CERT Training Platform over the Event-Recordable Container

Namjun Kim[*]
Sejong University
Seoul, Republic of Korea
bunseokbot@sju.ac.kr

Chanmo Yang[*]
Sejong University
Seoul, Republic of Korea
yanglampp@gmail.com

Daebeom Cho
Sejong University
Seoul, Republic of Korea
tmdgns743@gmail.com

Seung Hyeon Geum
Sejong University
Seoul, Republic of Korea
parosproxy@gmail.com

Ki-Woong Park[†]
Sejong University
Seoul, Republic of Korea
woongbak@sejong.ac.kr

ABSTRACT

The current COVID-19 pandemic has resulted in many changes in the IT systems and services of institutions, which also heightened the concerns regarding the potential increase in intrusion incidents, especially when most works in institutions are performed at home. The need for pre-training against intrusion incidents has then become extremely necessary. Unfortunately, current learning methods in existing studies are insufficient for application in the present demand because these methods were originally designed for environments that are tailored-fit for learners and not in actual environments. This paper proposes a training system, namely, computer emergency response team (CERT), that can be specifically designed for learners in an institution to provide intrusion-incident cases using a Web-based training system. CERT can easily replicate the service or system in an institution to a honeypot environment to automatically collect and classify intrusion incidents using diverse evaluation criteria so that learning can be achieved from different perspectives. Hence, the institution operating service and system can easily be replicated. Artifacts of intrusion incidents are collected using the Docker container technology and event-recordable container, which are analyzed using a Web browser without installing a separate program. Thus, optimal learning results from the analysis of actual attacks are expected.

CCS CONCEPTS

• **Applied computing → Learning management systems**; *Evidence collection, storage and analysis*; Investigation techniques.

KEYWORDS

Training Platform, Event-Recordable Container, Digital Forensics

ACM Reference Format:
Namjun Kim, Chanmo Yang, Daebeom Cho, Seung Hyeon Geum, and Ki-Woong Park. 2020. CERT Training Platform over the Event-Recordable Container. In *2020 ACM International Conference on Intelligent Computing and its Emerging Applications (ACM ICEA '20), December 12–15, 2020, GangWon, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3440943.3444738

1 INTRODUCTION

Recently, a pandemic caused by COVID-19 has occurred worldwide. As of the writing of this paper, more than 35 million cases have already been confirmed around the world, and more than one million people have died from COVID-19. This unprecedented pandemic situation has resulted in many changes in the IT system and services of organizations and companies. In particular, potential security threats have significantly increased because of the remote work of employees [1]. Accordingly, the importance of training to prevent security threats in advance and establish countermeasures when they occur has increasingly become important.

However, only the challenge in the environment intended for learners was covered in the training system proposed in previous studies [3, 6, 8, 12]. Therefore, existing studies suffer from limitations in that they cannot be used for intrusion-incident response training for services operated by organizations or companies or for systems that need to be introduced. The computer emergency response team (CERT) training system presented in this paper easily replicates the service or system operated by organizations or companies to a honeypot environment to automatically collect and classify intrusion incidents.

In addition, it maximizes the learning effect by diversifying the evaluation criteria so that learning can proceed from different perspectives. Therefore, the service and system in operation can be easily replicated, and artifacts for intrusion incidents can be collected using the Docker container technology and event-recordable container. In addition, the artifacts collected in this manner are analyzed using a Web browser without installing a separate program, and the various methods for learning the result are explained.

This paper is structured as follows: introduction of the existing studies for existing intrusion-incident training method and examination of its limitations are presented in Chapter 2. Then, an explanation of the event-recordable container used to solve the existing limitations and the classification method of artifacts collected in the process are provided in Chapter 3. Presentation of an intrusion-incident training system that introduces the classified challenges is made in Chapters 3 and 4. The conclusion is given in Chapter 5.

[*] Both authors contributed equally to the paper.
[†] Corresponding author.

