Taxonomy of Honeynet Solutions

Wenjun Fan[1], Zhihui Du[2], David Fernández[1]
[1]Departamento de Ingeniería de Sistemas Telemáticos
ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040, Madrid, Spain
Email: efan@dit.upm.es, david@dit.upm.es
[2]Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China
duzh@tsinghua.edu.cn

***Abstract*—Honeynet research has become more important as a way to overcome the limitations imposed by the use of individual honeypots. A honeynet can be defined as a network of honeypots following certain topology. Although there are at present many existing honeynet solutions, no taxonomies have been proposed in order to classify them. In this paper, we propose such taxonomy, identifying the main criteria used for its classification and applying the classification scheme to some of the existing honeynet solutions, in order to quickly get a clear outline of the honeynet architecture and gain insight of the honeynet technology. The analysis of the classification scheme of the taxonomy allows getting an overview of the advantages and disadvantages of each criterion value. We later use this analysis to explore the design space of honeynet solutions for the proposal of a future optimized honeynet solution.**

Keywords—taxonomy of honeynet solutions; honeynet solutions; virtual honeynet; design space

I. INTRODUCTION

Since Clifford Stoll published his experience tracking a network hacker in his book "The Cukoo's Egg" [1], honeypots, the tools used for malicious behavior investigation have been widely used in a variety of security scenarios. A general definition of the term honeypot was proposed by Lance Spitzner [2]: a honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. Based on this essential concept of honeypot, many different honeypots have been developed and proposed. Every kind of honeypot has its own features and advantages, but also has its shortcomings. High interaction honeypots can provide high-level fidelity of the data captured, allowing a deep investigation of adversary's behavior, because they typically use the same vulnerable service or software as the real system. But their higher cost in term of resources is a problem that limits their large-scale deployment. On the contrary, low interaction honeypots benefit by their inherent lightweight design, facilitating their large-scale deployment. However, due to its functionality limitation, low interaction honeypots suffer from lack of fidelity. Otherwise, dynamic honeypots can adapt itself to the attacker's network context in real time, but the design of a dynamic honeypot is complex, and reaching the right performance and response delay could be problematic. Meanwhile, static honeypot profits from a more compact design at the price of losing the adaptability. Moreover, most honeypots focus on server-side security, by analyzing attacks coming from clients. But there are also honeypots that can play the role of a client-side application, in order to investigate malicious software on servers. In addition, some honeypots can detect unknown attacks, while some other honeypots even can react to those attacks. Some honeypots are used to observe interesting traffic while some other honeypots are used to monitor the baleful behavior or system activity.

In summary, the capability of one single honeypot is limited. Thus, nowadays, more and more honeypot systems focus on the combination of different types of honeypots in a honeynet to get an optimized solution. For example, a typical honeynet solution is an hybrid system consisting of a combination of high interaction and low interaction honeypots, which provides a good balance among scalability, performance, fidelity and containment. From an attacker's point of view, the honeynet appears to have servers and desktop machines, many different types of applications, and several different platforms. That is why the term "zoo" is also used to name honeynets, as they allow capturing the wild hacker in their natural environment.

In a sense, a honeynet is an extension of the honeypot concept. Initially, a honeypot system can be made of an individual honeypot, but it also can be composed of a group of individual honeypots, considered as a whole system from outside. In a narrow sense, the honeynet term refers to Gen I, II and III honeynets that are of high-interaction type of honeypot system [3], which consists of multiple individual honeypots. However, with the development of multiple research projects around honeynets, the variety of honeynets proposed does not match any longer the narrow honeynet definition, so a generalized honeynet definition is needed.

In this paper, the honeynet term is not limited in its narrow definition. A honeynet is a network of individual honeypots deployed following a specific network topology.

Although there are many existing honeynet solutions, a honeynet taxonomy that facilitates their classification and study has not been already proposed. In this paper, we propose such taxonomy of honeynet solutions and apply it to the existing honeynet solutions, helping to get a clear outline of honeynets architecture, and gaining insight of the honeynets technology. The taxonomy can be used by security experts to design the appropriate and optimized honeynet solution according to its security scenario.

The organization of the paper is as follows: in section 2, the classification scheme of the taxonomy of honeynet solutions is

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

proposed and each class is described in detail; in section 3, the new classification scheme of the taxonomy is applied to a number of existing honeynet solutions; in section 4, we analyze the classification of the honeynet solutions for exploring some possible honeynet solutions based on our taxonomy; in section 5, the state-of-the-art about honeynets is presented; finally, in section 6, the conclusions are summarized and some future work is suggested.

II. Classification Scheme of Taxonomy of Honeynet Solutions

A honeynet solution should contain three main capabilities: data control, data capture and data collection.

Data control is a set of measures to mitigate the risk that the adversary uses the compromised honeypot to attack other non-honeynet systems, such as other organization system on the Internet. Therefore, outbound attacks must be controlled in order to protect the non-honeynet systems.

