MBNA Publishing House Constanta 2021

Proceedings of the International Scientific Conference SEA-CONF

SEA-CONF PAPER • OPEN ACCESS

Honeypot Management System Based on LXC Container Virtualization

To cite this article: Radu MOINESCU, Ciprian RĂCUCIU, Sergiu EFTIMIE, Dragoș GLĂVAN and Sorin BÎRLEANU, Proceedings of the International Scientific Conference SEA-CONF 2021, pg.212-218.

Available online at www.anmb.ro

ISSN: 2457-144X; ISSN-L: 2457-144X

doi: 10.21279/2457-144X-21-028

Honeypot Management System Based on LXC Container Virtualization

Radu MOINESCU, Ciprian RĂCUCIU, Sergiu EFTIMIE, Dragoș GLĂVAN, Sorin BÎRLEANU

Military Technical Academy "*Ferdinand I*"
radu.moinescu@gmail.com

Abstract. The paper proposes a general architecture for a honeypot system based on LXC container virtualization technology. The developed model uses the virtual container technology, deploying a virtual copy of the real network including the network services. Attackers, while interacting with the honeypot system, presume that they interact with the real network. The security administrator of the information and communications system analyzes the attacker's actions in real time and obtains the information about their priority targets, the tools they use and the vulnerabilities of the network elements. This allows the administrator to quickly take measures in order to increase the network security and avoid its compromise.

1. Introduction

Over the last decade, we have witnessed a multitude of well-orchestrated cyber-attacks against state, military and industrial entities. The main objective of most of these attacks was to leak large amounts of data. Although organized by various threat agents, these attacks showed a high degree of expertise and considerable resources, which allowed the use of several attack vectors to achieve the intended objectives. Traditional means of information protection, such as antivirus software, firewalls and intrusion detection and prevention systems, do not guarantee absolute security of information and communication systems.

The main indicator of the security of an IT&C system within an organization in the conditions of a cyber-attack is resilience (T_z). The IT&C system must be able to adapt quickly and/or recover from any type of interruption, in order to continue operating at an acceptable level, taking into account the objectives and impact that the interruption has on the security of the IT&C system. In the context of the business continuity process, the security measures needed to support the resilience of an IT&C system, including plans and procedures, are identified through a risk assessment and impact assessment process and are reflected in the business continuity plan, prepared at the level of the organization. [1] While the risk assessment process identifies critical functions and assets, as well as the risks that may disrupt the organization's mission, the impact assessment identifies potential damage or loss in the event of an incident, the form that the damage may take, and how which can evolve over time. From the design phase of the IT&C system, the time required to return it to normal operation (T_p) as a result of any type of interruption is established. To assess the security of information stored, processed or transmitted through the IT&C system, in the event of a cyber-attack, the following expression can be used

$$P_z(T) = P\left(T_z \geq T_p\right)$$

The effectiveness of the measures taken to ensure the security of an IT&C system may decrease as new threats arise, therefore the value of the $P_z(T)$ indicator decreases. In order to maintain an optimal

level of system security, it is necessary to carry out a predictive check, taking into account the current situation.

Most organizations use a vulnerability table established in the initial stage in the method of forecasting the security status of proprietary information and communications systems. This approach is convenient in terms of analyzing the consequences of implementing known vulnerabilities and predicting the subsequent operation of systems, but the disadvantage of this method is that it uses a fixed set of known vulnerabilities. The residual security risk is the risk that remains after the implementation of security measures in an IT and communications system, given that not all threats can be counteracted and that not all vulnerabilities can be eliminated or reduced. Threats and vulnerabilities are dynamic, so the residual risk is subject to change. Attackers find new ways to bypass the information security system, to discover new vulnerabilities in programs and protocols. Because of this, the risk will be managed throughout the life cycle of the information and communication system, which involves the allocation of adequate resources to carry out the risk management process.

