



The University of Bradford Institutional Repository

<http://bradscholars.brad.ac.uk>

This work is made available online in accordance with publisher policies. Please refer to the repository record for this item and our Policy Document available from the repository home page for further information.

To see the final version of this work please visit the publisher's website. Available access to the published online version may require a subscription.

Link to publisher's version: <http://voyager.ce.fit.ac.jp/conf/aina/2018/>

Citation: AL-Mohannad H, Awan I, Al Hamar J, Cullen A, Disso JP and Armitage L (2018) Cyber Threat Intelligence from Honeypot Data using Elasticsearch. 32nd IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA-2018) Pedagogical University of Cracow, Poland, May 16-18, 2018.

Copyright statement: © 2018IEEE. Reproduced in accordance with the publisher's self-archiving policy.
Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Cyber Threat Intelligence from Honeypot Data using Elasticsearch

Hamad AL-Mohannadi*, Irfan Awan*, Jassim Al Hamar[†], Andrea Cullen*,
Jules Pagan Disso[‡] and Lorna Armitage *

*School of Electrical Engineering and Computer Science
University of Bradford, United Kingdom, Bradford, BD7 1DP

Email: H.I.M.AL-Mohannadi@student.bradford.ac.uk, I.U.Awan@Bradford.ac.uk,
a.j.cullen@bradford.ac.uk, L.Armitage2@Bradford.ac.uk

[†]Ministry of Interior, State of Qatar, Doha
Email: j.alhamar@hotmail.com

[‡]Nettitude Limited, Leamington Spa, UK
Email: jpagnadisso@nettitude.com

Abstract—Cyber attacks are increasing in every aspect of daily life. There are a number of different technologies around to tackle cyber-attacks, such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), firewalls, switches, routers etc., which are active round the clock. These systems generate alerts and prevent cyber attacks. This is not a straightforward solution however, as IDSs generate a huge volume of alerts that may or may not be accurate: potentially resulting in a large number of false positives. In most cases therefore, these alerts are too many in number to handle. In addition, it is impossible to prevent cyber-attacks simply by using tools. Instead, it requires greater intelligence in order to fully understand an adversary's motive by analysing various types of Indicator of Compromise (IoC). Also, it is important for the IT employees to have enough knowledge to identify true positive attacks and act according to the incident response process.

In this paper, we have proposed a new threat intelligence technique which is evaluated by analysing honeypot log data to identify behaviour of attackers to find attack patterns. To achieve this goal, we have deployed a honeypot on an AWS cloud to collect cyber incident log data. The log data is analysed by using elasticsearch technology namely an ELK (Elasticsearch, Logstash and Kibana) stack.

I. INTRODUCTION

Cyber threat hunting is a complicated process for an organisation's network administrator or security personnel. The aim of threat hunting is to recognise cyber threats from alerts generated by IDSs in corporate networks to protect valuable assets. Understanding and mitigating cyber threats is a crucial and complex process. Honeypot data analysis is one of the ways to hunt for cyber threats. HoneyC [1], a low interaction client-based honeypot, which emulates only essential features of target clients. This is a client honeypot (HoneyC), which is able to detect client side attacks. This is a client honeypots. 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. SSH (Secure Shell) session honeypots are used for experimental purpose. Honeypot is one method of developing an understanding of any cyber-attack. More specifically, an SSH honeypot is analysed whilst the session is running and

the data is visualised using a visual analytical technique [2]. In addition, honeypots are used to mitigate Advanced Persistent Threat (APT). APT works with a combination of human and automated systems. The attacker in an APT does not jump into any attack without initially conducting reconnaissance and planning the attack. Jasek et. al [3] used honeypots to detect cyber-attacks. Honeypots are excellent resource as they give more resources to analysis for identification of cyber-attack than other technologies. Honeypot data analysis finds the anomalies to detect potential cyber-attack. Distributed Denial of Service Attack (DDoS) is a challenging threat for an organisation. Weiler [4] simulated the DDoS attack using honeypots to learn more about such cyber attack on network infrastructure. Honeypots are also able to emulate mobile devices to understand the threat on smart phone. The honeypots emulate real phone and collect data to understand what kind of malware infect smart phones. Such honeypots are called Nomadic [5], which provide infrastructure to collect threat intelligence data. The monitoring is also carried out by using visualisation techniques. A low interaction honeypot called Dionaea is used to collect attack data and analyse to understand the trend of cyber-attack [6]. They also build individual attacker profile by analysing collected data.

