Forensic Analysis of Docker Swarm Cluster using Grr Rapid Response Framework

Sunardi[1]
Department of Electrical Engineering
Universitas Ahmad Dahlan
Yogyakarta, Indonesia

Imam Riadi[2]
Department of Information System
Universitas Ahmad Dahlan
Yogyakarta, Indonesia

Andi Sugandi[3]
Master Program of Informatics
Universitas Ahmad Dahlan
Yogyakarta, Indonesia

***Abstract*—An attack on Internet network does not only happened in the web applications that are running natively by a web server under operating system, but also web applications that are running inside container. The currently popular container machines such as Docker is not always secure from Internet attacks which result in disabling servers that are attacked using DoS/DDoS. Therefore, to improve server performance running this web application and provides the application log, DevOps engineer builds advance method by transforming the system into a cluster computers. Currently this method can be easily implemented using Docker Swarm. This research has successfully investigated digital evidence on the log file of containerized web application running on cluster system built by Docker Swarm. This investigation was carried out by using the Grr Rapid Response (GRR) framework.**

Keywords*—Forensics; Network; Docker Swarm; Grr Rapid Response

I. Introduction

This research is motivated by the popularity of cloud computing where web applications are run in it by container machine [1]. Currently, Docker is one of the container machines implemented by almost 25% of the world's Internet companies [2]. Fig. 1 shows a significant rate of Docker utilization in Internet companies until the beginning of 2018.

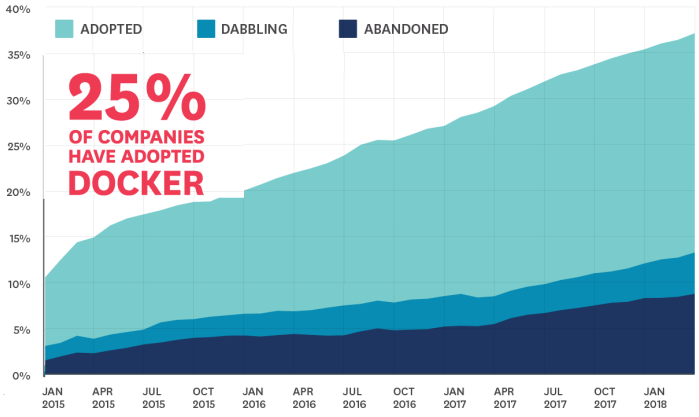

Fig. 1. Nearly One Quarter of Companies Have Adopted Docker

Docker has successfully implemented the container concept. It is isolating resources and programs to separate boxes with many features included.

Their other concepts of isolation are similar to Docker such as Virtual Machines (VMs), BSD jails, and Solaris containers, which can also isolate the resources of a host. However, since their designs differ, they are fundamentally distinct. The implementation of a VM is for virtualizing the hardware layer with a hypervisor. If an application is running on a VM, it needs to install a full operating system first [3]. In other words, the resources are isolated between guest operating systems on the same hypervisor.

The isolation relationship of container and VMs is illustrated in Fig. 2. Container isolates an application at the OS-layer (VM2), while VM-based separation is achieved by the operating system (VM1).

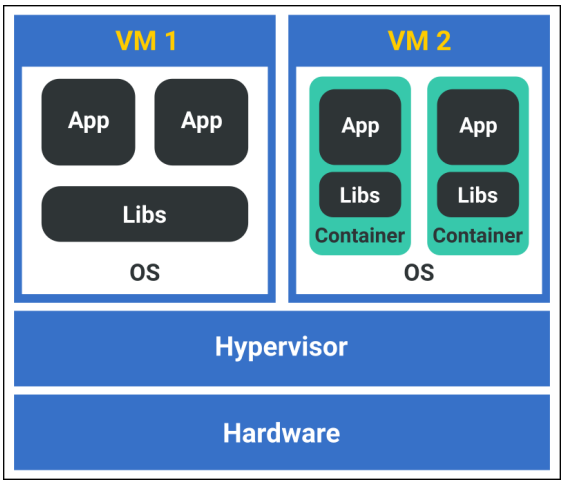

Fig. 2. Comparing various application runtime models.

In addition to became a Docker Swarm cluster, existing containerized application must deployed and managed by using Docker Swarm [4] and declare one machine (node) as a Swarm Manager and other node as worker. Service that will be provided by web application must define a number of instances we want to create, on what port service will be exposed to the outer world, storage resources available etc. Based on configuration defined, Docker tries to maintain that desired state in a sense that suppose if a worker node becomes unreachable [5], Docker schedules the tasks running on that node to other reachable nodes.

Even though Docker is increasingly popular, the security of web applications running in this container environment cannot avoid from the massive attacks on Internet networks, including those run by Docker [6]. Attacks on the Internet include SYN

Flood, IP Spoofing, DoS, UDP Flood, Flood ping, Teardrop, Land, Smurf, and Fraggle [7]. DoS attack causes user of web applications unable to access the server, which are caused by computer network attacks that interferes the operating system on the server, resulting loses of a lot of computer resources [8].