The purpose of data capture is to log all the attacker's activity for later investigation. Three critical layers of data capture were identified: firewall logs (inbound and outbound connections), network traffic (every packet and its payload as it enters or leaves the honeynet), system activity (attacker's keystroke, system calls, modified files, etc.).

Data collection includes all the means needed to securely forwarding all of the captured data from distributed systems to a centralized secure data collection point. Due to the fact that honeypots are themselves insecure systems, the captured data must be centralized as soon as possible to an external secure system. Besides, centralizing the data in a system makes its management easier.

This section presents the classification scheme of the taxonomy of honeynet solutions proposed. Figure 1 illustrates a graphical representation of the classification scheme.

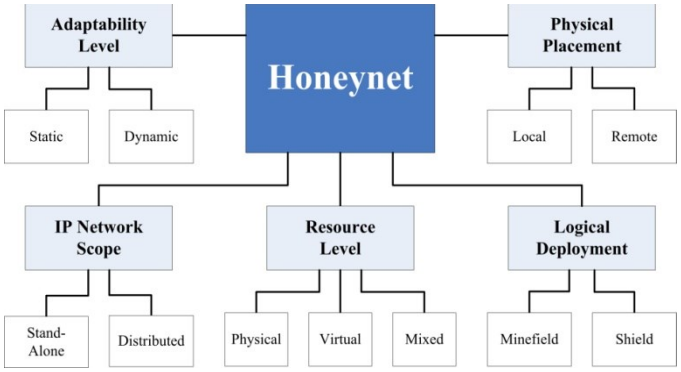

Fig. 1. Classification Scheme of Taxonomy of Honeynets

The classification scheme comprises five main criteria: resource level, adaptability level, IP network scope, physical placement, and logical deployment. A detailed description of each criterion follows.

A. Resource Level

Resource level is a criterion used to classify the honeynet solutions into physical, virtual, and mixed.

1) Physical

A physical honeynet consists of several honeypot systems running directly on physical machines following certain network topology. Being the honeypots physical means they are high-interaction honeypots that can get a high level fidelity of data capture, but with a higher resource cost.

2) Virtual

A virtual honeynet is made up of virtual honeypots following certain network topology. The virtual honeypots can be hosted by one or more physical machines. A virtual honeynet can be categorized into two types: self-contained virtual honeynet and hybrid virtual honeynet.

Self-contained virtual honeynet – A self-contained virtual honeynet is an entire honeynet system deployed onto a single physical system. The diagram of a self-contained virtual honeynet is showed in Fig. 2.

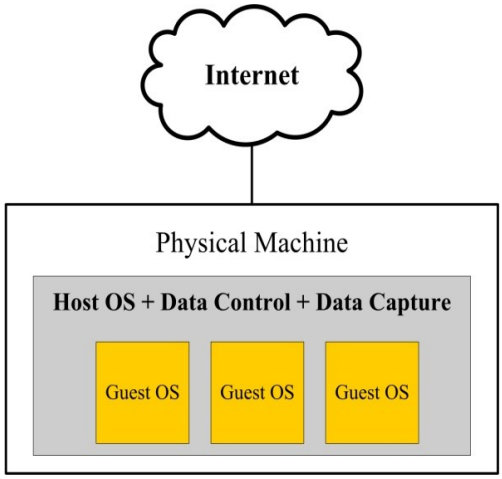

Fig. 2. Self-contained virtual honeynet

The steps needed to create a self-contained virtual honeynet follow. Firstly, the host operating system must be installed on the physical system. Then, the virtualization software has to be installed upon the OS. Lastly, the virtual machines running the guest OSs are created, controlled by the virtualization software. In this case, the host OS runs the data control and data capture functions of the honeynet, acting also as the gateway of each guest OS.

Hybrid virtual honeynet – Contrary to self-contained virtual honeynets, in a hybrid virtual honeynet the data control and data capture functions are implemented in a physically separate machine from the one running the virtual honeypots. Fig. 3 exhibits an overview of a hybrid virtual honeynet architecture.

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

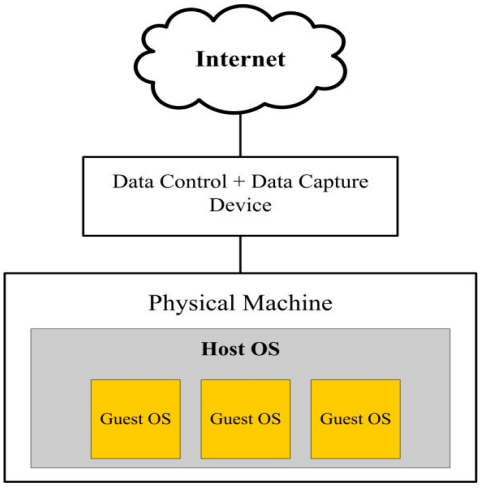

Fig. 3. Hybrid virtual honeynet

As can be seen, the first physical device is an isolated system that executes the data control and data capture functions of the honeynet. The second physical computer is used to hosts the virtual machines implementing the virtual honeypots, each one running their own operating system.