ACM ICEA '20, December 12–15, 2020, GangWon, Republic of Korea
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8304-2/20/12.
https://doi.org/10.1145/3440943.3444738

2 RELATED WORKS

A previous study that previously proposed an intrusion-incident education method using honeypot is introduced in this chapter. Romney et al. [12] proposed a method that used the environment at the time of intrusion and the attack information collected by an information-collection tool called Sebek [2] in the honeypot for security learning. However, freely installing the service or system operated by each institution in this study was difficult because the honeypot was separately operated in each operating system. In the case of attack detection, it encountered a drawback in that it could not be operated in an environment where unknown attacks existed or where security programs could not be installed because it relied on the Snort rules or external security programs. Another limitation of this method was that the analysis was performed only in a limited operating-system environment.

L. Christopher et al. [3] installed and operated previously developed Dionaea [7] and Kippo [13] honeypots in specific places based on the routine active theory [5], which is a criminology theory used in cybercrime research, and used the collected information to propose a training model for information security-awareness education of employees. However, the education and training model proposed in their paper suffered from the limitations in terms of responding and establishing preventive measures in case of actual intrusion incidents because their method only collected information for conducting security-awareness education. In addition, because the honeypot used in their model only targeted a limited environment, collecting and providing security-threat information suitable for the service or system operated by each institution were difficult. Many studies have been conducted on learning of various intrusion incidents using a honeypot.

However, most of these studies were conducted to focus on providing fragments of information such as collected malicious codes, IP addresses, or domain addresses instead of using intrusion incidents from the perspective of education. For this reason, existing studies suffer from the limitations in which the use of honeypot, which was confined to an already developed environment, is required, and the educational information that can be provided for learners is insufficient. Therefore, a method that maximizes the learning effect in the educational process is proposed in the present paper by enabling easy replication of the service or system operated by institutions in the honeypot environment and allowing the learners to directly analyze and respond to the environment at the time of the intrusion incident based on the collected information.

3 EVENT-RECORDABLE CONTAINER

The CERT training platform proposed in this paper operates the environment used by an institution by cloning it in the honeypot and provides the learner with challenges of the intrusion incidents that have occurred. In addition, an "event-recordable container" that enables recording of various events such as network and file system generated inside the honeypot is proposed. Fig. 1 shows a schematic diagram of the proposed CERT training platform in which intrusion-incident training can be performed in the environment used by the institution.

3.1 Honeypot

The Docker-based honeypot-management console is provided, as shown in Fig. 2, to efficiently operate the honeypot that replicates the operating environment of each institution. Institutions can either create a service or system in operation in a Dockerfile or create a honeypot using an application image in the Docker image registry [10]. The created honeypot is operated in an environment that is separate from the host server and creates a separate network interface for external communication.

3.2 Artifact collection

Once the honeypot is successfully created, elements that are needed for analysis of intrusion incidents, such as file-system events, process lists, and network packets in containers, are automatically collected when an event occurs. A "/var/log/wtmp" file is recorded if an external attacker successfully accesses the honeypot container of the organization that operates in a secure shell (SSH) environment, and all the contents are recorded in the network packet if the attacker enters a command. These artifacts, which are collected in a certain period, are stored by an artifact-collection command performed at each cycle designated by the administrator. The honeypot container that receives the collection command stops the new recording and saves all the artifacts recorded so far in a file format, as listed in Table 1.

Table 1: List of artifacts saved as files

Type	Filename
File System Dump	dump.tar
Network Dump	network.pcap
File System Events	container.diff
Processes	container.top
Container Standard Output	container.log

3.3 Artifact classification

The classification work is performed on the order shown in Fig. 3 once the artifacts from several viewpoints are accumulated. Its usability as a challenge in the classification stage is determined based on the events of the files collected during the classification work.

As a result of the data sorting of the accumulated artifacts according to the items using the aforementioned classification factors, we find that the frequency and the total number of events are higher in the case of an intrusion incident than those under another event, as shown in Fig. 4. According to this operation, unnecessary artifacts can be prevented from being registered in the training system, and administrators can check the number of security threats in the institution honeypot using a visibility graph. Therefore, only artifacts that passed the artifact-classification criteria can be registered in the training system.