Digital forensics is the use of scientific methods used to prove a case with the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence originating from digital sources, including data packets transmitted over computer networks, for the purpose of facilitating or continuing the reconstruction of events in criminal acts or as part of a criminal investigation [9], [10], [11], or help to prevent unauthorized actions from interfering with planned operations.

Dealing with dynamic data like data packet in TCP/IP networks needs different approach from static data like text, image, or multimedia documents. It needs special tools and conditions to meet the requirements to investigate on such data, even there are more strictly procedures involved in investigating on mobile device [12]. Static or persistent data will require static forensics [13], while dynamic data (computer RAM, running processes, log file, registry status, network status of network device) require live forensic, because data is not persistent, and will change periodically or even unconditionally [14]. Live forensics that investigates on network computers is called network forensics [15]. This situation brings the network forensics to the crowd as part of digital forensics. Network forensics is the science that deals with capturing, recording, and analysis of network traffic for detecting intrusions and investigating them [16].

Research in network forensics focuses on traffic captures, log files, and other artifacts related to a network incident, including analysis of network events in order to discover the source of security attacks [17]. Network forensics analyzes data traffic on network connections and interface statistics in network device such ethernet adapter on web server. The goal is to achieve the traceback to the source of the attack so that the origin identity of the attacker can be obtained.

The need for getting network forensics up and growing is relevant as DDoS attack in Internet has increased rapidly. As shown in Fig. 3, compared to third quarter of 2017, the number of DDoS attacks slightly increased due to September 2018, while in the summer and throughout the year, there was a noticeable drop in the number of DDoS attacks.

The graph in Fig 3 shows that the slight increase from last year is owed to September 2018, which accounts for the lion's share of all attacks (about 5 times more compared to 2017) [18]. This is a huge problem on network forensic and very challenging to encourage practitioners to give a hand and provide fast and proper solutions in form of framework to facilitate the investigation of information about attacker, when it happened, and what resource has been taken or accessed. Grr rapid response is an appropriate option to help practitioner providing a complete and fast incident response investigation and analysis of internet attack, such DoS or DDoS, remotely.

Grr rapid response framework has two working parts: client and server. GRR clients (as an agent running on computer) is deployed on computer victim that might want to investigate and analysis by polling GRR Frontend Server for works, asking

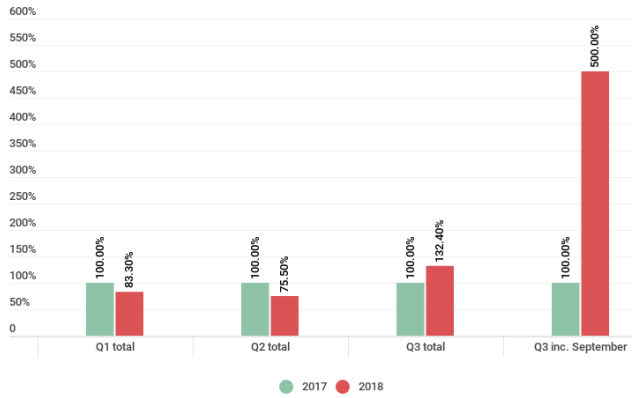

Fig. 3. Quarterly number of DDoS attacks in 2017–2018

server what task should be done next, either finding log files, downloading them, or listing the directory. While GRR server consists of three main infrastructures [19]: Frontend, Workers, and AdminUI, and other components like: data storage, a web-based graphical user interface and an API endpoint so practitioners can analysis the schedule actions on clients and view and process data.

The mechanism of client-server communication occurs between them are using concept of Messages. GRR server send messages as a (batched) "Requests" using HTTP protocol, the messages consisting of tasks of FLows that might want to investigate on client computers. GRR clients send messages as a (batched) "Responses", resulting data from what have been done on clients, succeed or not, then send the results to GRR Server through HTTP POST requests, as shown in Fig. 4, it gives an simple overview of how Messages between GRR server and clients.

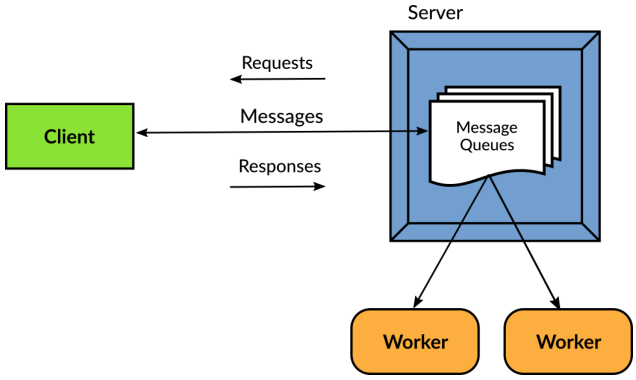

Fig. 4. A simple overview between Grr server and clients architecture

Choosing the Grr rapid response framework used in this research is for the reasons of the reliability, scalability and proven used in the enterprise machines and environment [19], simplicity of usage, and the promise of the continuation of its development in the future, because of the open/free software license choosed by developers (Apache License Version 2.0) [20] so no one will worry about the obstacle and dead-end development of this excellent forensic technology.

