Service Oriented Computing and Applications (2020) 14:175–187
https://doi.org/10.1007/s11761-019-00285-7

ORIGINAL RESEARCH PAPER

Analysis of adversary activities using cloud-based web services to enhance cyber threat intelligence

Hamad Al-Mohannadi[1] · Irfan Awan[1] · Jassim Al Hamar[2]

Received: 15 May 2019 / Revised: 18 December 2019 / Accepted: 31 December 2019 / Published online: 21 January 2020

Abstract
The understanding of cyber threats to a network is challenging yet rewarding as it allows an organisation to prevent a potential attack. Numerous efforts have been made to predict cyber threat before they occur. To build a threat intelligence framework, an organisation must understand attack data collected from the network events and analyse them to identify the cyber attack artefacts such as IP address, domain name, tools and techniques, username and password, and geographic location of the attacker, which could be used to understand the nature of attack to a system or network. However, it is very difficult or dangerous to collect and analyse live data from a production system. Honeypot technology is well known for mimicking the real system while collecting actual data that can be in near real time in order to monitor the activities on the network. This paper proposes a threat intelligence approach analysing attack data collected using cloud-based web service in order to support the active threat intelligence.

Keywords Threat intelligence · Cyber threat · Honeypots · Cloud services · Log analysis · Elastic Stack

1 Introduction

Cyber attacks are continuously growing and becoming significant concerns for all types of organisations. Organisations are putting several protection measures in place including regular penetration tests, setup intrusion detection system (IDS) and intrusion prevention system (IPS) devices, real-time monitoring systems, firewalls, etc., to prevent cyber attacks. However, these systems are attached to the organisation's production system. Efforts are being made to educate staff members to avoid unexpected phishing attacks and human errors. In most cases, companies are failing to make staff aware of cyber attack knowledge [1]. On the other hand, criminals are finding new ways of attacking and stealing company assets. They are increasingly making organised attacks on big organisations and public services. Such attacks, known as advanced persistent threat (APT), are becoming a significant issue [8], the attacker does not launch an attack without conducting research and planning. They are trained adversaries and use sensitive tools and techniques to target confidential information from high profile victims [14].

Identifying a cyber threat before it occurs is a complicated process for any network administrator or security personnel. It is quite challenging if it is a production system. So, it is essential to find a system that could act as a real system to the attacker and collects valuable information about attack events. Cyber threats can be identified using honeypot data collection and analysis, which gives an understanding of the nature of a cyber attack. For example, an SSH honeypot can be analysed while the session is running and the data is visualised using a visual analytical technique [35], which can reflect the real-time activities on the network. The main advantage of a honeypot is that they produce a huge amount of log data, which records each of the events that occurred with the time stamp. Log analysis using Elastic Stack has been used to enhance threat intelligence, which then helps in increasing the network security by taking appropriate measures. However, developing a general-purpose big data analysis tool is hard, so, honeypot could support for data collection. Honeypots and honeynets are well-known uncon-

✉ Hamad Al-Mohannadi
almohannadi7@gmail.com

Irfan Awan
I.U.Awan@Bradford.ac.uk

Jassim Al Hamar
j.alhamar@hotmail.com

[1] School of Electrical Engineering and Computer Science, University of Bradford, Bradford, UK

[2] Ministry of Interior, Doha, State of Qatar, Qatar

ventional cloud-based security services that allow collecting data and analyse them in order to learn more about cyber attack Sokol et al. [32] collected data from honeypots to hunt cyber-attack patterns.

Moor et al. [22] collected honeypot data including IP addresses of attackers for further analysis. Organisations information and technology assets such as hosts and applications are protected by various IDSs and IPSs that could be automated for identifying and mitigating cyber threats. However, automating cybersecurity tools may not be the complete solution for protecting valuable assets within an organisation. This paper exploits elasticsearch technique to analyse honeypot attack data since it provides the facility of flexible searching on the high-volume data set. This paper also proposes a threat intelligence approach to understand the attack pattern and behaviour of an adversary. The threat intelligence technique is evaluated following the collection of data. In order to achieve the goal, we have deployed cloud honeypots as services to find cyber-attack-related events through data analysis applying elasticsearch. The result demonstrates that honeypot data analysis could be used in cyber threat intelligence to support network protection for organisation.

This paper organised as follows. Section 2 discusses a wide range of related work for a better understanding of existing cyber-threat intelligence techniques. Section 3 analyses cyber threat hunting and proposes a new threat intelligence model. This section also presents an initial conceptualisation to describe the proposed threat intelligence model formally. Section 4 includes the experiment setup using an Elastic Stack technology and discusses the research outcomes. Section 5 evaluates the result of the experiment. Finally, Section 6 concludes the research work presented in this paper and identify the direction to the future work.

2 Related work

Organisations use various security of tools such as IDS, IPS, firewalls, anti-virus software, traffic shaper or sniffers in order to protect their assets. In most cases, these are rule-based detection systems that allow or reject traffic. Cybersecurity is a long-term process, which needs continuous monitoring of network traffic and improvement of the protection system. Therefore, it is essential to find advanced cyber threat managing focusing analytical methods. Threat hunting requires a more sophisticated approach than the traditional rule-based detection systems [5], as it helps the organisation to look for cyber threat proactively. Proactive cyber threat intelligence requires continuous data analysis from attack-related data. Jasek et al. [15] used honeypot technology to detect cyber attacks, which is an excellent resource that provides flexibility to identify cyber attacks compared to other techniques. Honeypot data analysis finds anomalies to detect potential cyber attacks on the organisation's production system such as database server and web server. A hybrid system called HALF is proposed by Angiulli et al., which operates on data using data mining techniques to find network intrusion anomalies [3].

In the cyber security world, a distributed denial of service attack (DDoS) is a challenging threat to any organisation as a flood of incoming connections could crash the whole network. Weiler [36] simulated the DDoS attack using honeypots to investigate cyber attacks on network infrastructures.

Cyber threats in mobile devices can also be emulated using honeypots like Nomadic [17], which provides infrastructure to collect threat intelligence data to understand the threat level in smart devices. Network monitoring can be carried out data collection, analysis and visualisation techniques. Dionaea, a low-interaction honeypot, is used to collect and analyse attack-data to understand the trends of cyber attacks and to create a profile of the attacker from the analysed data [16]. Security tools such as honeypots, sandbox and virtual machines are called emulated monitoring system. Papazis et al. [25] investigated into an indicator of deception for emulated monitoring systems and provided a taxonomy.

Pursuing cyber threat is not a new concept and has always been a focus of researchers. A number of techniques have been reported in the literature for actively searching threats. Miloslavskaya et al. [20] have presented a taxonomy for unsecure data process in security operation centres. This taxonomy classifies the information security (IS) threats (a threat of IS violation) is a set of conditions and factors that create an actual or potential opportunity for violation of IT assets [20].

