Scientific Journal of Informatics
Vol. 7, No. 1, May 2020

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Web Forensic on Container Services Using GRR Rapid Response Framework

Imam Riadi[1], Rusydi Umar[2], Andi Sugandi[3]

[1]Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[2,3]Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
Email: [1]imam.riadi@is.uad.ac.id, [2]rusydi.umar@tif.uad.ac.id, [3]andi1707048011@webmail.uad.ac.id

Abstract

Cybercrime on the Internet that keeps increasing does not only takes place in the environment that is running web applications traditionally under operating system, but also applications that are deployed in a more advanced environment like container service. Docker is a currently popular container service in Linux operating system needs to be secured and implements incident response mechanism that will investigate web server that was attacked by DDoS in fast, valid, and comprehensive way. This paper discusses the investigation using GRR Rapid Response framework on a web server that is running inside container service on Linux operating system. This web server then is attacked by DDoS, and the attacker running on Windows operating system. This research has successfully investigated digital evidence in the form of a log file of web servers running on container service and digital evidence through netstat on Windows computer.

Keywords: *Forensics, Network, Docker Container, GRR Rapid Response, Web*

1. INTRODUCTION

This paper is motivated by the increasing popularity of the web applications deployment on container services [1] as cloud computing arises since 2006. Currently, Docker is leading container services since 2017, that the deployments have increased by 75% until June 2018 [2]. Docker has successfully deployed and maintained the container service inside Linux operating system kernel. It isolates resources and programs to separate individual programs and libraries in the boxes, with many features included.

While Docker is growing popular by now, the cybercrime of web applications that are running inside container services cannot avoid from the coordinated attacks over the Internet, not so different those running by Docker [3]. Attacks on the Internet include Teardrop, TCP SYN flood, smurf, IP spoofing, session hijacking, UDP Flood, Flood ping, and DDoS [4].

DDoS attack causes the webserver cannot be accessed by legitimate user. This is caused by cyber-attacks that interfere with network or operating system services of the host, like web server [5], resulting in computer resources to be temporarily or indefinitely unavailable. Investigating dynamic data like data packets transmitted over Internet, or collecting data traffic on network interface devices to identify and analyze DDoS attack on web server that is running inside container services, needs special methods and tools to perform digital forensics. Like they need special and appropriate procedures involved in investigating on a mobile device [6]. There is also a static

forensics [7] to investigate digital evidence on static or persistent data (SSD or flash memory), while dynamic data (DRAM, running programs/processes, log files [8], network status of a network interface) require live forensics [9] or even network forensics [10], because data that will be investigated is not persistent.

Those DDoS attack numbers need appropriate response and tools to investigate by practitioner or incident response team when it happens. GRR Rapid Response (GRR) framework is comprehensive tool to provide practitioner a complete and reliable incident response investigation and analysis of Internet attack (DDoS), in live and remote mechanisms.

There are two important sides of GRR framework: client and server-side. GRR clients (an agent program that is running on targeted/attacked computer) is installed and deployed on a victim computer, later this computer will be investigated and analysis by activating polling of GRR Frontend Server to works, then ask GRR Server what tasks should be done afterward The tasks are such searching web server log files, and downloading them, or listing files or directory. In the other side, there are three main infrastructures of GRR server [11]: GRR Frontend, GRR Workers, and GRR AdminUI, Flow, and Hunt.

Messages is the concept used to communicate between GRR client and GRR server. By using HTTP protocol, GRR server will send messages as a (batched) "Requests", consisting of tasks (of GRR Flows) that want to investigate in client computers. Then GRR clients send responses as (batched) "Responses" messages, the resulting data after successfully investigating processes on client side.

GRR framework was chosen for the reason of the excellent features provided such as on quick response when investigating digital evidence in the form of log files of web server that is running inside container services, or even on investigating network status on computer attacker (both computers are running as a GRR clients).