The overall of forensic investigation in this research begins from the installation of GRR server on Linux server. A Linux

client then acts as GRR client, running a scalable containerized web application running on cluster system built by Docker Swarm [20], exposed to the public network through a web server. A Windows box acts both as an another GRR client and an attacker running DDoS script, attacks web application by sending SYNC Flood on port 80 of the web server. After Linux client detected an attack, GRR Server sends tasks through a Flow [21] to both GRR clients to start investigating the evidence by searching for information looking for web application log files on Linux client file system and additional information by using netstat tool on Windows box, to inspect the source of attacker and the timestamps. The GRR clients then send the results to GRR server to analysis and review.

The results sent by client are received by GRR Frontend on server, then forwarding them to GRR worker to save the results into data base and before displaying them through GRR WebUI. After displaying the resulting investigation, not only GRR WebUI displaying them on client web browser with complete reports, logs, and a comprehensive views (HexView and TextView), and option to download the results, but also waiting admin user to give another action or Flows through GRR WebUI [21], to run other investigation processes on GRR clients.

II. Literature Review

Today's research related to this study is divided into two parts: the study of forensics in the network security and research on Grr Rapid Response framework.

A. Forensics in Network Security

A today's technique used in digital forensics is showing the methods and tools used for digital forensics with more complex and needs more comprehensive collaboration between. Although many systems are moving into the cloud, little research has been performed on the tools, processes, and methodologies necessary to obtain legally defensible forensic evidence in that domain. Five Most investigations require evidence retrieval from physical locations, so cloud network forensic must be able to physically locate data with, for example, a given timestamp and trace network forensic data at a given time period, taking into account the authority at different locations.

Although the live and dead forensics categories still exist, cloud models present new challenges because network data is often difficult to locate, thus acquisition might be challenging or even impossible. Analysis without acquiring network data is not possible, so network forensic tools must evolve yet again, forming an amalgam of current live and dead collection and analysis methods, as well as incorporating the intelligence to find and predict artifacts based on forensic heuristics [22].

Forensic refers to the use of evidence after the attack to determine how the attack was carried out and what the attacker did. Data traffic on the network is very complicated to be monitored. Role of network forensics is to detect abnormal traffic and identify intruders.

Tools to assist with network forensics come in a variety of forms: some are merely packet sniffers, whereas others might focus on fingerprinting, mapping, location identification, email traffic, URLs, traceback services, and honeypots. Table I summarizes some of the tools more commonly used to support network forensic investigations, along with their properties [22].

TABLE I. Tools commonly used to support a variety of network forensics investigations

Tool	Website	Attributes
TCPDump, Windump	www.tcpdump.org; www.backtrack-linux.org/backtrack-5-release	F
Ngrep	ngrep.sourceforge.net	F
Wireshark	www.wireshark.org	F
Driftnet	linux.softpedia.com /progDownload/Driftnet-Download-15905.html	F
NetworkMiner	www.netresec.com/?page=NetworkMiner	F
Airmon-ng, Airodump-ng, Aireplay-ng, Aircrack-ng	www.backtrack-linux.org/backtrack-5-release	F, L, R, C
Kismet	www.kismetwireless.net	F
NetStumbler	www.netstumbler.com	F
Xplico	packetstormsecurity.org/search/?q=Xplico	F
DeepNines	www.deepnines.com	F
Argus	www.qosient.com/argus	F, L
Fenris	lcamtuf.coredump.cx/fenris/whatis.shtml	F
Flow-Tools	www.splintered.net/sw/flow-tools	F, L
EtherApe	etherape.sourceforge.net	F
Honeyd	www.citi.umich.edu/u/provos/honeyd	F
Snort	www.snort.org	F
Omnipeek, Etherpeek	www.wildpackets.com	F, L, R
Savant	www.intrusion.com	F, R
Forensic and Log Analysis GUI	sourceforge.net/projects/pyflag	L
Dragon IDS	www.enterasys.com; www.intrusion-detection-system-group.co.uk /dragon.htm	F, R, L, C

F filter and collect;
L log analysis;
R reassembly of data stream;
C correlation of data;
A application-layer view.

B. Grr Rapid Response Framework

The research in [21] discussed the usage, analyst and benefits of the investigating computer system using Grr Rapid Response framework at a company on a large scale at triaging environment.

The research in [23] discussed about storage usage in digital forensics using Grr rapid response framework. Authors were proposing a new distributed data store that partitions data into database files that can be accessed independently so that distributed forensic analysis can be done in a scalable fashion. The authors also showed how to use the NSRL software reference database in our scalable data store to avoid wasting resources when collecting harmless files from enterprise machines.

The research in [24] discussing network forensics on seeking to examine the use of Google Rapid Response (GRR) in the healthcare setting and the general necessity for a more in-depth approach to malware incident response in healthcare organizations in general. GRR is examined for its uses in the detection of malware, along with its meeting of HIPAA requirements such as privacy and the detection and notification of breaches (security being handled through the detection of this malware). It was determined that GRR has some great potential within this field, albeit it has some flaws and limitations that should be accounted for before implementing it within a healthcare organization.