One of the approaches that offers the ability to anticipate cyber-attacks is the use of information security tools that simulate the operation of real elements of information and communication systems, the so-called honeypot systems. Honeypot technology involves giving an attacker access to a known vulnerable resource. Once the attacker gains access, he will perform certain actions, waiting for a response from the system. Depending on how the system and the attacker interact, two different types of honeypot systems can be distinguished:

- systems with a high degree of interaction - completely mimics the behavior of the real system;
- systems with low degree of interaction - have limited functionality. [2]

Highly interaction honeypot systems are most often an individual host or device located in an organizational network, but not participating in information processes. Thus, accessing such a resource from the outside can be considered as a breach of the security of the network infrastructure. Because such systems provide the attacker with the full functionality of real systems, they can also be compromised and used as auxiliary means to penetrate the real network and for other purposes, such as botnet formation. It is therefore necessary to ensure the isolation of such systems.

Typically, highly interactive honeypot systems are used to obtain various information about the types of attacks, the attacker's target, and possible vulnerabilities in the organization's infrastructure.

Honeypot systems with a low degree of interaction use various mechanisms to simulate the behavior of a real computer system. It can be: an operating system, a web server, a database server or even a regular network application.

Unlike those with a high degree of interaction, low-interaction honeypot systems do not involve the allocation of significant resources, are easier to configure and cannot be used by a potential intruder to attack the real resources of the organization's network. On the other hand, they are easily detected by the intruder, which is why such systems are better suited to counteract the various automated tools used by attackers than to obtain information about their actions.

The main advantages and disadvantages of each honeypot system are presented in Table 1.

Honeypot systems	**with low degree of interaction:**	**with a high degree of interaction:**
Advantages:	Easy setup Safety	Full functionality Difficulty in detection More useful information about the attack
Disadvantages:	Easy to detect Little useful information about the attack	Complexity of implementation and configuration Security risk

Table 1. Honeypot systems, advantages and Disadvantages

Another type of honeypot system, to which it is difficult to assign a certain class, because on the one hand, it offers the attacker the full function of a service and, on the other hand, they do not produce "real" actions. Such a system is called an interactive honeypot.

The aim of this study is to develop a method of managing information security by recognizing a cyber-attack and predicting the scenario of further development of the cyber-attack, taking into account the possible consequences when making decisions on preventive actions.

2. Proposed method for information security assurance based on an interactive honeypot system

2.1 System architecture

The use of container-based virtualization technology, and in particular the LXC implementation, allows the creation of virtual hosts that have the following properties:

- working speed, compared to a real system;
- fast creation, configuration and launch, compared to classic virtual machines;
- ability to centrally manage the activity and configuration;
- high level of security (support for the security mechanisms of the Linux operating system, the ability to work on behalf of a user with low privileges (unprivileged);
- the possibility of limiting the usage of resources.

Using these properties, we are able to create honeypot systems, simpler in requirements and configuration, systems with a high degree of interactivity than systems that use classic virtual machines. The generalized architecture of such a system is presented in Fig.1.