All of the forensic activities involved in this research begins by installing GRR server side on Linux server. GRR server then produce three types GRR client programs afterward (for Linux, Windows, and Mac OS version). Then, GRR client program for Linux type (DEB or RPM) is installed on a Linux client that runs a web application on web server inside container service. This web application and its TCP port are exposed to the public network. Another GRR client program (for Windows version) then installed on Windows computer acts as both another GRR client and an attacker running DDoS script.

Linux client will be attacked by Windows computer by sending SYNC Flood on port 8888 of the webserver. After detected an attack, the practitioner then activates GRR Server to send GRR Tasks through a Flow [12] to both GRR clients to start investigating the evidence by searching for web server log files on Linux client file system, and network status on Windows computer (using netstat tool) to obtain the original source of the attacker and the timestamps of the accident. The resulted investigations on computer clients are then sent to GRR server to analyze and review.

2. METHODS

2.1. Literature Review

Today's researches related to this topic of investigating digital evidence by using GRR Rapid Response framework are the following.

The research in [12] discussed the triage of investigating digital evidence at enterprise environment using GRR framework. In [13] proposed a scalable storage on GRR framework. The research in [14] is discussing network forensics on gaining digital evidence by using GRR framework in the healthcare setting and organizations in general.

2.2. Network Architecture

This research using network architecture consisting of a single GRR server, A GRR client program on Windows computer (as attacker), and another GRR client program on Ubuntu Linux (as victim), running web application inside Docker container services as seen in Figure 1.

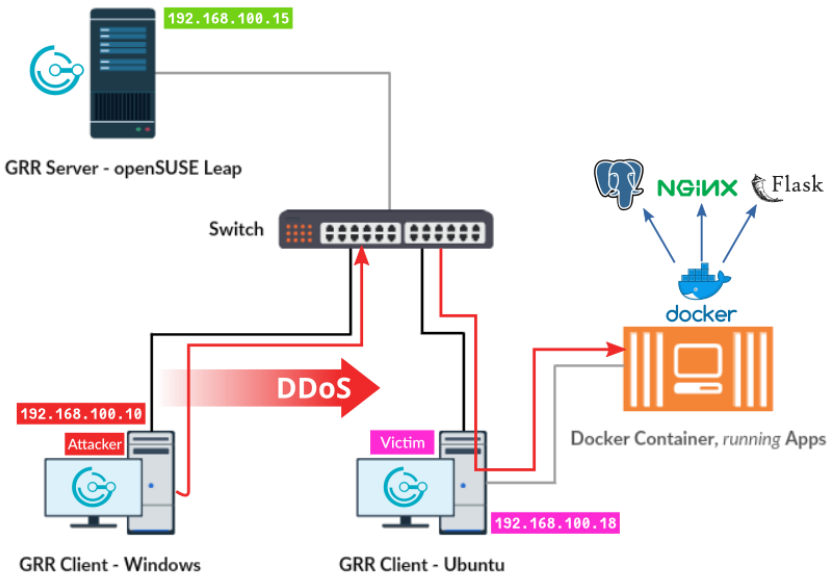

Figure 1. Network architecture overview of GRR server and GRR clients

The National Institute of Standards and Technology (NIST) [15] is used as a forensics method in this research, consisting of forensic stages like acquisition, examination, utilization, and review, as illustrated in Figure 2.

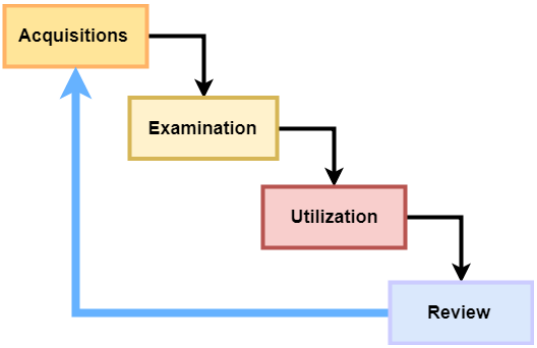

Figure 2. NIST Method

NIST is one of the agencies of the United States Department of Commerce that responsible for promoting innovation and industrial competitiveness by leading the development of technical standards for reliable, robust, trustworthy, secure, portable, and interoperable digital forensics competence [16].

