a smart phone" in the international journal of Engineering and Techniques, Volume 3 Issue 2, March 2017, ISSN:2395-1303,PP No.:84-91.
16. R.Karthikeyan,"A Survey on Position Based Routing in Mobile Adhoc Networks" in the international journal of P2P Network Trends and Technology, Volume 3 Issue 7 2013, ISSN:2249-2615.
17. K.Ramya and K.Pavithradevi "Effective Wireless Communication",International journal of Advanced Research, Vol 4(12), pp.1599-1562 dec 2016.
18. R.Karthikeyan,Dr.T.Geetha "FLIP-OFDM for Optical Wireless Communications" in the international journal of Engineering and Techniques, Volume 3 Issue 1, Jan - Feb 2017, ISSN:2395-1303,PP No.:115-120.
19. R.Karthikeyan,Dr.T.Geetha"Application Optimization in Mobile Cloud Computing" in the international journal of Engineering and Techniques, Volume 3 Issue 1, Jan - Feb 2017, ISSN:2395-1303,PP No.:121-125.
20. Kreibich, C. and Crowcroft, J. Honeycomb – Creating Intrusion Detection Signatures Using Honeypots Proceedings of the Second Workshop on Hot Topics in Networks (Hotnets II), Boston, 2003, 51-56.
21. R.Karthikeyan,Dr.T.Geetha"The Sybil Attack" in the international journal of Engineering and Techniques, Volume 3 Issue 3, May - Jun 2017, ISSN:2395-1303,PP No.:82-88.
22. R.Karthikeyan,Dr.T.Geetha"Automated Predictive Big Data Analytics using Ontology Based Semantics" in the international journal of Engineering and Techniques, Volume 3 Issue 3, May - Jun 2017, ISSN:2395-1303,PP No.:77-81.