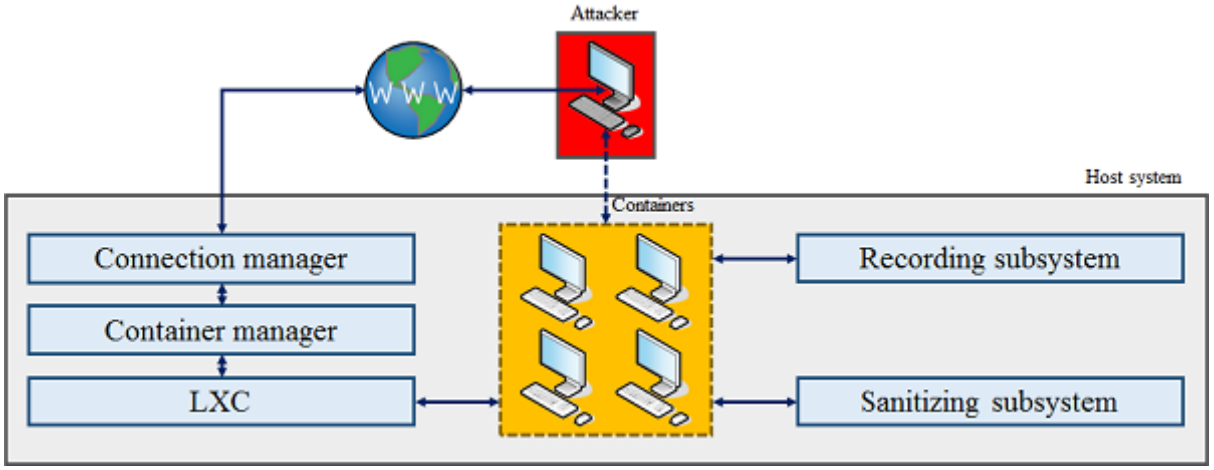

Fig.1. System architecture

The proposed system is a honeynet type, the difference from the classic concept is that not all the hosts, that are part of the honeynet, exist and work at a specific point in time. Because hosts are containers, the existence of a host means the presence of the container created according to the host parameters. The identification of the hosts will be easily done through their IP addresses in the internal network. The idea of building such a Honeypot system is to create an illusion of an intruder operating in an organizational network of any size and complexity, using a reasonable amount of computing and memory resources.

All the honeypot system components shown in Fig.1 are described below:

- *connection manager* – the role of this component is to forward packets destined to hosts included in our honeynet, on container manager for further processing. For this to be possible, the host network adapter on which the system will run must be set to transparent (promiscuous) mode, meaning it accepts all packets that pass through the network segment to which the host belongs, not just the packets. intended for it;

- *container manager* – this component acts as an intermediate link between the attacker, LXC and containers. When receiving packets from an attacker, the container manager decides what to do: respond to the attacker on his ow, performs any operations on containers, or reset the connection. In this component, various transformations can be performed on the packages: changing the headers, sanitizing, etc., depending on the functionality offered by the system. This component is also responsible for the interaction between the attacker and the running containers, in fact, acting as a proxy server. In principle, an attacker may be able to interact with containers immediately after their creation, but this approach makes it impossible to have complete control over the interaction;
- *LXC* – is a background daemon that performs all container management activities: creating, configuring, deleting, and monitoring state;
- *Containers* – are virtual hosts running inside containers created through the LXC. The key element of this proposed honeypot system is the constantly changing composition of existing and running containers, the image presented to the attacker may be completely different from the current state of the system. Using this approach, we can avoid the need to create containers unnecessarily - as long as they are not used in a certain way;
- *Recording subsystem* – the role of this component of the system is to record all the actions of the attacker inside the honeynet. The presence of such a system is an integral part of any honeypot. Recording can be done using integrated operating system mechanisms, for example, syslog or specialized tools created for highly interactive honeypot systems, such as Sebek. It should also be noted that the recording of events can be done both "outside" - on the host, inside which the containers operate, and inside each container. Combined solutions are also possible;
- *Sanitizing subsystem* – the purpose of this system component is to free up resources. The system can monitor the status of containers, the number of active network connections and other parameters and, based on them, suspend the operation of containers that can be considered inactive. It can also carry out the removal of containers considered unused by certain criteria. An example of such a criterion is, for example, a long time since the last launch of the container.

3. Scenario

3.1 Initial configuration

The connection manager is configured so that an attacker has access to a specific host with open ports. Containers containing selected operating systems and certain sets of applications are provided. Knowingly vulnerable versions of applications and operating systems can be used to make it easier for an attacker to "break" the honeypot. The container manager configures the desired structure of the Honeynet, rendering all the hosts and their different parameters. The hosts are divided into two categories:

- launched together with the system;
- triggered upon access.

The hosts launched together with the system make up the initial configuration of the system. Such hosts are usually of the greatest interest to the attacker as they represent web or e-mail servers, file servers etc. Because an attacker is more likely to access these hosts, they must be in working order so that there are no suspicious delays to the attacker on first access.

The category of hosts triggered on access are targets of low importance to the attacker and are represented, for example, by workstations. They do not need to be started with the Honeynet system because the attacker may not connect to them

3.2 Interaction with the attacker

The attacker attempts to connect to the honeypot by IP address.

The connection manager instructs the container manager to start the prepared container inside which the honeypot runs. At the same time, the container manager redirects all the attacker's packets destined for the honeypot to the container. The attacker, using one or another attack scenario, gains access to the honeypot.