2.3. Acquisitions

The research process begins with identifying data sources. Data acquisition steps consist of the activity of identifying and collecting data. Table 1 is a table of tools and devices used, and Table 2 are three accompanying computers (openSUSE, Ubuntu, and Windows) used in this research.

Table 1. Hardware and software used to investigate digital evidence using GRR framework

No	Hardware / Software	Description
1	Computer running GRR server	Intel i7 CPU, 32GB RAM, HDD 250GB
2	Computer running GRR client (Linux)	Intel i7 CPU, 4GB RAM, HDD 64GB
3	Computer running GRR client (Windows)	Intel i7 CPU, 4GB RAM, HDD 64GB
4	GRR framework program (software)	Version 3.23.2
5	Operating system server	openSUSE Leap 15.0
6	Operating system client #01	Ubuntu 18.04.02 (LTS)
7	Operating system client #02	Windows 10
8	Hammer DDoS program (software)	A Python3 script to launch a DDoS attack
9	Switch (network device)	CISCO Catalyst 2960 Plus

Table 2. The IP address on each host involved

No	Host	IP Address
1	openSUSE Leap 15.0	192.168.100.115/24
2	GRR Client #01: Ubuntu Linux 18.04.02 (LTS)	192.168.100.15/24
3	GRR Client #02: Windows 10	192.168.100.10/24

2.4. Examination

After the desired data has been collected, the following step is to examine the data, the processes of identifying, collecting, and organizing the relevant information from the acquired data.

2.5. Utilization
Data utilization of NIST describes the process required to prepare and present information that came from the examination step. The GRR framework point of view related to this utilization process is provided by the implementation [11] of GRR Flows.

2.6. Review
Digital forensics practitioners will continuously review their investigation processes and practices within the context of current tasks, in the purpose of helping to identify policy shortcomings, procedural errors, and other issues that may need to be evaluated and remedied.

3. RESULT AND DISCUSSION
Regarding the results and analysis processes of the research that has been done, there are criteria of the analyzed parameters used to clarify what the expected results have been collected, as seen in Table 3.

Table 3. List of parameters for the analysis process

No	Parameters	Result	
		YES	NO
1	Digital evidence (log files) can be obtained?	√	
2	The origin identity of the attacker (IP address) can be obtained?	√	
3	Digital evidence (log files) can be trusted?	√	

Based on NIST method described previously, digital forensics activities in this research area using the following steps to gain digital evidence (log files) produced by a web server that is running inside container services.

3.1. Acquisition
There are preparations that have to be done first before starting acquisition processes, based on the scenario illustrated previously in Figure 1. These preparations are:
1) Make sure that all main components of GRR server (Worker, AdminUI, and FrontEnd) are already running on the server computer.
2) Make sure that all GRR client programs are already running on each particular computer.
3) Run a web application [17] inside container services by using Docker Compose [18].
4) Begin to run DDoS script on Windows computer (attacker), give the parameters: destination IP address of victim computer (192.168.100.18) and the port number (8888).
5) Last, run acquisition processes on GRR Server WebUI.

The acquisition step in this paper is to run the GRR framework in each particular host. We can make sure that all GRR clients are already running by accessing the front page of the GRR Server WebUI as we see in Figure 3.