3) Mixed

A mixed honeynet is a honeynet that consists of virtual and physical honeypots deployed following certain network topology. Mixed honeynets can get a good balance between resource efficiency and service fidelity.

B. Adaptability Level

Adaptability level refers to the capability of dynamically modifying the configuration and topology of a honeynet. From the adaptability level point of view a honeynet can be static or dynamic.

1) Static

Traditionally, security experts determine the configuration of the honeynet beforehand and then deploy it. However, under some circumstances, the honeynet has to be modified, changing honeypots configuration or adding or deleting honeypots. A static honeynet is a honeynet without the capability of being modified or reconfigured. The main drawback of static honeynet is clear: if one honeypot needs to be reconfigured, the whole honeynet has to be redeployed. In other words, the security experts have to modify the configuration, stop the whole honeynet and restart it again. Static honeynets are not adequate, for example, for systems that require honeynet reconfigurations as a real time response to network events.

2) Dynamic

Dynamic honeynets are able to change the configuration of their honeypots, their topology or adding or deleting honeypots dynamically in real-time as a result of a management request or a response to a network event. Thus, dynamic honeynets can be very useful to create honeynets that are able to react, for example, to events triggered by intrusion detection systems and reconfigure themselves according to the characteristics of the attacker behavior. Dynamic honeynets overcome the limitations of static honeynets, allowing their partial reconfiguration without needing to restart and redeploy the whole honeynet.

C. IP Network Scope

The IP network scope criterion of a honeynet indicates how the IP addresses are assigned to honeypots. It can be classified into two categories: stand-alone and distributed.

1) Stand-Alone

In a stand-alone honeynet all the honeypots use IP addresses from a common IP network prefix, and they share one security toolkit that can capture all of the data from the whole honeynet. Some organizations only have one single honeynet. However, some others can also deploy several stand-alone honeynets, but they don't need to forward the data from multiple stand-alone honeynets into a central data server.

2) Distributed

A distributed honeynet always covers multiple networks concurrently in order to provide a greater ability to capture suspicious network events. This deployment applies only to organizations that have multiple honeynets in distributed environments. Organizations that have multiple honeynets logically or physically distributed around the world have to collect all of the captured data and store it in a central location by secure approaches.

D. Physical Placement

Physical placement indicates the physical location of the honeynet. It can be divided into two categories: local and remote.

1) Local

A local honeynet is a network of honeypots located within a physical limited area such as a computer laboratory, office building or organization using network media.

2) Remote

A remote honeynet does not impose any physical placement limitation to the honeypots. It can allow a group of physical remote honeypots integrated into one production network by tunnel technology, i.e. GRE tunnel, thus the honeynet can be located in any place of the world.

E. Logical Deployment

Logical deployment refers to the logical relationship between the honeynet and the production network. The logical deployment strategy can be classified into two categories: minefield and shield.

1) Minefield

As we all know, landmine will explode upon contact. Similarly, honeynets using the minefield strategy passively capture data upon interaction. In a minefield deployment, honeypots are often logically deployed among production systems, possibly cloning some of their real data. In other words, the honeypots are logically integrated into the production network. Thus, the honeynet is used to handle any kind of traffic including regular traffic and intrusion traffic. In a minefield deployment, the IP addresses assigned to the honeypots are chosen from the unused IP address of the production network.

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

2) Shield

Using the shield deployment strategy, the network of honeypots always acts as a mirror of the production network. This strategy allows intrusion detection system (IDS) or anomaly detection system (ADS) to investigate the network traffic based on destination port numbers. If the traffic is interesting, it will be redirected into the honeypot shield, protecting in this way the real system from the attack, as well as allowing to analyze the attacker behavior.. On the other hand, the honeynet shield and the production network are coupled, tightly or loosely. Thus the honeynet can reside in the same address space of the production network or resides on another subnet alongside the production network. It must use some approaches to redirect the interesting traffic such NAT or GRE tunnel.

III. Classification of Honeynet Solutions

A honeynet always takes multiple deception hosts (single honeypots), and turns them into an entire deception network. A typical honeynet may consist of many facades (because they are light-weight and reasonably easy to deploy), some instrumented systems for deep deception, and possibly some sacrificial lambs (conventional computers running as honeypot). Actually, there are several kinds of typical honeynet solutions, such as Gen III honeynet, shadow honeypots, hybrid system etc. In this section, we will decompose some honeynet solutions depending on our classification scheme of taxonomy of honeynet solutions.