The research in [25] discussed about using Grr Rapid Response on hunting threat activities on computer networks before an accident happen. The experiment is carried out by exploiting the client's remote code by configuring the rear door of the victim system. Research shows that the achievement of research is monitored by normal behavior patterns by identifying the threat of hunting. Grr Rapid Response is able to collect the necessary forensic data from the client data obtained by displaying time to facilitate information retrieval.

C. Network Architecture

Network architecture used in this research consists of a single GRR server, A Grr client on Windows box act as attacker, and another GRR client on Ubuntu Linux, as a hypervisor of scalable containerized web application running on Docker Swarm cluster. The network architecture can be seen in Fig. 5.

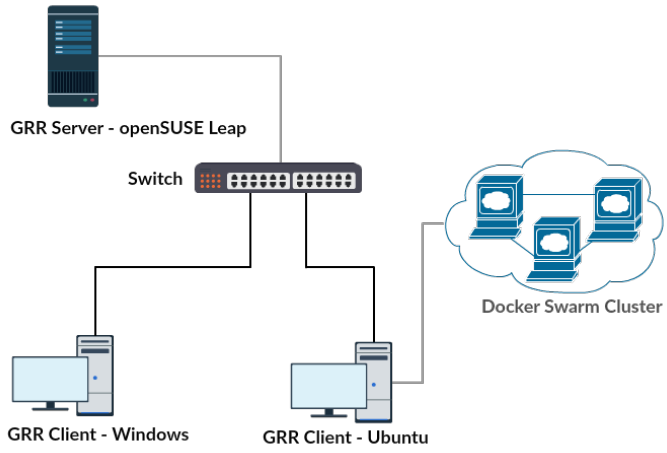

Fig. 5. Overview of Grr Rapid Response network architecture

Fig. 5 is a network architecture that will be used to simulate activity to get digital evidence using Grr rapid response on attacked host.

D. Methodology

The method used in this study is forensic methods based on the National Institute of Standards and Technology (NIST). With the forensic stages of acquisition, inspection, utilization, and review, as described in Fig. 6 [25].

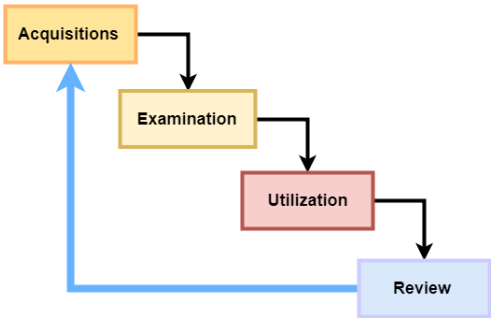

Fig. 6. NIST Method

The National Institute of Standards and Technology (NIST) is one of the institutions responsible for developing minimum standards, guidelines and requirements to provide adequate information security for all assets and parties with digital forensic competence.

1) Acquisitions: The first step in this research process is to identify data sources, The data acquisition phases that relate to certain events that will be identified, collected and protected. Table II is a table of needs of tools and materials needed.

TABLE II. TOOLS COMMONLY USED TO SUPPORT A VARIETY OF NETWORK FORENSICS INVESTIGATIONS

No	Tools	Description
1	GRR Server Computer	Intel i7 CPU, 32GB RAM, HDD 250GB
2	GRR clint Computer (Linux)	Intel i7 CPU, 32GB RAM, HDD 250GB
3	GRR clint Computer (Windows)	Intel i7 CPU, 32GB RAM, HDD 250GB
4	GRR Server and client (software)	Version 3.2.3.2
5	GRR Server operating system	openSUSE Leap 15.0
6	GRR Client operating system #01	Ubuntu 18.0.4 (LTS)
7	GRR Client operating system #02	Windows 10
8	Hammer DDos Script [26]	A Python3 script to launch DDoS attack
9	Switch	CISCO Catalyst 2960 Plus

To identify each computer on the network, in this research we give 192.186.100.0/24 network to three computers (openSUSE, Ubuntu, and Windows) as seen in Table III.

TABLE III. IP ADDRESS OF EACH HOST

No	Host	IP Address
1	openSUSE Leap 15.0	192.168.100.115/24
2	Ubuntu 18.0.4 (LTS)	192.168.100.18/24
3	Windows 10	192.168.100.10/24

2) Examination: After data has been acquired, the next phase is to examine the data, which is identifying, collecting, and organizing the relevant pieces of information from the acquired data. This phase may also involve bypassing or mitigating operating system or application features that obscure data and code, such as data compression, encryption, and access control mechanisms. Is a phase of testing the right tools and techniques for the type of data collected during the first phase to identify and analyze relevant information from the data obtained.

3) Utilization: Data utilization is the process of preparing and presenting information that resulted from the examination phase. Many factors affect data utilization, including data reduction, alternative explanations, audience consideration, and actionable information. The last phase involving the process of reporting and practice in the context of current events to identify policy shortcomings, procedural errors, and other issues need to be corrected.