From the aforementioned discussion, we selected nine application environments that are mainly used by companies or institutions, as listed in Table 2, and operated them from August 10, 2020 to September 30, 2020 on four servers.

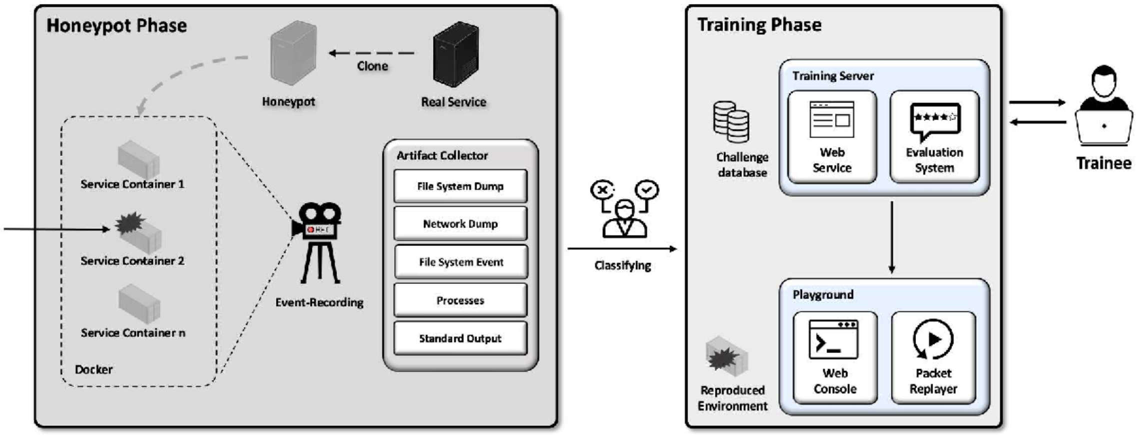
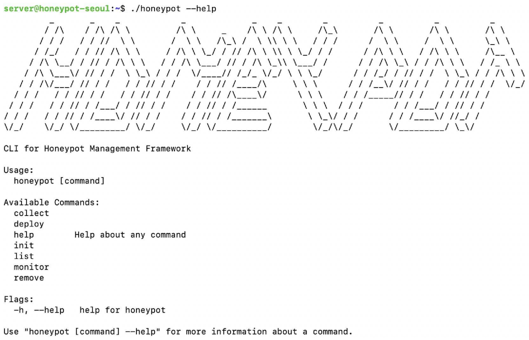


Figure 1: Overview of CERT Training Platform



```
server@honeypot-seoul:~$ ./honeypot --help

CLI for Honeypot Management Framework

Usage:
  honeypot [command]

Available Commands:
  collect
  deploy
  help        Help about any command
  init
  list
  monitor
  remove

Flags:
  -h, --help   help for honeypot

Use "honeypot [command] --help" for more information about a command.
```