The threats are usually modelled using various modelling techniques to analyse cyber attacks, such as Attack Graphs or Trees [26,28], to calculate the path between the attacker and the victim. The diamond model of cyber attack modelling deals only with the adversary, victim, capability and infrastructure. This is very simple but useful to understand complex cyber attack [7]. Kill Chain method of cyber attack was derived from the military, which has a series of steps like reconnaissance to Command and Control (C2C) [14,34]. However, a generic intrusion scenario was described by Graham [11], which includes detail discussion about the network intrusion detection system. Other models such as attack vectors [23], attack surface [18] and the open web application security project (OWASP's) threat model [37] are used to understand the cyber threat. These modelling techniques can be used individually or in conjunction with other models. An overview can be found in Al-Mohannadi et al. [2], which describes a number of cyber-attack modelling techniques developed to handle cyber attacks. Cyber-attack modelling techniques mainly concerned with identifying the attack patterns of the adversary but limited to provide an early intelligence before attack event. On the other hand, cyber threat

intelligence is a process of monitoring network traffic, collecting data and analysing event data to find anomalies as well as visualisation techniques, linked data analysis and model building [33]. The following subsections discuss details of some techniques about proactive threat intelligence.

2.1 Pyramid of pain

Binaco [5] has introduced the Pyramid of Pain to analyses how an Indicator of Compromise behaves. The idea of an Indicator of Compromise is that it identifies network-related components such as IP address, open port and domain address that could be the weakness of a network during the cyber attacks. The main idea of the Pyramid of Pain is to establish the different levels of Indicator of Compromise for cyber defence. The pyramid indicates the level of difficulty in handling cyber threats. The Indicator of Compromise, therefore, defines the components of the Pyramid of Pain. Figure 1 shows the Pyramid of Pain, which indicates levels of technical difficulty for both the adversary and the victim. The Pyramid of Pain provides a simplified view of the adversary's activities on the system. An adversary uses the Pyramid of Pain components for developing an attack on a network and leaves a footprint.

Thus, analysing a Pyramid of Pain could indicate the behaviour of the adversary in order to build or improve protection system. One of the crucial components of the pyramid is at the bottom layer is known as the hash value, which provides unique references for specific malware or to the payload that is used for the attack. Hash values are changed for a simple action on a file; for example, a minor change to the payload changes the hash. Therefore, keeping track of hash values is not worth as the nature of change. This means that attacks using hash values are easily identified and tackled, so the possibility of a system compromise is very low. On the other hand, IP addresses are fundamental indicators for identifying an attacker. Hiding IP addresses during a cyber attack is hard for an attacker; however, it is very easy to change the IP address that is used for the attack or hide. Practically, it is not possible to follow up every single IP addresses that have attempted to attack a system.

In order to get a domain name, the adversary must have registered with a hosting company, which can be found in the hosting database. So, technically, it is easy to trace the origin of the domain used by an attacker, although attackers could be disguised. Domain name users have to register; it is difficult to change domain names compared to the IP addresses.

Network artefacts are one of the critical indicators that differentiate the malicious activities from an attacker and a valid user. A host or a workstation that is attacked contains a huge amount of information. On the other hand, the tools that an attacker used to perform an attack could be unknown by the security personal. These tools could be used to prepare

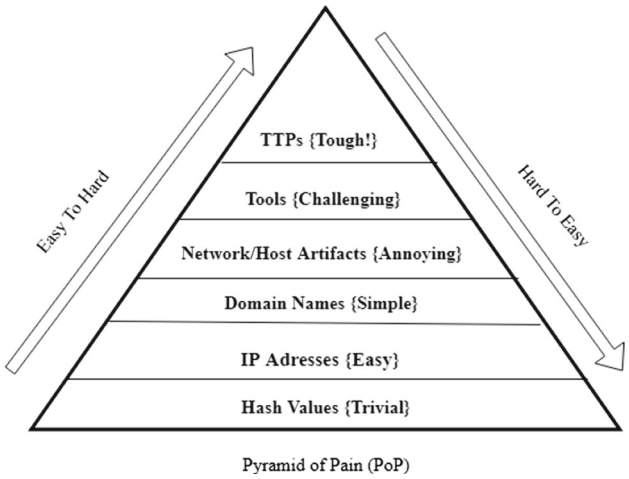

Fig. 1 Pyramid of pain. (Adapted from [5])

payloads, which could be different than regular attack tools. Finally, at the top of the pyramid, the Tactics, Techniques and Procedure (TTP) [21] could be used to identify the attacker, their behaviour, the malicious software tools and the payload.

2.2 Hunting maturity model

Finding potential threat that identifies an organisation's ability to build better protection by collecting and analysing threat data. Hunting maturity model (HMM) indicates a way of analysing and visualising threat data [33]. HMM includes of five levels of maturity of finding cyber threat.

Level 0 indicates that an organisation depends on IDS provided by third party, which could be automated. Higher-maturity level indicates the organisation's data collection and threat analysis. Therefore, the highest level of hunting maturity indicates that the organisation is involved in regular data collection using automated systems and analyse them to find anomaly. Threat hunting maturity is a linear process that improves an organisations data collection and analysis as illustrated in Fig. 2.

The process of threat hunting is not a one-off action; it must be a continuous process which employs the development hypothesis, identification of investigation tools and techniques, identification of new patterns through enhanced analytics. The Sqrrl Data [33] introduced the hunting loop and, as illustrated in Fig. 3, can be matched with the hunting maturity model to identify the strength of the organisation's data collection analysis. The process could be generalised and automated for similar types of cyber threats.

The following discuss the HMM process in detail:

– *Data Collection* Data collection is one of the important phases of the HMM in order to hunt real threats in the network. Data could be collected in many different formats

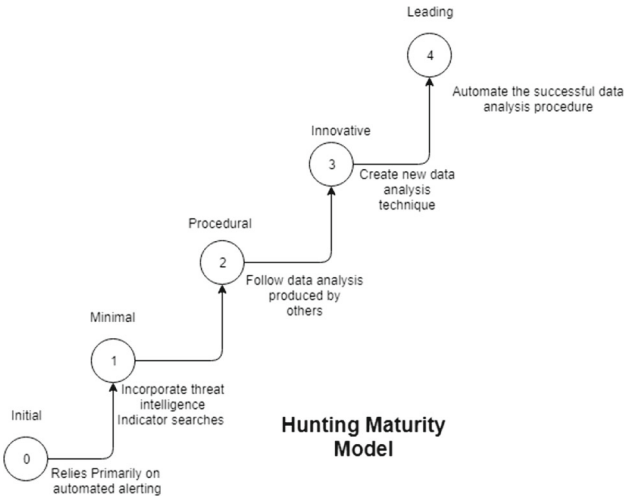

Fig. 2 Hunting maturity model. (Adapted from [33])

such as log from different sources such as honeypot, firewall or server. These data could also be collected using an automated process.

- Hypothesis Creation: Creating a hypothesis is very important in threat hunting. It is important to understand if the current threat hunting process is working or not. So, it is essential to review existing systems including IDS, IPS and firewall. The hypothesis needs a proper assessment within a typical network. Collected data could be used to create a hypothesis, which needs to be reviewed regularly. The hypothesis may need to change as required.
- *Tools and Techniques* There are a number of tools and techniques that can be used to identify a threat. SIEM and log analysis tools provide a minimum level of threat hunting in the network. So, it is necessary to test hypothesis against the tools and techniques in order to do active threat hunting.
- *Pattern and TTP Detection* There are different types of cyber attacks such as APT [15] or zero-day attacks [27] that are difficult to predict in advance or identify. A zero-day attack does not match any known attack pattern that has been experienced by the experience or anyone else. So, it is crucial to have higher at the hunting maturity model and to hunt for new types of threat.
- *Analytic Automation* Tools and techniques are the heart of cyber threat hunting. This is impossible to manage all these tools manually requires automation, which is an important factor in such situations. Automation needed to be applied in all level of threat hunting, detection and mitigation.