The utilization process on Grr rapid response framework point of view is implemented by the inner working [19] of GRR Flow:

1) The GRR server starts by executing the initial Flow state.
2) Then the state asks for one or more client actions can be performed on the client.
3) The GRR server clears all the resources this Flow has requested and waits for responses from the client.
4) When message responses are received, the server fetches all the requested resources again and runs the Flow state where these responses are expected. If more client actions are requested by this state it goes back to step 2.
5) Otherwise, the results of this Flow are stored and the flow state is updated.

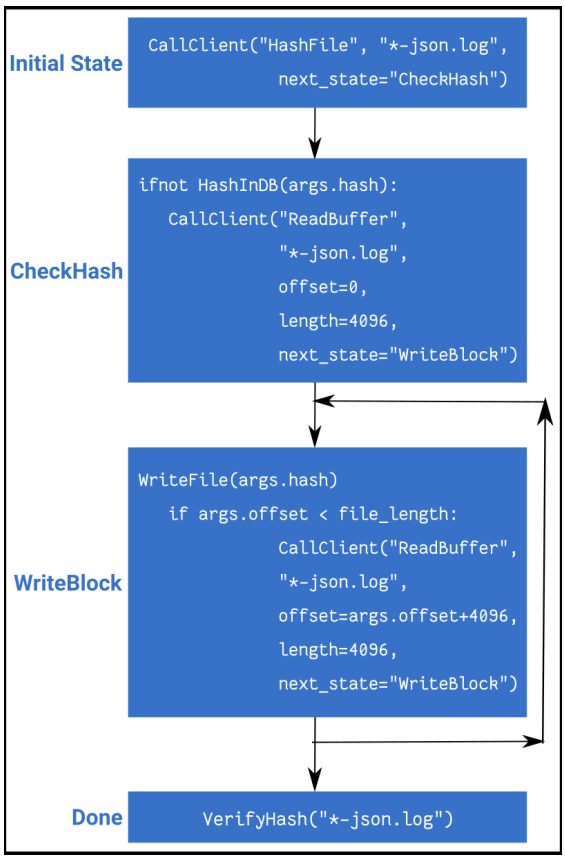

Fig. 7. A Flow to copy a log file from the client

Fig. 7 [19] shows a typical flow to copy a log file. First, GRR server sends a request message to Grr client, requesting the hash of a log file. After this request received by particular client, the GRR Flow is suspended and serialized to disk. When the client becomes available, the request is carried out and sends responses message to the Grr server. The server can then resume the flow and push the responses to the next state.

4) Review: Analysts should continuously review their processes and practices within the context of current tasks to help identify policy shortcomings, procedural errors, and other issues that may need to be remedied. Periodic refreshing of skills through coursework, on-the-job experience, and academic sources helps ensure that people performing data analysis keep pace with rapidly changing technologies and job responsibilities. Periodic review of policies and procedures also helps ensure the organization stays current with trends in technology and changes in law.

III. Result and Analysis

Based on the results and analyzer of the research that has been done, here is the criteria of the analyzed parameters used to clarify what the expected results has been made, as seen in the Table IV.

TABLE IV. Parameters used for the analysis process

No	Parameter	Result
1	Could digital evidence (log files) be obtained?	Yes
2	Could identity (IP address) of the attacker be obtained?	Yes
3	Could the digital evidence (log files) be trusted?	Yes

To identify and getting the process of digital forensic of the research, the following are the steps taken on getting digital evidence (log files) produced by scalable web application running one Docker Swarm cluster.

A. Acquisition

The acquisition of this research is to run the GRR Rapid Response framework in proper places, including to check the minimal requirements. In Fig. 8 we can see that all Grr clients are already running and ready to investigate.

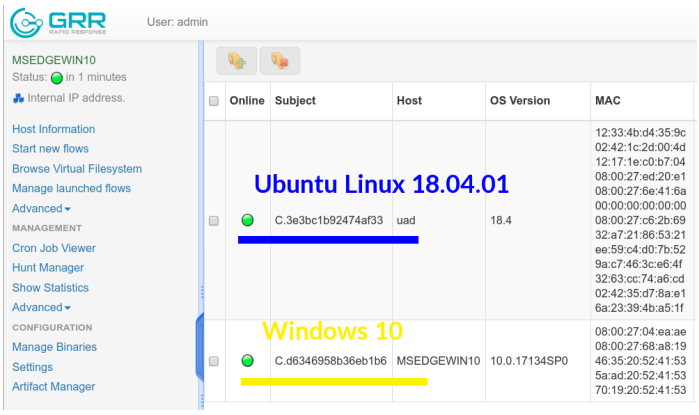

Fig. 8. Two GRR clients: Ubuntu 18.04.01, and Windows 10

The other requirements that have to be prepared on the purpose of acquisition on this research are:

1) All main components (Worker, AdminUI, and FrontEnd) of GRR server are already running on server computer.

2) All GRR clients are running on each particular computer.
3) On Ubuntu computer:
   a) The Docker Swarm has to be initialized first, and choose one of the node as a Swarm Manager, then add at least one node to become the worker.
   b) Run the scalable web application on the Docker Stack [27] so this application can be distributed on cluster system.
4) Run DDoS attack on Windows computer, the destination IP address of the DDoS script is the IP of victim computer (Ubuntu).
5) Finally, runs acquisition on GRR Server WebUI.