Figure 2: Honeypot management console

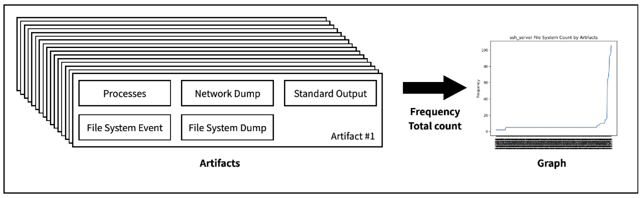


Figure 3: Artifact classification process

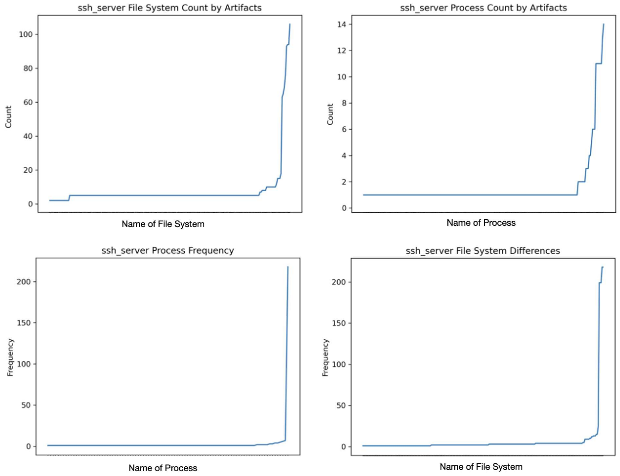


Figure 4: Graph of the frequency and total count per event (SSH server)

Table 2: Honeypot operating-environment information

Name	Location	Comments
Nginx	Seoul	Web Server
SSH	Seoul	Ubuntu 18.04 with SSH Server
Wordprcess	Tokyo	Company Website
RDP	Tokyo	Remote Desktop
MySQL	Seoul	Database
Elasticsearch	Seoul	Search Engine with mock account data
Redis	Seoul	Cache Server
CUPS	Seoul	Open Source Printer Server
Telnet	Seoul	IoT Device

From the operation of the honeypot server, 2,356 artifacts were collected in which 196 were classified as intrusion incidents and registered in the training system. Therefore, the environment operated by the institution was replicated in the honeypot, and the artifacts required for the training system were automatically collected through the classification work.

4 TRAINING SYSTEM

The main objective of the present study is to develop a training system that can improve the ability to analyze intrusion incidents and establish countermeasures. Therefore, the intrusion-incident training system developed by our research team has been implemented so that learners can learn through a Web-based browser without installing a separate program. The training system is mainly composed of a training server where the learner directly accesses and performs training on a playground server that reproduces the intrusion-incident environment, as shown in Fig. 5.

4.1 Training Server

The training server consists of learner-list management, intrusion-challenge list, and the detailed information, submitted analysis report, and scoring function by the administrator. This server provides an educational environment in the form of a Web service for learners through continuous communication between the honeypot and playground servers. The honeypot server registers information on the challenges determined to be a suspected intrusion incident by the training server and discloses it to the learners. This disclosed challenge is displayed to the learner, as shown in Fig. 6. The challenges of the intrusion incidents found in the environment are displayed when the learner selects the environment to practice, and the collected information on the challenges can be checked by selecting the desired challenges. The information provided to the learners is shown in Fig. 7.

The learner can perform not only a static analysis based on the provided intrusion incident information but also another analysis by directly accessing the environment where the intrusion incident is reproduced. When the learner requests access to the reproduced environment from the training server, the training server requests the playground server to reproduce the environment of the intrusion incident. After the reproduction is completed, the learner can access the reproduced environment through a Web browser. In addition, the learner can download the file-system dump and packet dump files, which can be used to analyze the environments when the Internet is unavailable or to analyze the files in detail if necessary.

4.2 Playground server

The playground server reproduces the intrusion-incident environment to enable the learner to access the intrusion-incident environment using a Web browser. The learner can use this process to access and analyze the reproduced intrusion-incident environment without installing a separate program, as shown in Fig. 8. In addition, because a separate intrusion-incident environment is provided for each learner, an advantage is gained in that programs required for the analysis can be freely installed and used in the reproduced environment.

4.3 Evaluation system

An evaluation method to determine whether a learner has properly performed an intrusion-incident analysis and established a countermeasure through the training system is presented in this section. Evaluation items such as an indicator of compromise (IoC), Snort rule coverage and incident report, and evaluation methods for each item are provided in this evaluation system.