A. Gen III Honeynet

A typical Gen III honeynet is made of a containment gateway called Honeywall [4] and computer systems installed Sebek/Qebek [5] acting as honeypots controlled by the honeywall. The GenIII honeynet architecture is the same with the typical GenII honeynet architecture [6] shown in Figure 4.

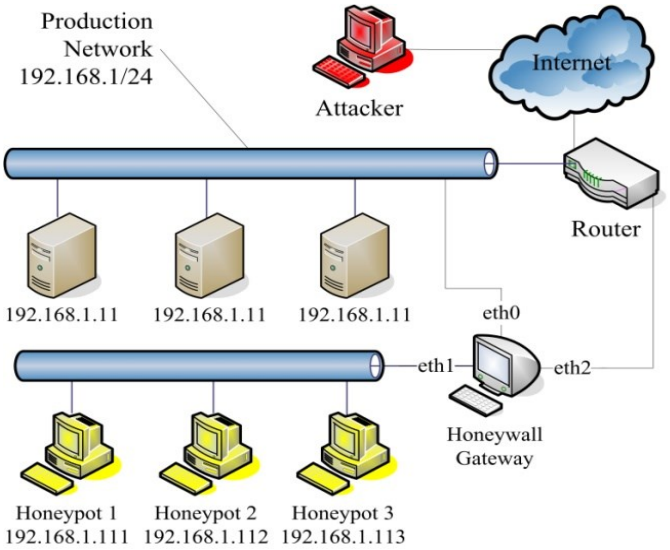

Fig. 4. An overview of Gen III Honeynet architecture

The most important tool in Gen III honeynet is the honeywall. It is traditionally a layer 2 bridging device employed to monitor the unauthorized traffic directed to honeypots. It can capture and collect the data and contain the attack to avoid compromising other systems from the honeypots.

Resource Level – When the Gen III honeynet was devised, it was installed on multiple separate physical machines. As such, we classify its resource level as physical.

Adaptability Level – The whole Gen III honeynet had to be configured manually beforehand by a security expert since it traditionally uses separate physical machines without the capability of dynamic configuration. As such we classify its adaptability level as static.

IP Network Scope – Gen III honeynet traditionally deploys on a single production network assigned by the unused IP address. Thus, its IP network scope is stand-alone.

Physical Placement - In Gen III honeynets, the honeypots are bridged into the production network. Thus, these honeypots can receive the interesting traffic and prevent the production systems from being attacked by confusing the adversary. As a result, we classify its physical placement as local

Logical Deployment - The honeypots integrate into the production networks and are assigned address from the unused IP addresses set. They accept all kinds of input traffic. Thus, Gen III honeynet uses the minefield deployment strategy.

B. Shadow honeypot

The shadow honeypots [7] based honeynet solution segments anomalous traffic from regular traffic and transfer the anomalous traffic to a honeynet made of shadow honeypots. If the traffic previously marked as suspicious by the anomaly detection system is confirmed as an attack by the shadow honeypots, it is captured for later analysis. However, if the shadow honeypot detects that the traffic classified as an attack is legitimate traffic (a false positive), the traffic will be handled by the system, and the information about this false positive will be used to update and improve the detection system..

Figure 5 shows an example scenario using the shadow honeypots approach to protect several servers.

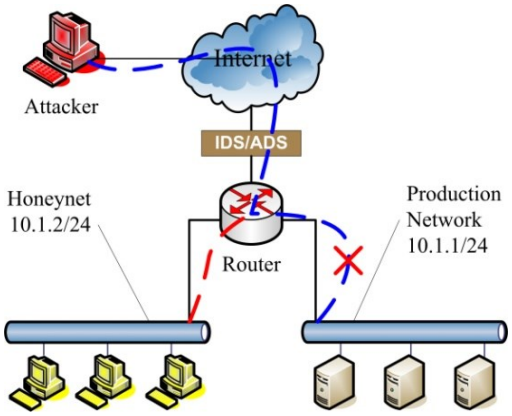

Fig. 5. Using shadow honeypots as shield

Resource Level – The honeynet solution where shadow honeypots were deployed did not mention they were using virtual machines. We therefore classify its resource level as physical.

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

Adaptability Level – Though shadow honeypots can be created automatically by the tool, we were not able to devise a test case that can dynamically reconfigure shadow honeypots in real time. As such we classified its adaptability level as static.

IP Network Scope – Shadow honeypots were placed on a single network. As such, the shadow honeypot was classified as stand-alone.

Physical Placement - The shadow honeypots and the production servers can be coupled tightly by using the same address or coupled loosely by physically deployed alongside the production servers. As a result, we classified its physical placement as local.

Logical Deployment - The regular traffic to and from the server is not affected, but any suspicious traffic destined to the server is instead handled by the shadow honeypots. Thus, we classified its logical deployment strategy as shield.