1) Acquisition on Docker Swarm Cluster (Victim): The acquisition in Docker Swarm cluster environment, we must create a custom ArtifactCollectorFlow because of collecting log file produced by Docker is not available on default installation GRR Server. So this is the dockerlogs.yml file as seen in Fig. 9.

```
name: LinuxDockerFiles
doc: Collect stat of all Linux Docker log files
sources:
- type: LIST_FILES
  attributes:
    paths:
      - '/var/lib/docker/containers/*/*-json.log'
labels: [Logs]
supported_os: [Linux]
```

Fig. 9. Custom ArtifactCollectorFlow: dockerlogs.yml

Upload the dockerlogs.yml file through Artifact Manager on GRR AdminUI, named it: LinuxDockerFiles, and begin to launch the Flow, as seen in the Fig. 10.

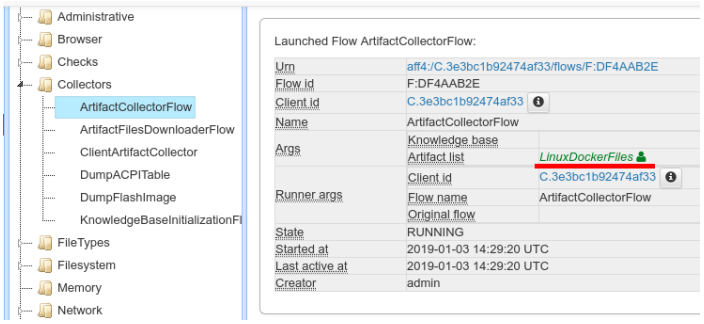

Fig. 10. Launching ArtifactCollectorFlow: LinuxDockerFiles

Depending on the availability of the client, this acquisition process will take about 10 to 15 minutes, of course there other possibilities involved to get the exact time consuming this process.

After we are done in the process of acquisition on the Ubuntu side as a victim computer, next step is going to examine the result in the following step after we collect other digital evidence from the view of Windows 10 as an attacker.

2) Acquisition on Windows (Attacker): To complete the acquisition on the client side, we have to do another acquisition, to prove that the attacker was coming from this client. To do this, GRR Server provides Flow Artifact called Netstat. Third artifact collector has a purpose to gain network information and status of the interface card on the particular computer, including IP address source and destination, port number involved, the type of connection (TCP or UDP), process name and the state of the particular connection, etc.. So to begin the acquisition, as not so different as on Ubuntu client, the process take the same step as we see in Fig. 11.

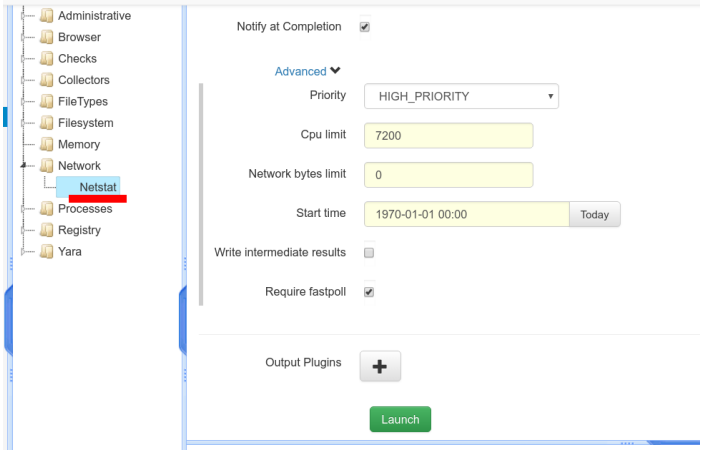

Fig. 11. Launching ArtifactCollectorFlow: Netstat

To see the preview that the Netstat ArtifactCollectorFlow has been launched, we can see it like we did on Ubuntu process.

In Fig. 12 we can see the Flow Netstat which the task is to collect network status on Windows 10 (attacker) has been successfully launched.

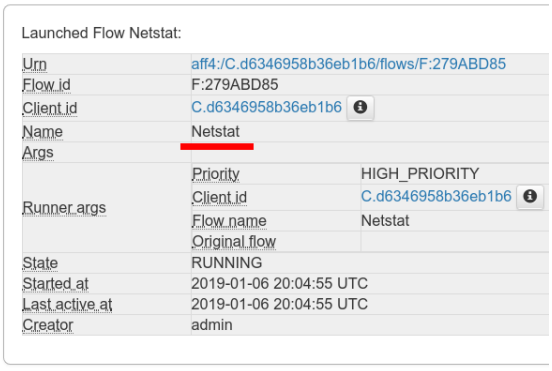

Fig. 12. Flow Netstat

After all acquisition processes both on Ubuntu Linux and Windows are finished, then we will go to the next process: Examination.

B. Examination

Like as we did in the previous process (Acquisition), we are now going to examine the result of the acquisition on both clients side.

1) Examination on Docker Swarm Cluster (Victim): Examination process in Docker Swarm Cluster on Ubuntu Linux is getting the result back from the GRR client. In this research,

Grr rapid response finally gets the examination process done with no hassle.