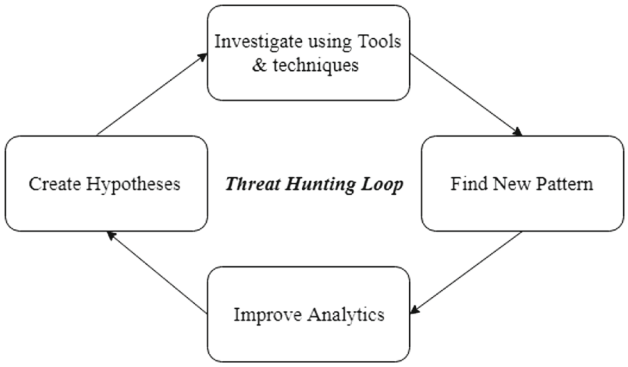

Fig. 3 Threat hunting loop

2.3 Matrix of indicator of compromise

The matrix of the Indicator of Compromise [5] evaluates cyber threat using three criteria such as trace, identify and response. In the following, we discuss three rules against Indicator of Compromise for a better understanding of those indicators.

- *Trace* It is essential to trace an attacker during their visit to a network or a host. Tracing a hash value is not entirely beneficial as the attacker could change it during the next attack. On the other hand, if the payload is changed, the hash value will be different. It is therefore difficult to identify if the same attacker performs any subsequent attack. There is always an IP address used in a cyber attack situation, which can be changed at any time by the attacker.
  Network host artefact is one of the important evidence that an attacker may leave behind although they might change the IP address or domain name. The investigator may need to perform an enhance investigation on the network artefacts. It is possible that some of the tools are being used repeatedly to make an attack on victim's network or host. The attack tools need to be tested, where applicable TTP needs to be used from the top layer of the Pyramid of Pain.
- *Identify* It is essential to trace an attack to find footprints of the attacker. The evidence left by the attacker could be used to prevent a future attack. IP address tracking system could be used to understand more about the attacker in real time, which could potentially identify the attacker's behaviour.
- *Response* The response is important for a cyber attack for the prevention purpose. If an IP address is detected as an attack or threat component from data analysis, it could be blacklisted for any future traffic through the network. A prompt response can help identify a potential cyber attack in the network [5].

2.4 Honeypots

Understanding and predicting a cyber attack is a difficult task. This requires active threat hunting using big data analysis. Honeypot technology provides a safe way of collecting cyber threat-related data. It is essential to understand that honeypot attracts attackers to interact with the honeypot. It is safe as the attacker does not know about the production system. So, it is convenient to collect data through honeypot and analyse them to understand the nature of the attack. Some researchers used honeypot technology to understand various types of attacks such as malware attack, botnet activity, fishing and spreading spam.

A honeypot could be deployed within or outside of the network of an organisation. However, it is not safe to implement honeypot in a network where there is a possibility of getting to the production system being exposed. A honeypot can also be deployed in the cloud [6] as the cloud services are the secure option since the cloud providers handle all the security aspects. Cloud honeypots can be used as an additional security service for cloud users, which could be considered as Infrastructure-as-a-Service (IaaS). HoneyC is a low-interaction client-based honeypot, which emulates only essential features of target clients. This is a client honeypot that can detect client-side attacks and record them in log files. In essence, it uses simulated clients to interact with real servers. HoneyC is a platform-independent framework, which consists of three main components: the Queuer, Visitor and an Analysis Engine [29].

Honeypots are classified by attack resources and level of interaction. An attack resource could be the honeypot interaction as a client or a server. A server-side honeypot acts like a server that listens on the port for any request from a client. On the other hand, client honeypot consists of client-side application such as a web browser and can connect to a remote server or service. The advantage of using a honeypot to collect data is that it can play both client and server roles. In this paper, we use Kippo, which is a low–medium-interaction honeypot in a Linux environment. Kippo is an SSH honeypot. It simulates the file system, which records all the attack logs for a brute force attack. This can also record the behaviour on the operating system. This can emulate the whole system and appears as a fully functional machine to the attacker [12].

3 Threat intelligence

Intelligence is the combination of data, information and knowledge, which can be defined as 'evidence-based interpretation of data, collected on or against the identities goals, motives, TTPs and targets of malicious actors' [4]. Threat is a set of states and factors that could create an environment of violating organisation asset [20] where intelligence can be the information that can be used to change outcomes [19]. The intelligence can also be a particular kind of information formulated from the Indicator of Compromise such as the IP address, username, password, network port etc. If an attack event is originated from an IP address, it is essential to know about the IP address. Cyber Threat Intelligence involves in the knowledge of the cyber threat and acquisition of information that is relevant, valuable and available, which allows to prevent or mitigate cyber attack [9]. So, cyber threat intelligence can help the victims or defenders to identify on going or potential cyber attacks [30].

The conceptual model of threat intelligence consists of three components as described below:

- *Attack* An adversary originates, to a system or a network to gain access and control. An attack could be successful or unsuccessful. However, an attack always leaves a footprint to trace back. In terms of attack, the threat intelligence may ask a question, why am I seeing this IP address several times? What is this IP address about?
- *Behaviour* Doing similar activities such as trying to accessing the system at the same time could indicate an adversary's behaviour. Threat intelligence looks for the repetitive behaviour from a historical or current data set.
- *Pattern* Pattern can be defined as similar event occurs repetitively. Cyber attack pattern is the combination of the attack event and the behaviour of the attacker over the time.

An attack is a systematic approach by an attacker to gain access into a system, a network or a host. A cyber-attack event data can be recorded using appropriate data collection. The behaviour of the attacker can be identified from the data collected if the same attacker attempted several attacks. In the event of cyber attack, an attacker is either a human or a machine. For both cases, the behaviour is an indicator of the method used. So, there is a good relationship between human behaviour and cyber attacks [24].

System logs are collected by most of the system for future use or record; however, they are rarely being used. These data can be considered as big data as the data have velocity, verity and volume [13]. If we think of only the volume of the data set, it would require special techniques to analyse and present. By analysing these data, we can identify attack events. These attack events could repeatedly occur over time, which could form patterns. The aim of using data analysis is to define such a pattern. Data can be analysed more intelligently and efficiently by using big data analysis techniques. Figure 4 shows the threat intelligence triangle, which can be used to understand the attack, pattern and behaviour of the attacker. Using this model, attack data could be separated from the regular one. The attack data tell the behaviour of

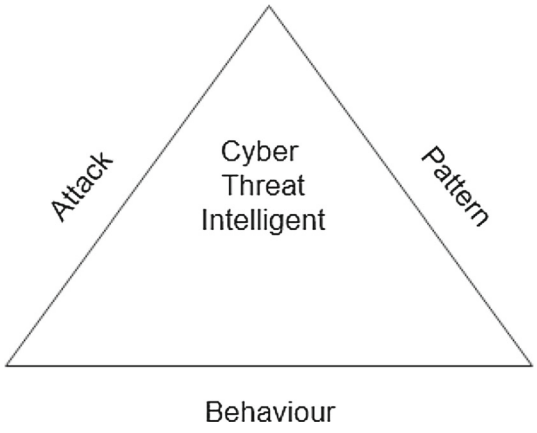

Fig. 4 Cyber-attack concept

the attacker as a result of the analysis. So, the pattern of the attacks can be revealed to mitigate cyber attack on the organisation's network.