4.3.1 IoC. IoC is an intrusion index for traces left by hackers after intrusion attempts. IoC is responsible for ensuring that CERT makes accurate decisions and actions. This item can be used by a learner to calculate the result in a formalized index of the analyzed intrusion incident. The OpenIOC [9] format developed by Mandiant was selected in the evaluation system. OpenIOC is a form of intrusion-incident evaluation index that displays intrusion-incident items such as IP, domain, file hash, and file name based on logical operations such as AND and OR. For the convenience of learners, a page similar to the OpenIOC Editor is provided in the training server in this evaluation system. The learner uses this editor to analyze the intrusion incident and then prepares and submits the IoC. The submitted IoC calculates the consistency of the submitted IoC based on the artifact information of the challenge in the evaluation system in a ratio form and provides the result for the learner.

4.3.2 Snort rule coverage. Snort is an open-source network-intrusion-prevention system [4] that can perform traffic analysis and log in the network. The research team evaluates whether the learners can analyze intrusion incidents and properly establish firewall policies that can block attacks based on the Snort rule coverage, which provides not only an intrusion-incident analysis but also training on establishing countermeasures at the firewall level. After the intrusion-incident analysis, the learner submits the Snort rule that can block attacks such as intrusion-incident attack. The submitted rule assesses the coverage by replaying the packet dump of the artifact in the Snort-applied environment. Therefore, the detected rule written and submitted by the learner in the actual packet can be numerically checked.

$$Coverage = \frac{Detected\ packets}{Replayed\ packets} * 100(\%) \tag{1}$$

4.3.3 Incident report. The intrusion-incident analysis report is a report that describes the countermeasure and analysis of an intrusion incident that occurred. The completion of the learner in learning the intrusion incident that occurred under a relevant challenge is finally evaluated in this report. The report form provided by the evaluation system for learners can be encoded in the training server using some of the items in the intrusion-incident analysis report proposed by the National Institute of Standards and Technology [11]. The learner can develop the ability to write the analysis result of the intrusion incident in the form of a report by creating an intrusion-incident analysis report, and the evaluator can finally check the report to identify whether the learner has correctly learned the intrusion incident using the evaluation system.

5 CONCLUSION

This paper has proposed a method of providing a training system for intrusion incidents by operating honeypots specialized for each institution. In the existing intrusion-incident training system, the information and analysis environment in which learners can access intrusion incidents are limited. To minimize these limitations, the training system proposed in this paper duplicates the service operated by the institution, operates it in a honeypot, collects intrusion-incident information, and provides an environment for learners through the training system. Through this process, a training system in which a customized intrusion-incident challenge is introduced can be operated for each institution. We expect that the learning effect on the learners will be maximized in the process of analyzing actual attacks.