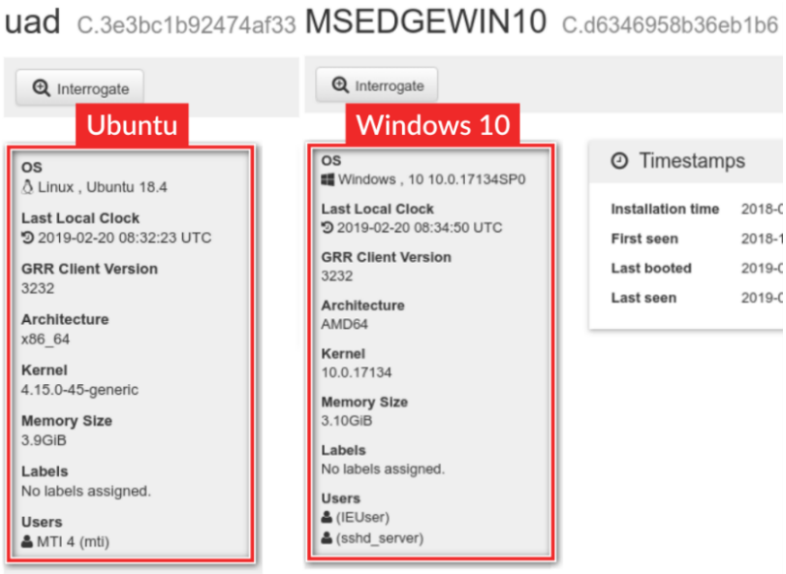

Figure 3. Host information of Ubuntu Linux and Windows 10

3.1.1. Acquisition of Docker Container (Victim)

At the acquisition step in Docker container services, we must create a custom ArtifactCollectorFlow: dockerlogs.yml [19], because GRR framework does not have the feature to collect log files in the default installation.

After we finishing the process of the acquisition on the victim computer (Ubuntu), the next step is to collect other digital evidence from the view of the attacker (Windows 10).

3.1.2. Acquisition on Windows (Attacker)

We have to prove that the attacker was coming from this Windows 10 computer. Netstat of GRR Flow Artifact has the ability to collect valuable network information of the interface card on the particular computer.

3.2. Examination

In this examine step, we begin the examination process on all GRR clients

3.2.1. Examination on Docker Container (Victim)

The examination process on the Ubuntu Linux side begins after the Result Message returns from the GRR client and arrives in GRR Server. Choose which data is related to DDoS that attacks web server inside Docker container services.

3.2.2. Examination on Windows (Attacker)

In the GRR WebUI interface (Manage launched Flows menu), we can examine network information on a computer that runs DDoS script. This is a response message from the GRR client that received by GRR Server. After this, we begin to utilize the examined data in the following steps.

3.3. Utilization

The utilization step from the view of GRR Server is easy with the help of GRR WebUI interface. This really useful for practitioners to get appropriate information from all GRR clients: Ubuntu Linux (run Docker container services), and Windows 10 (run DDoS script).

3.3.1. Utilization on Windows (Attacker)

We have to elaborate data examined from the next step and utilize it with the information shown in Figure 4, to make sure that the original identity of the attacker was coming from Windows 10 (192.168.100.10).

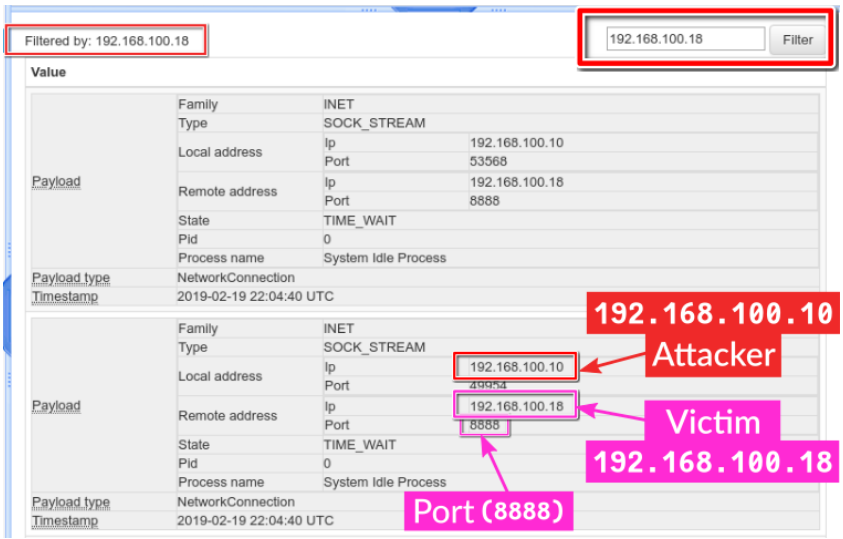

Figure 4. Utilization result of Netstat Response Message

In this step, we successfully find the source of the attacker. In Figure 4, it tells that the original identity of the attacker was coming from IP address 192.168.100.10, as we expected.