We also designed a cyber threat intelligence system that allows analysing cyber threat data in real time to generate an alert. This architecture is to build threat intelligence for a corporate network. Honeypots can be installed in parallel to the servers and other security devices within the organisation. These honeypots would be low–high-interaction honeypots, which gives the attacker a feeling of interacting with a real system (e.g. computer). A low–medium-interaction honeypot such as Kippo acts as an operating system or a server that can collect valuable log data. Since the honeypot mimics the real device, it gives the opportunity to obtain near-real-time data for analysis. These honeypots can be installed in different locations to capture log data for different geo-locations.

3.1 Problem analysis

An initial conceptualisation of the cyber attack is described as follows.

Threat intelligence can be defined as tuple of three components, such as $\{A, B, P\}$, where

- ***A*** denotes Attack as a set of actions such as $\{a_1, a_2, a_3, \ldots, a_n\}$ on a systems.
- ***B*** denotes Behaviour, which includes any repetitive actions performed by an attacker.
- ***P*** denotes Pattern, which is the combination of attack and behaviour using intelligence.

An adversary from a remote location is the originator of a cyber attack. An attack will have one of the two outcomes: (a) successful, which means the victim's system was compromised or, (b) unsuccessful, which means the victim's system was not compromised.

A cyber-attack event is defined as a series of actions $\{a_1, a_2, a_3, \ldots, a_n\}$ performed by an attacker by using malicious tools and techniques to access valuable assets of a victim. The attack is performed through the Internet, which is an interconnected network. A cyber attack can be considered as a directed graph (V, E), where vertices V stand for nodes and edges E stand for a path. An attacker makes an attack from a node V_a to another node V_v, which is the victim's machine. The communication link between the attacker and the victim is the edge E. In the event of an unsuccessful attack, the path remains a single direction and terminates by itself.

The system or network is compromised if an attack is successful. In that case, the path between the attacker and victim becomes bi-directional, i.e. a connection is established. The Internet consists of a heterogeneous topology. However, we are only interested in the abstract edges and vertices since the path could be so long with hundreds of nodes in between. Moreover, we can collect a few artefacts of the attacker such as IP address and domain name, and most of the data are collected from the victim's machine or the infected node. Our primary interest in the vertices is that they identify the attacker's and victim's machines, which are V_a and V_v, respectively. In a victim's machine or network, the data, which we will call *assets*, X, could be in three different stages.

Assets $X = \{X_\mathrm{r}, X_\mathrm{p}, X_\mathrm{m}\}$, which represents that

- the asset is resting (X_r),
- the asset is in process (X_p), and
- the asset is on the move, respectively (X_m).

Let us assign $T \subseteq \mathbb{R}_0^+$ as a time stamp. The time stamp starts from zero and lasts until the attack session remains.

Let us assume that in a cyber-attack event, an attack starts at time t and lasts for $\triangle t$. Given the time stamp, we have formalised cyber-attack event as follows:

> *attack* an attacker comes to the contact of the victim's system at time t_1 with an action a_1 and leaves at time t_n . The elapse time is δt that depends on the activities of the attacker on the victim's machine or network.
> *access* attacker tries to access victim's *asset* by using some techniques such as brute force. If the attacker is successful for gaining accessing, he/she can advance towards the goal like command and control.
> *event* events Z in victims $node$ called V_v can be discrete, which can be stored in a series of the time stamp. In the event of a cyber attack at the victim's system, the time stamp is recorded. The event of each time stamps contains information that may or may not be attack related. We only consider the events that are related to cyber attack. The attack event is identified by the victim's $node$ and

recorded the time stamp as t_e, where e denotes the time of an event.

At t_e, an attacker starts a new connection using a protocol like SSH, which requires 'username' and 'password'. Whether the attack is successful of not, the attack event Z is recorded for each of the time stamps T. The attacker leave an artefact, which is an IP address. The attacker may use the same artefact repeatedly.

So, Behaviour B of an adversary is the tuple of three $= \{A, V, T\}$, which can be repeated several times in a similar fashion using similar IP address. However, the attack Pattern P could be extracted from the combination of the attacker's behaviour B and the attack Events Z where an attack event is represented as $Z = \{t_e((a_1 \times t_1), (a_2 \times t_2), (a_3 \times t_3), \ldots, (a_n \times t_n))\}$.

An attack event Z happens to a *node* at time stamp T is recorded by logging. The goal of the attacker is to get *access* to the system to get a valuable asset from the *node*. Each of the attackers is different and has unique behaviour as they all look for the different artefact at a different time. Also, attackers leave artefacts, such as IP address, hash values and domain name, that would help to track the attacker and their behaviour. The traces that an attacker leaves behind are significant for the cyber attack intelligence. Although IP address can be changed at any time, the attacker may keep trying to attack using the same IP address at the same time every day. This may tell a story about the motive of the attacker. This kind of behaviour can lead to pattern and could be generalised by analysing a huge amount of data.

In the following section, we design an experiment using honeypot data and analyse them to identify attack pattern and behaviour of attackers.

4 Experiment setup

The experiment has been set up using the Elastic Stack, which consists of Elasticsearch, Logstash and Kibana formerly known as ELK[1] stack. The Elastic Stack helps to present data and create visualisation and a dashboard for any size of data in real time. One of the advantages of using elasticsearch technique is that scalability is not an issue as it can handle big data and search is faster than other approaches. To support the Elastic Stack for data discovery, we used Filebeat to get multiple files to the elasticsearch. Figure 5 illustrates the architecture of the experiment. We briefly explain the Elastic Stack and associated technologies that are used in this paper as follows:

[1] https://www.elastic.co/.

– *Elasticsearch* Elasticsearch is one of the prominent search techniques that can search text from almost any format or platform. Elasticsearch is highly scalable and flexible [10]. This is also distributed and uses RESTful search technique, which ensures the exact search result. Elasticsearch is the heart of elastic technology.
– *Logstash* Logstash works as pipeline between the data and the Elastic engine, which provides the input stream to the engine. It is a log parsing engine, which uses JSON for parsing logs.
– *Kibana* Is the visualisation platform in the Elastic Stack, which is also highly scalable. Kibana can help user to create different types of charts including bar chart, pie chart, etc., and plots data to them from large volume. User can create multiple dashboard from the log analysis. It also allows to have visual search on the text using Elasticsearch.
– *Filebeat* Its main job is to push logs files to the Logstash, which makes the pipeline. Filebeat can handle multiple file sources at the same time.

Honeypot deployed in the cloud is considered as low to high interaction, which intends to attract attackers. The aim is to collect real-time data with a time stamp. To maximise threat hunting, we have installed two low–high-interaction honeypots called Kippo and Dionaea [31] on Amazon as cloud services. The location of these honeypots is in Amazon's cloud services system called EC2, which is located in China and the USA. The log also comes with time stamps that indicate when the event took place. We have collected over 5GB of honeypot log data for about three years.