C. Potemkin

Potemkin [8] is a hybrid system based honeynet solution. It shares similar ideas with Collapsar [9] to provide high-interaction honeypots to very large address spaces. Figure 5 presents Potemkim's hybrid honeynet solution based on a honeyfarm.

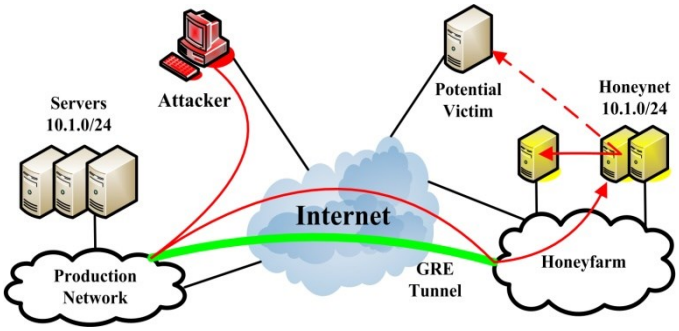

Fig. 6. A hybrid honeynet solution

A honeyfarm is a centralized group of honeypots that can be located anywhere in the world. The traffic is redirected from the distributed capture points to the honeyfarm by using IP tunnels (like GRE), in order to centralize data capture and facilitate the analysis and the correlation of events. Honeyfarm provides many synergies that help to mitigate many of the deficiencies of traditional honeypots. For instance, honeypots often restrict outbound traffic in order to avoid attacking non-honeypot nodes. However, this restriction allows honeypots to be identified by an attacker. Honeyfarm can be used as redirection points for outbound traffic from each individual honeypot. These redirection nodes also behave like real victims.

Potemkin consists of a network gateway and VMM (virtual machine monitor) based high-interaction virtual honeypots. The gateway acts as an agent and takes responsibility to send traffic to a honeyfarm server. The routers all over the Internet are configured to GRE tunnel an address prefix to the gateway. Besides, Potemkin's gateway also achieves internal reflection to contain outgoing attack via redirecting the traffic to a new created virtual honeypots. The gateway instructs VMM (virtual machine monitor) that runs on each physical server to create a new high-interaction virtual honeypot on demand for each active destination IP address. The VMM is responsible for managing high-interaction virtual honeypots. Thus, if one high-interaction honeypot is idle, the VMM will destroy it and reclaim the resources when instructed by the gateway.

Because the high-interaction virtual honeypot will waste CPU and memory resources of the host when high-interaction virtual honeypot has vulnerability but no one exploits it. Potemkin uses dynamic creating high-interaction virtual honeypots on physical servers to achieve efficient resources usage.

Resource Level – Potemkin uses virtual servers to create new high-interaction honeypots on demand for each active destination IP address. As such, we classify its resource level as virtual.

Adaptability Level – Because the VMM can create virtual honeypots on demand for each destination IP address and also destroy it if it is idle. As such we classified its adaptability level as dynamic.

IP Network Scope – Potemkin can allow a group of physical remote honeypots integrated into distributed production networks by GRE tunnel, using unused IP address of distributed production networks. As such, we classified its IP network scope as distributed.

Physical Placement - Since the benefit of GRE tunnel the high-interaction virtual honeypots can be put any place. Thus the placement of the honeynet is arbitrary. As a result, we classify its physical placement as remote.

Logical Deployment - The high-interaction virtual honeypots theoretically accept any kind of input traffic from the production network where the virtual honeypots logically deployed by GRE tunnel with the unused IP address of the production network. As such, Potemkin uses minefield deployment strategy.

D. Anti-Phishing framework based bank honeynet

Anti-Phishing framework [14] devised a new honeypot based way to protect the e-banking system. In this novel anti-phishing framework there are four different kinds of honeypots: phoneypot; phoneytokens; spamtraps; and phoneybots. The anti-phishing framework deploys the honeypots among the production network and accepts any kinds of traffic. The normal honeyed e-banking system runs on a physical machine, and a phishing detector is embedded into the honeyed e-banking system to automatically detect the phishing attack. It has all the functions of a normal e-banking system. The honeyed e-banking system always passively waits phisher to log in. It can monitor online transactions. The phishing detector in the e-banking system can issue an alert once it detects a phisher tries to transfer money from a phoneytoken to a non-phoneytoken account.

The phoneybots running on virtual machine are used to actively feed phoneytokens to pharmers and phishing malware. It mimics real user's behavior to access the real e-banking system in order to confuse the adversaries. The phoneybot can execute automatically online transaction with e-banking system

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

from time to time following the average behavior of all bank customers.

Phoneytokens are fake credentials which are used by the phoneybots, the spamtraps and the honeyed e-banking system. The phoneytoken is just data information that can be accessed by the phisher. It is used in online transaction or phishing email. It is a face copy of real credential.