3.3.2. Utilization of Docker Container (Victim)

Docker Logs [20] has the ability to save system log (output and error log) inside a log file on the host file system. This feature will make the digital forensics practitioner's life easy and saves time in implementing the utilization process in this research.

In Figure 5, the utilization step is implemented by the GRR AdminUI interface, so we can successfully collect and display the digital evidence inside log files that were coming from Docker Logs on GRR client (Ubuntu Linux).

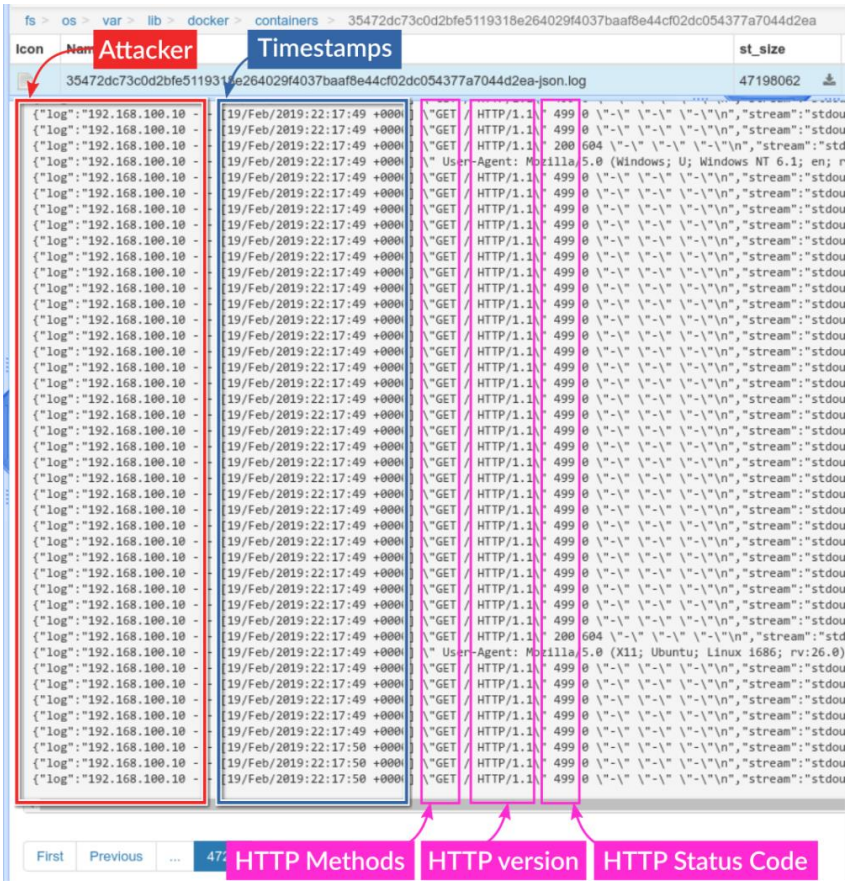

Figure 5. Utilization result of LinuxDockerFiles Response Message

3.4. Review

According to the investigations, IP addresses (source and destination), port numbers, and timestamps are successfully determined by GRR framework to gain evidence both from investigating web server log files in computer victim and netstat in computer attackers, as we can see in Table 4 and Table 5.