The honeypots were installed in the AWS cloud that appeared as real operating systems, which attracts many attackers. These log data contain time stamp and date for each of the events. Events are recorded if anyone tries to interact with the honeypot. The collected data are huge in volume, which is very difficult to analyse simply by using regular tools. Therefore, we have adopted the Elastic Stack to find the meaning of the log data. The main advantage of Elastic Stack is that it combines elasticsearch and visualisation. Since the elasticsearch is highly scalable, it can search within any size of data. It can also do all related database operations such as create, read, update and delete. It can also connect with different types of application programming interfaces (APIs) for searching and analysing data. Many organisations such as Wikipedia use Elasticsearch for full-text searching, which is called search-as-you-type; GitHub uses it for searching 130 billion lines of code; and Stack Overflow uses it for full-text searching for geo-location queries. It is not only used by technology giants, but also by many startups for finding meaning within data [10].

Fig. 5 Experiment setup with ELK stack

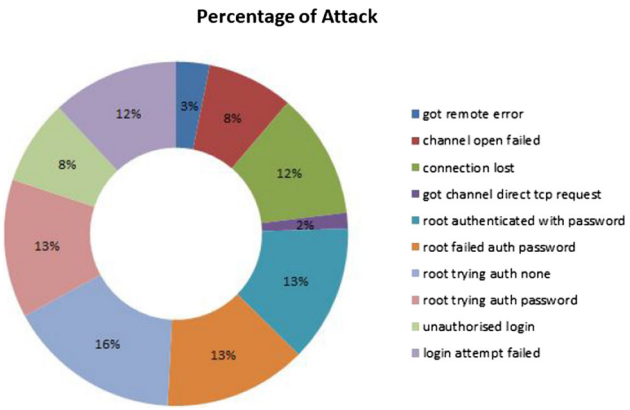

Fig. 6 Kippo honeypot log event visualisation using Kibana

Table 1 Attack events analysis

Event name	No. of time occurred	% of occurring
Root trying auth none	5,802,714	16.31
Root failed auth password	4,766,810	13.4
Root trying auth password	4,627,586	13.01
Unauthorised login	2,837,373	7.98
Got remote error	1,125,619	3.16
Got channel direct-tcpip request	528,303	1.49
Connection lost	4,198,733	11.8
Root authenticated with password	4,574,932	12.86
Channel open failed	2,864,106	8.08
Login attempt failed	4,246,430	11.94

5 Result and discussion

The Elastic Stack helped to performed keyword search and kibana provided the visualisation. The aim was to identify attack-related activities the honeypots. A significant number of attack events were identified from the log analysis using Elastic Stack. Figure 6 illustrates the attack events in the Kippo honeypot and about ten attack-related keywords were identified that indicate the attack events within the honeypot. However, most of the events that occurred in the network are not attack related. Some of the keywords, like remote error, connection lost, etc., are not attack related.

Each of the keywords in Table 1 identified from the honeypot data indicates cyber-attack events. Each of the events is presented by using Kibana visualisation in Fig. 6 and number of attack events presented in Fig. 7. The following explains the cyber-attack event keywords in detail for better understanding the attackers activities in the honeypot.

- *Root trying auth none* Attacker tried to get access to root but failed. In a UNIX-based system, getting access to the root gives attackers full control to the system. Since the authentication was not confirmed, the attacker could not get the access. This is the top event so far as the event has occurred 5,802,714 times, which is 16.31%.
- *Root failed auth password* Attacker's password not authenticated. The attackers tried different password and failed at every attempt. This is one of the highest occurring events as it happened 4,766,810 times, which is 13.4% of overall events.
- *Root trying auth password* In this event, attackers have been trying with password. This event has occurred 4,627,586 times, which is 13.01% of total events.
- *Unauthorised login* Unauthorised login detected in the honeypot for about 2,837,373 times, which is about 7.98%. This means that the honeypot system was compromised several times using password.
- *Got remote error* Unknown remote error occurred several times, which is not an attack-related keyword.
- *Got channel direct-tcpip request* Direct request for tunnelling is a remote request to make a tunnel between two systems to send and receive data. There were about 1.49% requested made. Any successful tunnelling may give the attacker an opportunity to get access to the victim's system.
- *Connection lost* Lost connection with remote host, which is considered as attack event, but there is no significance for this to be a cyber attack.
- *Root authenticated with password* There are many occasions, where 'root authenticated with password' event has been hit. This is one of the highest events, which is about 4,574,932 and 12.86%.
- *Channel open failed* This event happened 2,864,106, which is about 8.08%.
- *Login attempt failed* Login attempt is one of the common attempt that attackers make to a honeypot. In our honeypot, there are around 4,246,430 number of times login attempt has failed. The percentage of this event is

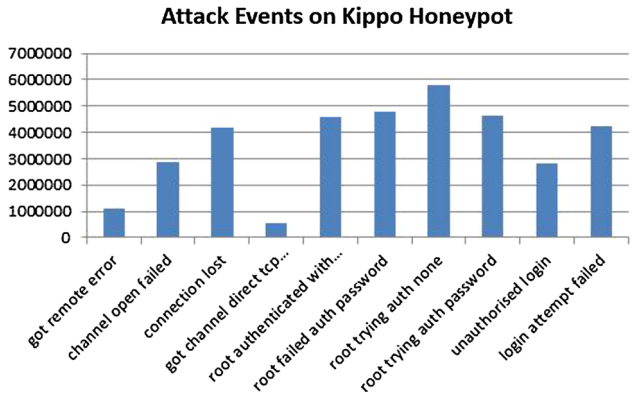

Fig. 7 Attack events in bar chart view

11.94%, which is very high compared to other events that happened to the honeypot.

The results are summarised in Table 1 to identify the statistics of those events that occurred. Login attempts are a serious attempt to get into any computer systems. The attacker tried to log in to the honeypot, where the outcome could be either 'success' or 'failure'. There are 7,083,803 times login attempts were made to the Kippo honeypot, which is about 19.92%.

We have noticed that 'root trying auth none' occurred some 5,802,714 times, which is about 16.31% of the total number of events found up to this point of data collection. Since the honeypots are Linux machines, the attackers try to access root. The event 'root failed auth password' occurred about 4,766,810 times; or a total of 13.4%. This is another attack event where attackers are trying to access the machine by using brute force attack. The frequency of attacks indicates that in any moment, attackers are trying to gain access to the system. Many different types of attacks are identified by analysing the log data. One such attack event was an attempt to 'got channel direct-tcpip request', which is used to create an SSH tunnel with the system. All these keywords that are identified during the honeypot data analysis are elements that could be very important for threat hunters for finding intelligence. This gives an important message that an attacker tries various techniques on honeypot unknowingly as they believe that this is a real system.