The spamstraps are used to attract phishing emails and submit phoneytokens to phishing sites. The framework set up virtual machine to run spamtrap. The spamtrap can submit the phoneytoken to the phishing site by the human manager when it is deceived by phishing email. It running on virtual machine mimics real user's behavior to click phishing link.

Resource Level –The normal honeyed e-banking system runs on a physical machine but the phoneybots and spamstraps run on virtual machines. As such, we classify its resource level as mixed.

Adaptability Level – The anti-phishing framework does not provide the capability of dynamic configuration. We therefore classify its adaptability level as static.

IP Network Scope – The honeyed e-banking system, phoneybots and spamstraps were placed on a single network. As a result, we classify its IP network scope as stand-alone.

Physical Placement - Since all of the honeypots are physically deployed in a bank, the physical placement is a limited area. As a result, we classified its physical placement as local.

Logical Deployment – The honeypots are deployed among these production systems and used to confuse the adversary. As such, anti-phishing framework based honeynet solution uses minefield logical deployment strategy.

IV. Analysis for Exploring the Design Space of Honeynet Solutions

A. Analysis

In our proposed taxonomy of honeynets, there are five criteria. In this section, we analyze the advantages and disadvantages of each value of every criterion.

First of all, from the point of view of the resource level we can use physical machine and virtual machine to deploy honeynet. Physical honeynets are not used nowadays because of their cost in terms of resources needed. Besides, virtual machines can provide almost any kind of service a physical machine can offer. Even the Gen II and Gen III honeynet architectures can be also deployed on virtual machines [10] [11]. The main drawback of a virtual honeynet is probably the performance of the service provided by the virtual machine. The adversary can use several tests to detect the virtual honeynet, one of which is to test the service performance. Thus, for a virtual honeynet, the security experts must pay attention to provide very similar services.

Secondly, the adaptability level implies that all physical honeynets are classified as static honeynets, while the capability of dynamic configuration is always provided by the virtualization software. But then again, it does not mean that any kind of virtualization software or virtualized tool can provide the capability of dynamic configuration. Dynamic virtual honeynet profit from its flexible configuration and deployment has become more and more popular. For example, the widely used famous low interaction virtual honeypots framework, Honeyd [12] can simulate multiple honeypots simultaneously following certain network topology. Nevertheless, static honeynet also has advantages. It is easy to facilitate a static honeynet and it is suitable for many security static scenarios.

Thirdly, for IP network scope, we have two kinds of honeynets, and both of them have strong points and shortcomings. Stand-alone honeynet is deployed on a single network, thus it is easy to deploy and data collection is not a complex task, either. However, it lacks the scalable view of multiple networks. Distributed honeynets include honeypots across multiple networks concurrently. We know that most automated malware can attack a large range of networks but some advanced attack only focus on several specific networks. Thus the distributed honeynets can provide a global view of interesting traffic. The main drawback of distributed honeynets is the data collection. It always needs the tunnel technology such as GRE tunnel to securely collect the data from different honeynets into a central point. It is a complex and resource cost task.

Fourthly, the physical placement is another important aspect we must consider. Local honeynet has the placement limitation which means the honeypots have to physically put in a relative centralized area. Remote honeynet has arbitrary physical placement. The remote honeynet can use GRE tunnel to integrate into any production network but the service performance decrease may arouse the attacker's suspicions.

Fifthly, the two logical deployment strategies both have their own concern. The minefield honeynets are suitable for preventing production honeypot from attack. While the shield honeynets are always used to detect unknown attack.

B. Exploring the Design Space of Honeynet Solutions

Depending on the analysis of our classification scheme of taxonomy, we can try to explore the future honeynet solutions.

Firstly, the virtualized tool which can provide the capability of dynamic configuration and deployment to virtual honeynet has a large design space. Because it does not only hit the historical requirement since the physical honeynets have been out of date, but also can provide the capability of dynamic configuration and deployment which are more and more important for current network environment.

Secondly, hybrid honeynet is an appropriate solution for IP network scope, since it can cover both the stand-alone deployment and distributed deployment. It benefits by the combination of low-interaction and high-interaction honeypots, thus it can gain a good balance among scalability, fidelity and performance. Although there are a number of hybrid honeynet architectures [8][9][13], due to the technology limitation most of them are static. Thus, the hybrid dynamic honeynet solution is still desired to devise.

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

Thirdly, the honeynet deployment strategy consists of two criteria, physical placement and logical deployment. Figure 7 depicts a coordinate based representation of honeynet deployment strategy.