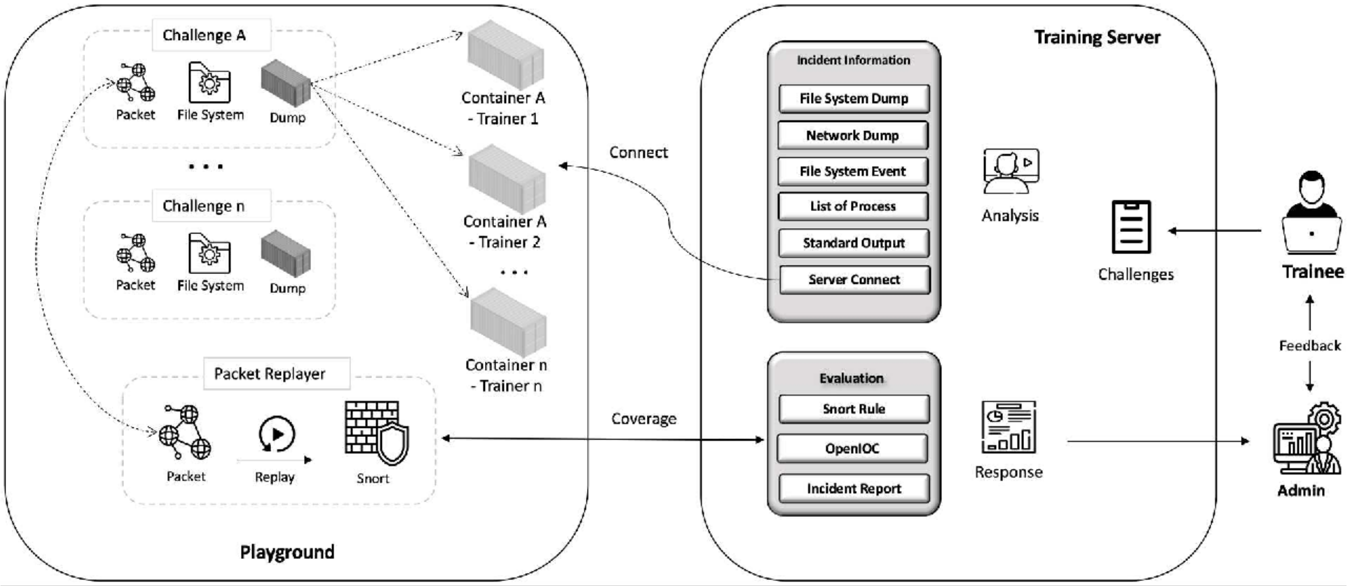
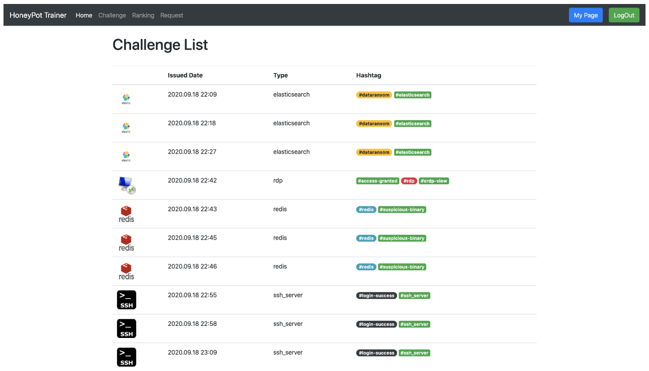


Figure 5: Overview of training system



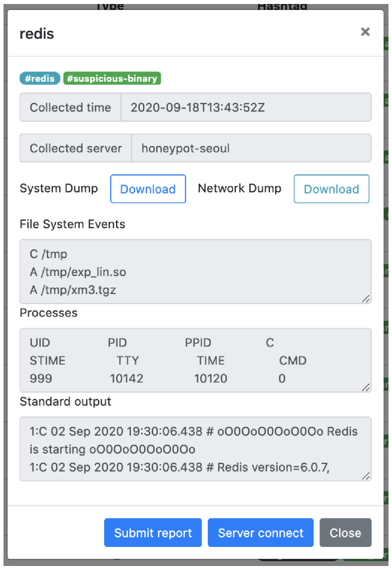
Challenge List

	Issued Date	Type	Hashtag
elastic	2020.09.18 22:09	elasticsearch	#dataransom #elasticsearch
elastic	2020.09.18 22:18	elasticsearch	#dataransom #elasticsearch
elastic	2020.09.18 22:27	elasticsearch	#dataransom #elasticsearch
	2020.09.18 22:42	rdp	#access-granted #rdp #xrdp-view
redis	2020.09.18 22:43	redis	#redis #suspicious-binary
redis	2020.09.18 22:45	redis	#redis #suspicious-binary
redis	2020.09.18 22:46	redis	#redis #suspicious-binary
>_ SSH	2020.09.18 22:55	ssh_server	#login-success #ssh_server
>_ SSH	2020.09.18 22:58	ssh_server	#login-success #ssh_server
>_ SSH	2020.09.18 23:09	ssh_server	#login-success #ssh_server

Figure 6: Training server

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) (NRF-2020R1A2C4002737) and the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00420).