The brute force attack is one of the popular methods for attackers to a system. Attackers try different password combinations to get access to the system. In our Kippo honeypot data, we have identified a number of brute force attack. Attackers use different password combinations such as password, 123456, admin and other popular passwords. Table 2 provides a list of passwords that were used to attack the honeypot. Although the list is not complete it gives an idea of how the attackers are attacking the system. The most used password is 123456, which is about 37.35% compared to other

Table 2 Brute force attack

Password	No. of time occurred	% of occurring
123456	60,099	37.35
abc123	5086	3.16
admin	57,839	35.94
asdf	3848	2.39
asdf.*	20,310	12.62
password	8149	5.06
Password1	939	0.58
qwer	4619	2.87

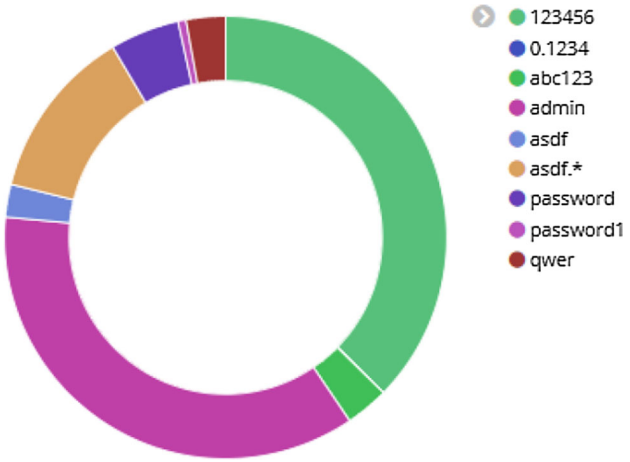

Fig. 8 Brute force attack on honeypot

elements on the list. These are the common passwords that were used by the attackers. Also, they seem to use a number of abusive languages to get into the system.

During each of the attack events, the attackers leave artefacts such as IP address, which is one of the important elements to understand attackers behaviour. Some of the IP address keep appearing in log data for several times. One of the reasons could be the same attacker was trying to breach the system security. However, the frequency of attack is very high, which is about 503 attacks/min, i.e. about 8.39 attacks/s. The attack frequency and duration are determined from the time stamp of the log file. On the other hand, some attackers (IP: 116.31.116.16) keep trying to log in for several minutes or hours and failed. For the above IP address, the attacker seems to have used various username and password combination. Some of the commons username and password are [root/p123456789], [root/p0o9i8], [root/parr0lla789], [root/pass!@#], [root/onlyidc2010] and [root/nhs39f40201]. PUTTY was used as a tool for SSH connection establishment. Figure 8 shows the frequency of the brute force attack by matching username and password.

Many attacks originated by the attackers start with a 'New connection'. They always use 'NEW KEYS' although they

```
31,10.0.2.15] NEW KEYS
31,10.0.2.15] starting service ssh-userauth
rauth on HoneyPotTransport,31,10.0.2.15] root trying auth none
rauth on HoneyPotTransport,31,10.0.2.15] root trying auth keyboard-interactive
rauth on HoneyPotTransport,31,10.0.2.15] login attempt [root/123456] succeeded
rauth on HoneyPotTransport,31,10.0.2.15] root authenticated with keyboard-interactive
rauth on HoneyPotTransport,31,10.0.2.15] starting service ssh-connection
nection on HoneyPotTransport,31,10.0.2.15] got channel session request
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] channel open
nection on HoneyPotTransport,31,10.0.2.15] got global no-more-sessions@openssh.com reques
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] pty request: xterm (
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] Terminal size: 24 80
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] request_env: '\x00\x

 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] getting shell
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] Opening TTY log: log

 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] /etc/motd resolved i
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] CMD: ls
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] Command found: ls
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] CMD: pwd
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] Command found: pwd
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] CMD: cd ~
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] Command found: cd ~
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] CMD: ls
 (0) on SSHService ssh-connection on HoneyPotTransport,31,10.0.2.15] Command found: ls
```

Fig. 9 A successful attack event

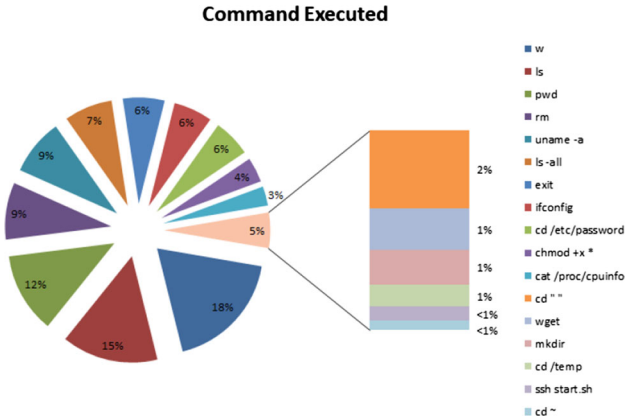

Fig. 10 Command executed by the attacker after logging to the system

do not have the public key. However, they managed to enter the authentication layer to start service ssh-userauth. At this stage, the honeypot logs the 'root trying auth none' and then 'root trying auth password'. So the attacker provides username and password such as 'root/joisber' for authentication. The honeypot detects 'login attempt' with the username and password did not match. Finally, the honeypot logs the cyber attack event as the 'root failed auth password' since the username was 'root'. Also, this is considered as 'unauthorised login' for the attacker's IP address.

In the event of a successful attack the scenario, the attacker uses correct username and password, which allows the attacker to enter the system. In this case, Kippo keeps the log as 'login attempt [root/123456] succeeded' followed by 'starting service ssh-connection' and 'got channel session request'. So, the session started and the attacker is in the system as a root user. Once the attacker is logged in to the system, he/she can do several activities such as access any files and folders of interest, implant a malware, delete a file and many more. Figure 9 is the part of the successful attack, which shows that after the attacker logged into the honeypot, he/she is using UNIX commands.

Moreover, it has been noticed that once the attacker managed to get into the system, they look at various information while leaving attack artefacts. Figure 10 illustrates a number of commands executed by the attacker in the honeypot. The command used by attackers varies depending on their needs. However, it is clear that they are using Unix commands. In this experiment, we have noticed that 'w' is used most of the time after the logging into the system, which means that the attackers want to know the user logged into that system on that given time. The next used command is 'ls', which is used to see the list of items in a directory. Other commands such as 'pwd', 'rm' and 'ssh' are used several times. The user of commands provides insights into the attacker's motive. However, some attackers user random commands, which means they do not have the planning of the attack.

Each of the attackers come in different ways using the different IP address are different. Some attackers keep changing their IP addresses and domain name. Some IP addresses were used only once that could be found afterwards. Some behaviour of the attackers is identified as they keep trying a different combination of username and password for a long time. The frequency of attack implies that these kinds of attack could be automated by using automated attack tools. This can determine the attack pattern and behaviour of an adversary, and their TTP identifies how they operate an attack.

Since the attackers attack the honeypot system assuming that they are attacking a real system, each of the attackers, who attack the kippo honeypot, uses a similar kind of network artefact. In most cases, they keep changing their IP addresses, which is very easy to change as we can see in the Pyramid of Pain in the literature review section. On the other hand, they always have some common characteristics, whether they are human or another machine.

For example, an attacker may always attack at a certain time of the day using a similar type of tools and techniques. They may use similar techniques each time they attack, such as user name as 'root' with a series of the password. It has also been noticed that the attacker does not change the IP address every time he or she attacks.