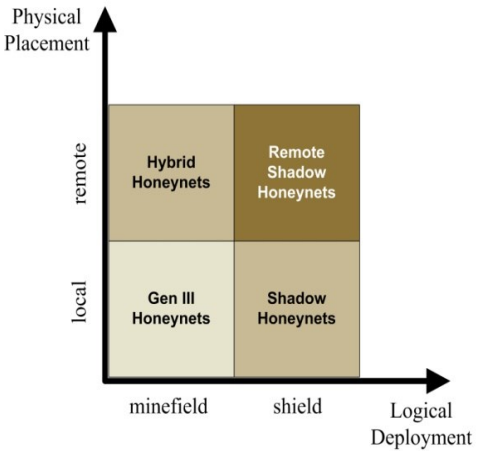

Fig. 7. Honeynet Deployment Strategy

The coordinate based diagram presents two criteria to classify honeynet solutions using different deployment strategies. The first criterion is the horizontal axis namely logical deployment. The second criterion is the vertical axis called physical placement.

We can use these two criteria to classify different honeynet deployment strategies. From the classification of honeynets, we know that Gen III honeynets are local minefield honeynets, hybrid honeynets are remote minefield honeypots, and shadow honeynets are local shield honeynets. But there is rare honeynet deployment standing on remote shield, which we provide the terminology namely remote shadow honeynets. Thus, there is a large design space for remote shadow honeynets. Actually, we do have remote shadow honeynets in practical security research. Figure 8 shows an example of remote shadow honeynet deployment.

The remote shadow honeynet could be an entire copy of the potential victim network in order to keep a high level of camouflage and fidelity. The honeynet using remote shadow deployment strategy is always a representative honeynet, which has all of the characteristics of the potential victim network even the same IP addresses. The remote shadow honeynet can be used for online attack catch or offline security research.

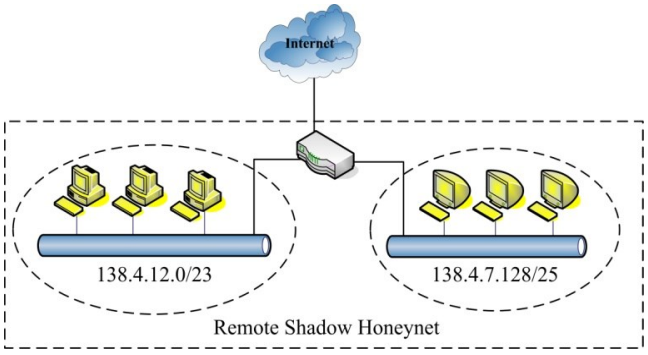

Fig. 8. Remote Shadow Honeynet Deployment

For online attack catch the functionality is very similar to the local shadow honeynets. But it must use a different IP network address space and apply a tunnel technology such as GRE to redirect the interesting traffic to the remote shadow honeynet.

On the other hand, this representative honeynet can be established in an off-line lab environment standing alone from the target production network. As such, remote shield honeynet deployment means that the honeynet is completely independent from the production network implying its high position flexibility. The offline research always investigates certain impact to the production result from specific harmful traffic.

Besides, the remote shield honeynet also can be deployed in an isolated environment for the attacking and defending games and experiments. For example, the CTF (Catch the Flag) competition always uses this kind of honeynet. In a CTF competition, there are several teams belonging to several subnets, and each team has the honeypot systems with the same vulnerabilities. The job of each team is to find these vulnerabilities, to patch them on their own honeypots but to use them to exploit the other honeypots of the opposite team.

V. State of the Art

As stated before, this paper defines a honeynet as a network of honeypots following a certain network topology. However, some early articles might not agree this definition. For example, in the Symantec deception server experience [15], the security experts presented three deployment strategies, minefield, shield and honeynet. The terminology honeynet, however, was used as a kind of deployment strategy. Actually, this paper provided a good practical experience about honeypots instead of theoretical classification of honeypot systems. The deployment strategies proposed in this paper are very interesting and can give security experts inspirations.

The book titled “Virtual Honeypots” [16] describes various virtual honeypot systems. The classification scheme of virtual honeypots in the book is the interaction level. The authors provided many details including the installation and configuration guides on a number of low-interaction and high-interaction honeypots. Moreover, they also described several hybrid systems that are very typical honeynet solutions. A survey of recent advances and future trends in honeypot research [17] investigated around 60 papers after year 2005. The survey is quite complete. From this survey the security experts can quickly get knowledge of various honeypot researches. In this survey, some honeypot researches are honeynet solutions. However, the author didn’t distinguish the honeynet solutions from the honeypot systems.

Another famous book from the Honeynet Project called “Know Your Enemy” [18] described the development history of Gen I, II, and III honeynet. It also presented the concept of virtual honeynet in detail. In addition, we also can find the research on distributed honeynets. Although the book didn’t provide us a theoretical classification of honeynets, we can gain affluent knowledge of honeynets from it.

978-1-4673-7606-8/15/$31.00 ©2015 IEEE

VI. Conclusion and Future Work

In this paper, we proposed a new classification scheme to create a taxonomy of honeynet solutions. We provided five criteria that conform to common honeynet terminology to cover the current characteristics of honeynet technology. We believe the taxonomy of honeynets is comprehensible to help the security experts applying our taxonomy to gain further insights into the honeynet technology.