This examination process on GRR server is scalable as we can do the same thing not just on single client, but also for hundreds or even thousands of clients. This feature is called Hunt [21]. A GRR Hunt specifies a Flow, the Flow parameters, and a set of rules for which client computers to run the Flow on.

Fig. 13. Flow LinuxDockerFiles Response Message from the client

In Fig. 13 showing that the GRR Sever has finally found and collected the results as a manifestation of the Flow Response Message from the GRR client, so the examination process on Docker Swarm Cluster will take us to the valuable information, the source of the attacker, destination of port number, and timestamp. This important data will be discussed in the latter steps after finishing examination process on Windows computer as attacker.

2) Examination on Windows (Attacker): In the Manage launched Flows on GRR WebUI interface, we finally are able to collect network information on attacker computer that runs DDoS attack script. This response from the client is received by Server, and we are going to utilize it in the next step.

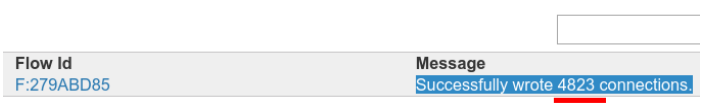

Fig. 14. Flow Netstat Response Message from the client

Fig. 14 shows the report of Message Response from Grr client that acts as attacker in this research.

C. Utilization

GRR Server utilizes the Message Response returned by GRR client in the proper and easy-to-use way. So in this utilization process, we are also have a great help from this excellent tools provided by Grr Rapid Response framework, by exploring the web interface with only clicking the available menu.

This step also will give us the appropriate information from both targeted investigation clients: Docker Swarm cluster on Ubuntu Linux, and DDoS attacker on Windows.

1) Utilization on Docker Swarm Cluster (Victim): Docker Swarm Cluster deployed on Ubuntu Linux has numbers of powerful utilities to provide and expand the usage of cluster system. One of the great feature is Docker Logs [28] where the instance of Docker container puts the log (output and error log) inside a log file on host file system, so practitioner can make use of the information provide by Docker Logs.

In this research, we can finally collect the log files and utilize them through GRR AdminUI component.

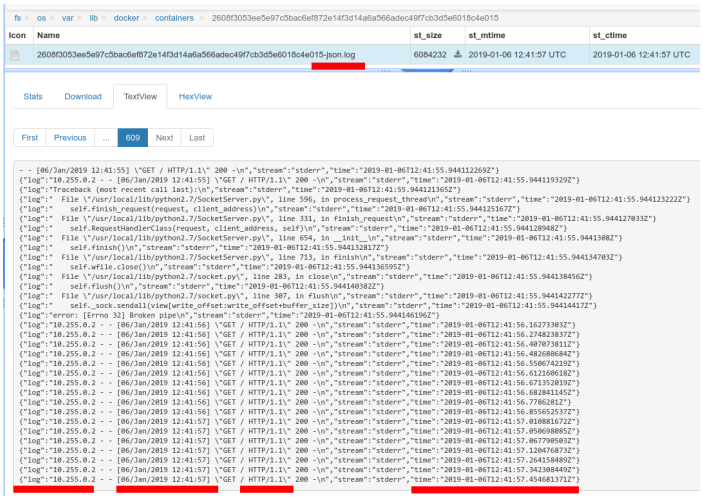

Fig. 15. Utilization of LinuxDockerFiles Response Message

Fig. 15 shows the result of utilization from resulting investigation on the Docker Swarm cluster. But as we can see, the IP address source is not coming from the original computer, it should 192.168.100.10, but it is 10.255.0.2. This IP is coming from the ingress network component [29] produced by Docker when it is initialized Docker Swarm cluster for the first time. it is used for every node so they can publish ports for services to make them available to resources outside the Docker Swarm cluster.

2) Utilization on Windows (Attacker): To make sure that the attacks occurred coming from Windows 10, we can elaborate with data examined from the previous step and utilize it with the information shown in Fig. 16, so we can have a proper and responsible conclusion.

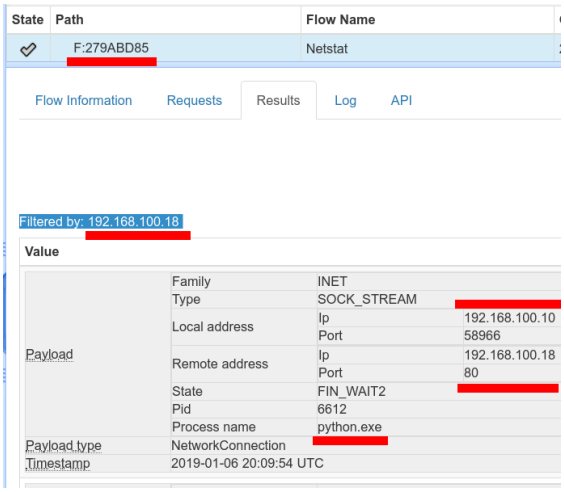

Fig. 16. Utilization of Netstat Response Message

In Fig. 16 we can finally find the origin identity of attacker, it was coming from computer that has IP address 192.168.100.10, as we expected.