3.3 Further actions of the attacker

Once it has gained access to the server, the attacker will try to ensure its persistence and gain control of the network. To achieve this, he, first of all, will need to obtain information about the structure of the network. To do this, the attacker will start a network scan (using, for example, a ping sweep tool).

Packages with echo requests should be directed to the container manager, where a decision will be made on the need to create a container. If the address of the workstation that is part of the honeynet structure is specified in the destination address, the appropriate container is created and launched, or launched if the container was created in advance. The container will respond to the attacker's echo request. Thus, while the attacker performs a network scan through the container manager, containers representing network workstations are created or launched, and the attacker can interact with the running containers.

4. The main features of the proposed honeypot system

The technologies used provide many opportunities for expanding the functionality of the system within the framework of the proposed architecture, as well as increasing the efficiency of some system components.

4.1 Using container images

To create a container in LXC, a base image must be specified. The role of the base image can be performed by either an existing (predefined) container or a custom one. There are three container management approaches that can be used to build honeypot systems:

- *use of pre-defined containers* – in this case, a container is created for each honeypot in advance, which is launched when needed. This approach is the fastest, but at the same time the most complex in terms of initial configuration and the most expensive in terms of resources, because it is necessary to keep all the containers created;
- *using a container image* – this approach is more resource efficient - multiple containers can be created based on a single image, but at the same time creating a container from an image takes much longer than launching an off-the-shelf container. If such a delay is acceptable, then this approach is a good alternative to the first;
- *cloning of existing containers* - this approach is similar to the previous one, but allows a more flexible configuration of the containers created. It is much more convenient to configure the host in a familiar operating system environment than to customize the image using the LXC API.

4.2 Host emulation

Within the proposed Honeypot system, the functionality implemented in HoneyD can be used, namely container emulation. When a packet is received from the attacker, the container manager may try to respond to the request on its own before creating a container to handle the request. Specifically, for example, when an attacker performs a network scan, it is sufficient to respond to the received request (e.g., ping, ARP, etc.) so that the attacker believes that such a host actually exists within the network. The container, on the other hand, will be created only when the attacker subsequently contacts the host. This will reduce the number of existing and working containers and significantly save resources.

4.3 Securing the proxy connection

If the attacker establishes a secure connection with a container, for example an SSH session, then the data transmitted in the communication process will not be available outside the container. This problem

can be solved using the man-in-the-middle attack technique. [3] The container manager will act as a proxy.

4.4 Securing the proxy connection

The proposed architecture involves the simplest network topology inside a honeynet - the honeypots are located on one network segment connected to the host interface which acts as a switch. In principle, special containers can be created that will act as various network devices (e.g., switches, routers etc.) and the created containers can be customized so that the honeynet has an arbitrary network topology of any complexities.

4.5 Container manager algorithm

The main functions of the container manager are:

- reception of incoming packets addressed to containers.;
- creation, launch and configuration of containers.;
- forwarding packages to the appropriate containers.

The container management algorithm is shown in Fig. 2.