Figure 11 compares the login attempt failure with success. So, the unauthorised login success rate is above 8%, whereas the failure rate is about 12%. The 8% login success gives the idea of how the attackers are trying to get access to a system. This finding is beneficial for a threat intelligence point of view as it provides a perfect estimation of how attackers are trying to access information. Since this is a simple honeypot, which only presents a few services like operating systems, the attacker could not get any data. If attackers managed to access to a real system, they could control the whole system and possibly infect the entire network.

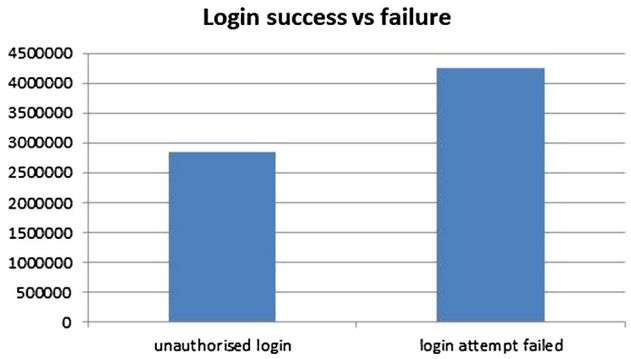

Fig. 11 Login success versus failure

We take the analysis even further to find the interest in networking port using Kibana and Elasticsearch. There are several networking ports that have been used to attack the honeypot system. In general, attackers use ports to attack a system by using port scanning. In this experiment, the most exciting port by the attackers was 'port 22', which uses secure shell protocol. To attack a system using port 22, the attacker needs to use the IP address, username and password. Figure 12 illustrates the percentage of attack using different ports. Port 22 has been used for about 48%, and telnet (port 23) has been used during 35% of the attack. Other ports such as MySQL Database System (port 5306) are used about 6% of the attack. However, it is interesting to see that there few attacks on Microsoft Active Directory (port 445), Microsoft Terminal Server (port 3389), Microsoft EPMAP (port 135) and MSSQL. Trying to attack through a Microsoft-related implies that the attackers do not have any idea about the host operating systems.

Finally, we have identified the geographic location of the attackers from the IP addresses they used. Figure 13 illustrates that attackers form the USA hit about 51% followed by China, which is approximately 40%. Other countries including Pakistan, Iran and Romania are in the list of originating attacks.

This experiment gives essential insights into an attack event including the way they attack. There were several instances of attack been identified in the honeypot. In most of the attacks, attackers have different ways to attack. Many SSH attacks have been identified, which used root as username and different combinations of passwords. During the attack events, the attackers leave network artefacts such as the same IP address. Also, one IP address appeared a different day and time. When the IP address appears many times, it indicates that the same attacker made the attack. Furthermore, these attackers always use similar tools and techniques. Finding and matching the attack event is a complex task as each of the attacks is related to many artefacts. To find those artefacts, elasticsearch played a significant role, which helps to identify various types of cyber attack events using full-text search. It has become apparent that attackers are continually targeting honeypots. Most of the attacks are similar to the attackers attempt to gain access to the system. This experiment into honeypot data for cyber intelligence is valuable as it can be used to identify and mitigate future cyber attacks.

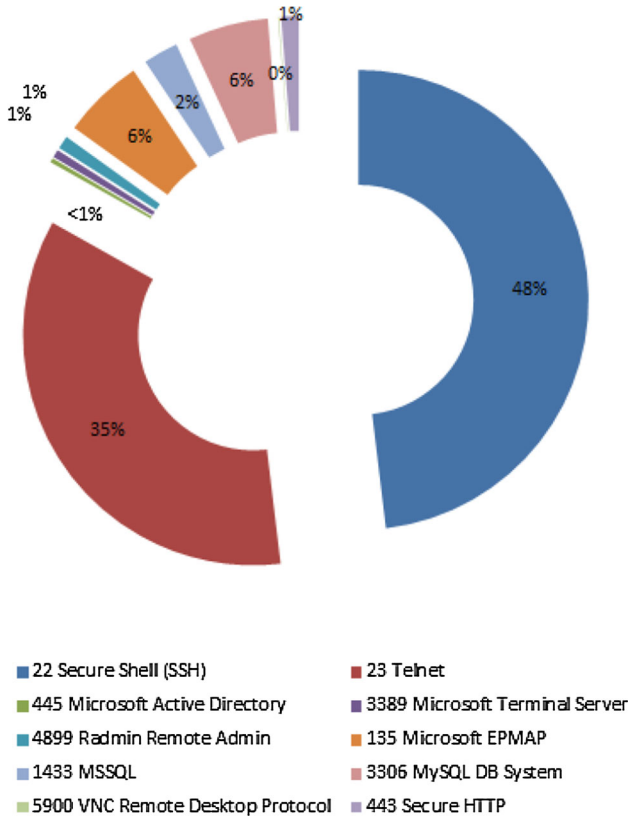

Fig. 12 Attack using port

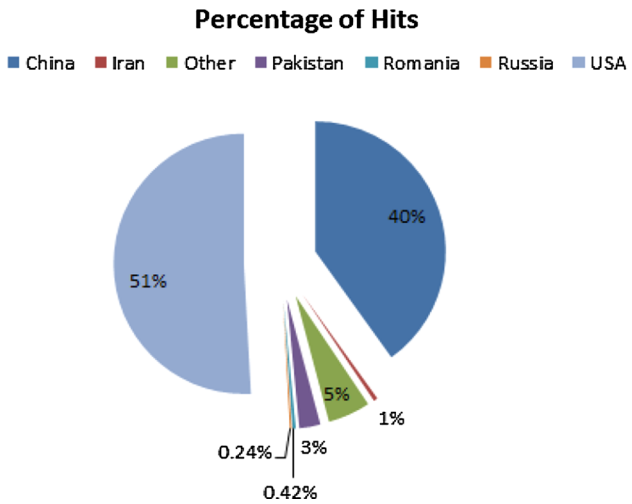

Fig. 13 Number of attacks from geographic locations

6 Conclusion

Threat intelligence is the knowledge that needs to be handled by using appropriate data collection and analysis. From the initial investigation, we have identified that cyber attacks to the honeypot system generate useful intelligence, which can be applied to the systems such as support IDS, IPS and firewalls to protect organisations' production. However, threat intelligence does not only help to detect attacks but to identify the way they attack by analysing the behaviour. Consequently, the threat intelligence techniques can help the protection systems to decide whether the 'new connection' request should be accepted, rejected, disrupted, degrade, deceive or destroy.

In future network defence, it is important to have intelligent driven mechanism. It is equally important in our understanding of a cyber attack to understand the behaviour of attackers. Identifying conduct can establish a pattern for an individual attacker. On the other hand, understanding the nature of repetition of an adversary by analysing the personal preference, convenience, use of tools give significant amount of indicators to prevent future attack. The model works only when there are a significant number of network attack-related data for analysis. The data are analysed using Elastic Stack for log data visualisation, which is highly scalable and flexible technology for analysing and visualising data. Honeypot data analysis for threat intelligence also provides cost-effective way to gather intelligence and then using production system.