#redis #suspicious-binary

Collected time	2020-09-18T13:43:52Z
Collected server	honeypot-seoul

System Dump Download Network Dump Download

File System Events

```
C /tmp
A /tmp/exp_lin.so
A /tmp/xm3.tgz
```

Processes

```
UID         PID         PPID        C
STIME         TTY         TIME         CMD
999         10142        10120        0
```

Standard output

```
1:C 02 Sep 2020 19:30:06.438 # oO0OoO0OoO0Oo Redis
is starting oO0OoO0OoO0Oo
1:C 02 Sep 2020 19:30:06.438 # Redis version=6.0.7,
```

Figure 7: Detailed information on the intrusion incident

REFERENCES

[1] Frank Adelmann and Tamas Gaidosch. 2020. *Cybersecurity of Remote Work During the Pandemic*. IMF COVID-19 Special Series. International Monetary Fund, Washington, D.C.

[2] Michael Davis Camilo Viecco and Sebek Droids. 2006. Sebek. http://honeynet.onofri.org/tools/sebek/. (2006).

[3] Lek Christopher, K-KR Choo, and Ali Dehghantanha. 2017. Honeypots for employee information security awareness and education training: a conceptual EASY training model. In *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*. Elsevier, 111–129.

[4] Cisco. 2020. Snort. https://www.snort.org/. (2020).

[5] Sarah Cohen, Werner Nutt, and Yehoshua Sagic. 2007. Deciding equivalances among conjunctive aggregate queries. *J. ACM* 54, 2, Article 5 (April 2007), 50 pages. https://doi.org/10.1145/1219092.1219093

[6] Thibault Debatty and Wim Mees. 2019. Building a Cyber Range for training CyberDefense Situation Awareness. In *2019 International Conference on Military Communications and Information Systems (ICMCIS)*. IEEE, 1–6.

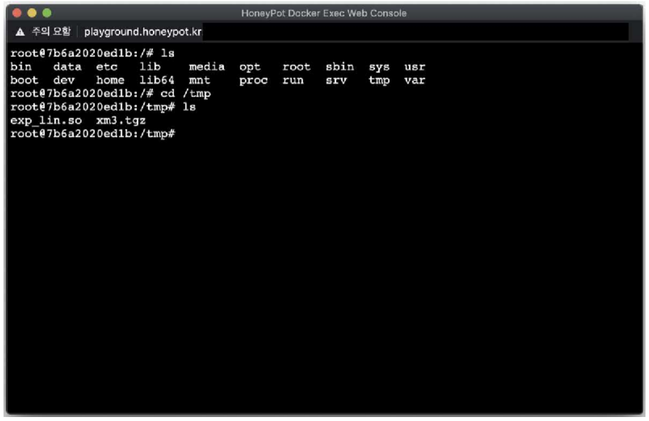
[7] DinoTools. 2013. Dionaea - catches bugs. https://github.com/DinoTools/dionaea. (2013).

[8] P Fanfara, M Dufala, and E Chovancová. 2013. Usage of proposed autonomous hybrid honeypot for distributed heterogeneous computer systems in education process. In *2013 IEEE 11th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 83–88.

[9] FireEye. 2012. IOC Editor. https://www.fireeye.com/services/freeware/ioc-editor.html. (2012).

[10] Docker Inc. 2020. Docker Hub. https://hub.docker.com. (2020).

[11] National Institute of Standards and Technology 2012. *Computer Security Incident Handling Guide*. National Institute of Standards and Technology.



```
root@7b6a2020ed1b:/# ls
bin   data  etc   lib    media  opt   root  sbin  sys  usr
boot  dev   home  lib64  mnt    proc  run   srv   tmp  var
root@7b6a2020ed1b:/# cd /tmp
root@7b6a2020ed1b:/tmp# ls
exp_lin.so  xm3.tgz
root@7b6a2020ed1b:/tmp#
```

Figure 8: Reproduction of the intrusion-incident environment

[12] Gordon W Romney, Jeremiah K Jones, Brandon L Rogers, and Philip MacCabe. 2005. IT security education is enhanced by analyzing Honeynet data. In *2005 6th International Conference on Information Technology Based Higher Education and Training*. IEEE, F3D/10–F3D/14.

[13] Upi Tamminen. 2009. Kippo. https://github.com/desaster/kippo. (2009).