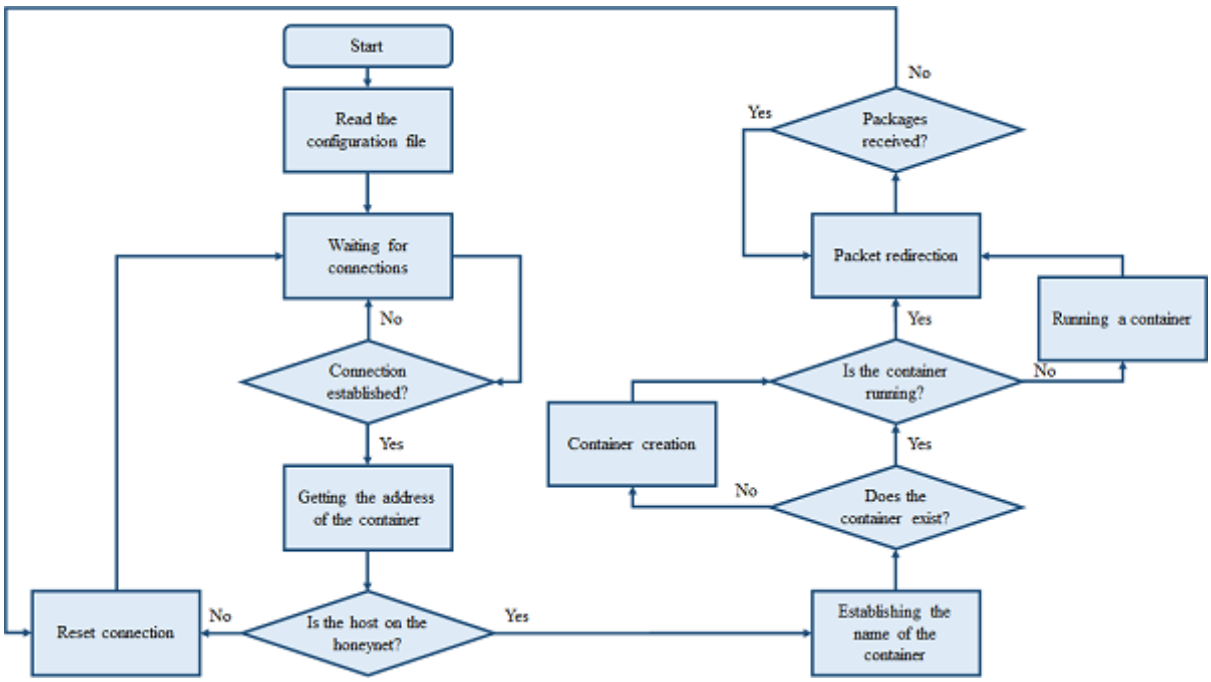

Fig.2. Container management algorithm

In essence, the container manager is a proxy server that, when a client connects to it, creates a container if necessary, and then redirects the packets received from the client to this container.

This developed proxy server has the following properties:

- each connection with the client leads to the creation of a new connection with the container;
- each package received from the client is redirected to the appropriate container, if it is defined in the configuration file;
- each packet received from containers is redirected to the client if there is an established connection;
- ensures simultaneous interaction with several clients.

5. Conclusions

In the modern realities of information security, dynamic highly interactive honeypot systems are the more relevant and useful in dealing with the emerging cyber threats.

Container virtualization has a sufficient level of security in comparison with classical virtualization and can be successfully used to build honeypot systems.

The novelty of the proposed method in comparison with existing solutions is as follows:

- processes have been added for obtaining information about the priority goals of the attacker, the means he uses, and the vulnerabilities of various network entities. For the technical implementation of the method, a proxy server is included, on which a virtual copy of a real network is deployed using container virtualization technology, including network services in conditions of limited resources;
- real-time analysis of the attacker's interaction with honeypot system allows the IT&C system administrator to respond in a timely manner to attempts to achieve the cyber-attack goal. The use of effective methods of analysis and forecasting increases the reliability of information, reduces the time of response to the attack on IT&C system;

The following advantages of the proposed method can be highlighted:

- the ability to timely detect an attempt of unauthorized access, produce statistics of attacks and determine ways to counteract them;
- centralized management of the system allows for rapid changes in its structure, adapt to changes in the real IT&C system, and analyze the data obtained;
- a proactive work model provides protection against new strategies for influencing the IT&C system;
- the isolation from the real IT&C system suggests the possibility of compromising a server with container virtualization without harming the real one.

References:

[1] The National Registry Office for Classified Information, *Principal Directive of 21 march 2014 regarding INFOSEC field - INFOSEC 2*, published in Monitorul Oficial no.262 of 10 april 2014, http://legislatie.just.ro/Public/DetaliiDocumentAfis/157287, accessed on March 9, 2021

[2] Mohssen MOHAMMED, Habib-ur REHMAN, *Honeypots and Routers Collecting Internet Attacks*, CRC Press publishing group, 2016, ISBN 978-1-4987-0220-1 (eBook - PDF)

[3] Andrew Michael SMITH, *Quick and Easy SSH MITM*, Andy Smith's Blog, Mar 13, 2014, https://andrewmichaelsmith.com/2014/03/quick-and-easy-ssh-mitm/, accessed on March 9, 2021