Furthermore, we analyze the strong points and weaknesses of the values in every criterion in order to explore the design space of honeynet solution. We found that the virtualized tools that can provide dynamic configuration and deployment for virtual honeynets have a large design space. What's more, we predict the design of hybrid honeynet solution will keep on developing since hybrid honeynets combines both the advantages of low-interaction honeypots and high-interaction honeypots but offsets the disadvantages on both of them. In addition, we discover the remote shadow honeynet deployment will become popular since it has practical value.

In the future, we will apply our taxonomy to more existing honeynet solution and explore much more design space of honeynets in order to provide optimized honeynet solutions. We hope security experts can get benefit from our work.

Acknowledgement

This research is supported in part by National Natural Science Foundation of China (No. 61440057,61272087,61363019 and 61073008), Beijing Natural Science Foundation (No. 4082016 and 4122039), the Sci-Tech Interdisciplinary Innovation and Cooperation Team Program of the Chinese Academy of Sciences, the Specialized Research Fund for State Key Laboratories. It also has been partially funded with support from the Spanish MICINN (project RECLAMO, Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management, with codes TIN2011-28287- C02-01 and TIN2011-28287-C02-02) and the European Commission (FEDER/ERDF).

References

[1] Stoll, C., *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*, New York: Pocket, 1990.

[2] Lance Spitzner. (2001, Oct. 10). *The Value of Honeypots, Part One: Definitions and Values of Honeypots* [Online]. Available: http://www.symantec.com/connect/articles/value-honeypots-part-one-definitions-and-values-honeypots

[3] Honeynet Project. (2001, April 26). *Know Your Enemy: Honeynets* [Online]. Available: http://www.symantec.com/connect/articles/know-your-enemy-honeynets.

[4] Honeynet Project. (2005, August 17). *Know Your Enemy: Honeywall CDROM Roo* [Online]. Available: http://old.honeynet.org/papers/cdrom/roo/index.html.

[5] Honeynet Project. (2010, Nov. 03). *Know Your Tools: Qebek – Conceal the Monitoring* [Online]. Available: http://www.honeynet.org/papers/KYT_qebek.

[6] Honeynet Project. (2005, May 12). *Know Your Enemy: GenII Honeynets* [Online]. Available: http://old.honeynet.org/papers/gen2/

[7] Anagnostakis, K. G., et al. "Detecting targeted attacks using shadow honeypots." Proceedings of the 14th conference on USENIX Security Symposium, pp. 129-144, ACM, 2005.

[8] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Scalability, fidelity, and containment in the Potemkin virtual honeyfarm. In SOSP '05: Proceedings of the Twentieth ACM Symposium on Operating Systems Principles, pp. 148–162, New York, 2005. ACM Press.

[9] X. Jiang and D. Xu. Collapsar: A VM-based architecture for network attack detention center. In Proceedings of the USENIX Security Symposium, August 2004.

[10] Lok Kwong Yan, "Virtual honeynets revisited," Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC , vol., no., pp.232,239, 15-17 June 2005

[11] Abbasi, F.H.; Harris, R.J., "Experiences with a Generation III virtual Honeynet," Telecommunication Networks and Applications Conference (ATNAC), 2009 Australasian, vol., no., pp.1,6, 10-12 Nov. 2009.

[12] N. Provos, "A Virtual Honeypot Framework," SSYM'04 Proceedings of the 13th conference on USENIX Security Symposium, volume 13, 2004.

[13] Lengyel, T.K., Neumann, J., Maresca, S., Payne, B.D., Kiayias, A.: Virtual machine introspection in a hybrid honeypot architecture. In: Proceedings of the 5th USENIX Conference on Cyber Security Experimentation and Test, CSET 2012, p. 5. USENIX Association, Berkeley (2012)

[14] Shujun Li and Roland Schmitz, "A Novel Anti-Phishing Framework Based on Honeypots," Proceedings of eCrime Researchers Summit, October 2009, pp. 1-13.

[15] Hernacki, B., Bennett, J., and Lofgran, T. "Symantec Deception Server: Experience with a Commercial Deception System," Proceedings of the 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15 - 17, 2004.

[16] Niels Provos and Thorsten Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*, 1st ed., Boston: Addison-Wesley Professional, 2007

[17] Matthew L. Bringer, Christopher A. Chelmecki, Hiroshi Fujinoki, "A Survey: Recent Advances and Future Trends in Honeypot Research," IJCNIS, vol.4, no.10, pp.63-75, 2012.

[18] Honeynet Project, *Know Your Enemy: Learning about Security Threats*, 2nd Edition, Boston: Addison-Wesley Professional, May 27, 2004

978-1-4673-7606-8/15/$31.00 ©2015 IEEE