D. Review

Based on the investigations that have been conducted starting from acquisition, testing, utilization, then the last step is to do a review. Grr Rapid Response framework has successfully managed to get digital evidence using live forensics through computer network. The evidence is in the form of a log file that is living inside host file system, which is then carried out and analyzed. Grr Rapid Response framework managed to get evidence in the form of an IP address source and destination, port number, and timestamps.

IV. Conclusion

Based on the research that has been investigated, Grr Rapid Response framework successfully accomplished the acquisition and analyzed the log file of scalable containerized web application running on cluster system built by Docker Swarm. Grr Rapid Response framework managed to obtain evidence in the form of IP addresses, port number, and timestamps. In the future work, Grr Rapid Response can be developed to identify digital evidence not only on embedded systems, but also smartphones.

References

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

[2] Datadog, "8 surprising facts about real docker adoption," https://www.datadoghq.com/docker-adoption/, 2018.

[3] T. Combe, A. Martin, and R. Di Pietro, "To docker or not to docker: A security perspective," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 54–62, 2016.

[4] Docker, "Docker swarm," https://docs.docker.com/engine/swarm/, 2018.

[5] Docker Team, "Swarm concept," https://docs.docker.com/engine/swarm/key-concepts/, 2018.

[6] N. Naik, "Building a virtual system of systems using docker swarm in multiple clouds," in *Systems Engineering (ISSE), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1–3.

[7] L. Jingna, "An analysis on dos attack and defense technology," in *Computer Science & Education (ICCSE), 2012 7th International Conference on*. IEEE, 2012, pp. 1102–1105.

[8] I. Riadi, "Internet forensics framework based-on clustering," *Editorial Preface*, vol. 4, no. 12, 2013.

[9] G. Palmer *et al.*, "A road map for digital forensic research," in *First Digital Forensic Research Workshop, Utica, New York*, 2001, pp. 27–30.

[10] NIST-a, Information Testing Laboratory, "Computer forensics tool testing program," www.cftt.nist.gov, 2012.

[11] NIST-b, "Guide to integrating forensic techniques into incident response," http://csrc.nist.gov/publications/nist-pubs/800-86/SP800-86.pdf, 2012.

[12] R. Umar, I. Riadi, G. M. Zamroni *et al.*, "Mobile forensic tools evaluation for digital crime investigation," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 3, pp. 949–955, 2018.

[13] F. Albanna and I. Riadi, "Forensic analysis of frozen hard drive using static forensics method," *International Journal of Computer Science and Information Security*, vol. 15, no. 1, p. 173, 2017.

[14] A. Yudhana, I. Riadi, and F. Ridho, "Ddos classification using neural network and naïve bayes methods for network forensics."

[15] M. A. Zulkifli, I. Riadi, and Y. Prayudi, "Live forensics method for analysis denial of service (dos) attack on routerboard."

[16] A. K. Kaushik, E. S. Pilli, and R. Joshi, "Network forensic system for port scanning attack," in *Advance Computing Conference (IACC), 2010 IEEE 2nd International*. IEEE, 2010, pp. 310–315.

[17] I. Riadi, J. E. Istiyanto, A. Ashari *et al.*, "Log analysis techniques using clustering in network forensics," *arXiv preprint arXiv:1307.0072*, 2013.

[18] Oleg Kupreev, Ekaterina Badovskaya, Alexander Gutnikov, "Ddos attacks in q3 2018," https://securelist.com/ddos-report-in-q3-2018/88617/, 2018.

[19] M. I. Cohen, D. Bilby, and G. Caronni, "Distributed forensics and incident response in the enterprise," *digital investigation*, vol. 8, pp. S101–S110, 2011.

[20] GRR Developers, "Grr software license," https://github.com/google/grr/blob/master/LICENSE, 2011.

[21] A. Moser and M. I. Cohen, "Hunting in the enterprise: Forensic triage and incident response," *Digital Investigation*, vol. 10, no. 2, pp. 89–98, 2013.

[22] R. Hunt and S. Zeadally, "Network forensics–an analysis of techniques, tools, and trends," *Computer*, pp. 1–1, 2012.

[23] F. Cruz, A. Moser, and M. Cohen, "A scalable file based data store for forensic analysis," *Digital Investigation*, vol. 12, pp. S90–S101, 2015.

[24] S. Acharya, W. Glenn, and M. Carr, "A grreat framework for incident response in healthcare," in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2015, pp. 776–778.

[25] H. Rasheed, A. Hadi, and M. Khader, "Threat hunting using grr rapid response," in *New Trends in Computing Sciences (ICTCS), 2017 International Conference on*. IEEE, 2017, pp. 155–160.

[26] Can Yalçın, "Hammer ddos script," https://github.com/cyweb/hammer, 2014.

[27] Docker, "Get started, part 5: Stacks," https://docs.docker.com/get-started/part5/, 2014.

[28] Docker Team, "Docker logs," docs.docker.com/config/containers/logging/, 2018.

[29] Docker Doc Team, "Docker swarm ingress," https://docs.docker.com/engine/swarm/ingress/, 2018.