Table 4. Reviewing the investigation result of web server log files

No	Parameters	Result	
		YES	NO
1	Was the log file(s) retrieved successfully?	√	
2	Were source and destination IP Addresses identified?	√	
3	Could [log files be trusted (by using sha1sum/sha256/md5sum)?	√	

Table 5. Reviewing the investigation result of netstat

No	Parameters	Result	
		YES	NO
1	Was the netstat information retrieved successfully?	√	
2	Were source and destination IP Addresses identified?	√	
3	Could [the netstat information be trusted (by using sha1sum/sha256/md5sum)?		√

4. CONCLUSION

According to the research in this paper, GRR framework successfully achieved the investigation to determine the digital evidence in log files (in the form of IP addresses, port numbers, and timestamps) from the webserver running inside Docker container.

5. REFERENCES

[1] Liu, D., & Zhao, L. (2014, December). The research and implementation of a cloud computing platform based on docker. In *2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (pp. 475-478). IEEE.

[2] Datadog. (2018, June 13). Eight surprising facts about real docker adoption. Retrieved from https://www.datadoghq.com/docker-adoption

[3] Combe, T., Martin, A., & Di Pietro, R. (2016). To docker or not to docker: a security perspective. *IEEE Cloud Computing*, *3*(5), 54-62.

[4] Jingna, L. (2012, July). An analysis on DoS attack and defense technology. In *2012 7th International Conference on Computer Science & Education (ICCSE)* (pp. 1102-1105). IEEE.

[5] Mualfah, D., & Riadi, I. (2017). Network forensics for detecting flooding attack on web server. *International Journal of Computer Science and Information Security*, *15*(2), 326.

[6] Umar, R., Riadi, I., & Zamroni, G. M. (2018). Mobile forensic tools evaluation for digital crime investigation. *International Journal on Advanced Science, Engineering and Information Technology*, *8*(3), 949-955.

[7] Albanna, F., & Riadi, I. (2017). forensic analysis of frozen hard drive using static forensics method. *International Journal of Computer Science and Information Security (IJCSIS)*, *15*(1).

[8] Riadi, I., Istiyanto, J. E., & Ashari, A. (2013). Log analysis techniques using clustering in network forensics. arXiv preprint arXiv:1307.0072

[9] Zulkifli, M. A., & Dahlan, U. A. (2018). Live forensics method for analysis denial of service (DOS) attack on routerboard. *Int. J. Comput. Appl*, *180*(35), 23-30.

[10] Yudhana, A., Riadi, I., & Ridho, F. (2018). DDoS classification using neural network and naïve bayes methods for network forensics. *International Journal of Advanced Computer Science And Applications*, *9*(11), 177-183.

[11] Cohen, M. I., Bilby, D., & Caronni, G. (2011). Distributed forensics and incident response in the enterprise. *Digital Investigation*, *8*, S101-S110.

[12] Moser, A., & Cohen, M. I. (2013). Hunting in the enterprise: forensic triage and incident response. *Digital Investigation*, *10*(2), 89-98.

[13] Cruz, F., Moser, A., & Cohen, M. (2015). A scalable file based data store for forensic analysis. *Digital Investigation*, *12*, S90-S101.
[14] Acharya, S., Glenn, W., & Carr, M. (2015, November). A GRReat framework for incident response in healthcare. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 776-778). IEEE.
[15] NIST-a. (2012). Information testing laboratory, "computer forensics tool testing program". *URL www.cftt.nist.gov*.
[16] NIST-b. (2012). Guide to integrating forensic techniques into incident response. *URL http://csrc.nist.gov/publications/nist-pubs/800-86/SP800-86.pdf*.
[17] Juggernaut. (2017, September 28). A working example of nginx flask postgres multi-container setup using Docker compose. Retrieved from *https://github.com/juggernaut/nginx-flask-postgres-docker-compose-example/tree/auto-reload-nginx-with-python*
[18] Docker Inc. (2017, September 28). Docker compose. Retrieved from *https://docs.docker.com/compose*.
[19] Sunardi , Imam Riadi and Andi Sugandi. (2019). Forensic analysis of docker swarm cluster using grr rapid response framework. *International Journal of Advanced Computer Science and Applications (IJACSA*), *10*(2).
[20] Docker Inc. (2017, September 28). View logs for a container or service. Retrieved from *https://docs.docker.com/config/containers/logging.*