In future work, we aim to extend the cyber attack model. One of the dimensions of this extension could be setting up honeypots to extract attack data. The cyber attack pattern and attacker behaviour can be extracted by using a machine learning algorithm. This can be used in the real system rather than the honeypot. So, in future, the threat intelligence model will be implemented in real-time log data which would collect, analyse and present in near-real-time and advanced system for action in cyber attack event. The implementation could be fully automated with minimum human interaction. Also, the visualisation could be used for live monitoring in the control centre, Network operations center (NOC), Security operation center (SOC) or other relevant departments. It will also generate alerts for the interested stakeholder and designated system depending on the business need for an organisation.

These attack patterns for real-time implementation could be used cyber threat hunting techniques for a better understanding of cyber attacks in the APT. Another dimension of this research could be to further development of a model for analysing APTs using honeypots data and attack modelling techniques such as Attack tree, Diamond Model and Kill Chain.

References

1. Al-Mohannadi H, Awan I, Al Hamar J, Al Hamar Y, Shah M, Musa A (2018) Understanding awareness of cyber security threat among it employees. In: 2018 6th international conference on future internet of things and cloud workshops (FiCloudW). IEEE, pp 188–192
2. Al-Mohannadi H, Mirza Q, Namanya A, Awan I, Cullen A, Disso J (2016) Cyber-attack modeling analysis techniques: an overview. In: 2016 IEEE 4th international conference on future internet of things and cloud workshops (FiCloudW), pp 69–76
3. Angiulli F, Argento L, Furfaro A, Parise A (2018) A hierarchical hybrid framework for modelling anomalous behaviours. Simul Model Pract Theory 82:103–115
4. BankofEngland (2016) Cbest intelligence-led testing: an introduction to cyber threat modelling. Bank of England Publication, London
5. Binaco D (2015) A framework for cyber threat hunting part 1: the pyramid of pain. http://blog.sqrrl.com/a-framework-for-threat-huntingpart-1-the-pyramid-of-pain. Accessed 5 Apr 2017
6. Brown S, Lam R, Prasad S, Ramasubramanian S, Slauson J (2012) Honeypots in the cloud. University of Wisconsin-Madison, Madison
7. Caltagirone S, Pendergast A, Betz C (2013) The diamond model of intrusion analysis. Technical report, DTIC Document
8. Chen P, Desmet L, Huygens C (2014) A study on advanced persistent threats. In: IFIP international conference on communications and multimedia security. Springer, Berlin, Heidelberg, pp 63–72
9. Dalziel H (2014) How to define and build an effective cyber threat intelligence capability. Syngress, Waltham
10. Gormley C, Tong Z (2015) Elasticsearch: the definitive guide: a distributed real-time search and analytics engine. ÓReilly Media, Inc., Sebastopol
11. Graham R (2000) What is a typical intrusion scenario? FAQ: network intrusion detection systems. https://linuxsecurity.com/resource-les/intrusiondetection/network-intrusion-detection.html. Accessed 20 June 2018
12. Grudziecki T, Jacewicz P, Juszczyk Ł, Kijewski P, Pawliński P (2012) Proactive detection of security incidents II – Honeypot. https://www.enisa.europa.eu/publications/proactive-detection-of-security-incidents-II-honeypots. Accessed 7 Jan 2019
13. Hilbert M (2015) Big data for development: a review of promises and challenges. Dev Policy Rev 34:135–174
14. Hutchins EM, Cloppert MJ, Amin RM (2011) Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead Issues Inf Warf Secur Res 1:80
15. Jasek R, Kolarik M, Vymola T (2013) Apt detection system using honeypots. In: Proceedings of the 13th international conference on

applied informatics and communications (AIC'13). WSEAS Press, pp 25–29
16. Kelly G, Gan D (2011) Analysis of attacks using a honeypot. In: International cybercrime, security and digital forensics conference
17. Liebergeld S, Lange M, Mulliner C (2013) Nomadic honeypots: a novel concept for smartphone honeypots. In: Proceedings of the Workshop on mobile security technologies (MoST'13), together with 34th IEEE symposium on security and privacy
18. Manadhata PK, Wing JM (2011) An attack surface metric. IEEE Trans Softw Eng 37(3):371–386
19. Mandt EJ (2017) Integrating cyber-intelligence analysis and active cyber-defence operations. J Inf Warf 16(1):31–48
20. Miloslavskaya N, Tolstoy A, Zapechnikov S (2016) Taxonomy for unsecure big data processing in security operations centers. In: 2016 IEEE 4th international conference on future internet of things and cloud workshops (FiCloudW). IEEE, pp 154–159
21. MITRE: a framework for cyber threat hunting (2016). https://medium.com/mitre-attack. Accessed 5 Jan 2019
22. Moore C, Al-Nemrat A (2015) An analysis of honeypot programs and the attack data collected. In: International conference on global security, safety, and sustainability. Springer, pp 228–238
23. Mulazzani M, Schrittwieser S, Leithner M, Huber M, Weippl ER (2011) Dark clouds on the horizon: using cloud storage as attack vector and online slack space. In: USENIX security symposium, San Francisco, CA, USA, pp 65–76
24. Ovelgönne M, Dumitras T, Prakash BA, Subrahmanian V, Wang B (2017) Understanding the relationship between human behavior and susceptibility to cyber attacks: a data-driven approach. ACM Trans Intell Syst Technol 8(4):51
25. Papazis K, Chilamkurti N (2019) Detecting indicators of deception in emulated monitoring systems. Serv Oriented Comput Appl 13:17–29
26. Phillips C, Swiler LP (1998) A graph-based system for network-vulnerability analysis. In: Proceedings of the 1998 workshop on new security paradigms, NSPW 98. ACM, New York, pp 71–79
27. Portokalidis G, Slowinska A, Bos H (2006) Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. In: ACM SIGOPS operating systems review, vol 40. ACM, pp 15–27
28. Schneier B (1999) Attack trees. Dr Dobb's J 24(12):21–29
29. Seifert C, Welch I, Komisarczuk P et al (2007) HoneyC: the low-interaction client honeypot. In: Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand
30. Shackleford D (2015) Who's using cyberthreat intelligence and how?. SANS Institute, Bethesda
31. Sochor T, Zuzcak M (2014) Study of internet threats and attack methods using honeypots and honeynets. Springer, Cham, pp 118–127
32. Sokol P, Pekarčík P, Bajtoš T (2015) Data collection and data analysis in honeypots and honeynets. In: Proceedings of the security and protection of information, University of Defence
33. SQRRL: a framework for cyber threat hunting (2016). http://sqrrl.com/media/Framework-for-Threat-Hunting-Whitepaper.pdf. Accessed 5 Apr 2017
34. United States. Joint Chiefs of Staff (2000) Joint tactics, techniques, and procedures for joint intelligence preparation of the battlespace. Joint Chiefs of Staff
35. van der Lelie-jop J, Breuk-rory R (2012) A visual analytic approach for analyzing SSH honeypots. http://ext.delaat.net/rp/2011-2012/p26/report.pdf. Accessed 10 Feb 2019
36. Weiler N (2002) Honeypots for distributed denial-of-service attacks. In: Eleventh IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises, 2002. WET ICE 2002. Proceedings. IEEE, pp 109–114
37. Xiaoli L, Pavol Z, Ron R, Dale L (2009) Threat modeling for CSRF attacks. In: 2013 IEEE 16th international conference on computational science and engineering, vol 3, pp 486–491