It has been noticed that threat hunting is done by many researchers by using honeypots data collection and analysis. On the other hand, honeypots produce huge amount of data. It is not easy for general-purpose data analysis tools to analyse such amount of data. In this paper we invoke elasticsearch technology to analyse honeypots data as it gives flexibility of searching on any size of data set. It is well known that honeypots and honeynets are unconventional security tools that allow security personnel to collect data and analyse them to learn more about cyber attack. In [7], authors collected data from honeypots to hunt cyber attacks patterns. Honeypot data is collected in by Moor et. al [8], which captured IP address of attacker for further analysis.

There are a number of IDSs and IPSs available on the market to protect networks, hosts and applications that form

part of an organisations information and technology assets. These tools can be automated for finding and reducing threats. However, automating cyber security tools are not the complete solution for protecting valuable asset within an organisation. The automation requires analysis of activities of the intruders to provide better protection.

In this paper, we have proposed a new threat intelligence approach. The threat intelligence technique is evaluated following collection of data. This is achieved by deploying honeypots to find cyber-attack events through the analysis of this data using elasticsearch technique. The results are promising, thus demonstrating that honeypot data analysis could be used in cyber threat intelligence instead of using more traditional production systems.

In section II of this paper, we discuss a wide range of related work for better understanding of existing cyber-threat hunting techniques. In section III, we analyse cyber threat hunting and propose a new threat intelligence model. We also provide an initial conceptualisation to describe this formally. In section IV, we setup and experiment using an ELK stack and discuss the outcome. Finally, we summarise the paper and provide future directions for this research.

II. RELATED WORK

Corporate networks are equipped with several security devices such as traditional firewalls, IDS, IPS, anti-malware software, traffic sniffers etc., to protect valuable assets. Most of these devices are rule-base detection systems that allow or reject traffic according to the rule-sets. On the other hand, cyber security is a process not a product, which needs continued monitoring and improvement. Therefore, it is important to think of advanced cyber threat handling in a more analytical fashion. Hunting potential threats is a more advanced approach than the traditional rule-base detection system [9].

Pursuing cyber threats is not a new concept. The threats are usually modelled using various modelling techniques. This acts as a support in attack situations and enables any prevention strategy to consider a number of scenarios. There are many cyber-attack modelling techniques used to analyse cyber-attack such as: Attack Graphs or Trees [10] [11], Attack Vectors [12], Attack Surface [13], Diamond model [14], the OWASP's threat model [15] and the Kill Chain[16], [17]. These modelling techniques can be used individually or in conjunction with other models. In [18] a number of cyber attack modelling techniques developed to handle cyber attacks efficiently were discussed. Cyber attack modelling is mainly concerned with identifying the attack patterns of the adversary. On the other hand, cyber threat hunting is a process of monitoring, data collection and analysis of event data to find anomalies. It also deals with the visualisation techniques, linked data analysis and model building [19].

A. Pyramid of Pain

The Pyramid of Pain (PoP) was introduced by Binaco [9], which analyses how an IoC behaves. The idea of an IoC is that it identifies a comprise of some network-related components

that usually are used to perform cyber attacks. The main idea of the PoP is to establish the different levels of IoC for cyber defence. From the bottom up, the pyramid indicates the level of difficulty for handling cyber threats. So, the IoC defines the components of the PoP.

Figure 1 shows the PoP, which gives more levels of technical difficulty for both the adversary and the victim. PoP provides a simplified view of the adversarys activities on the system. An adversary uses the PoP components for developing an attack on a network. In addition, in a cyber-attack, an adversary generally leaves some form of footprint which could be the combination of the PoP components. So, analysing a PoP could reveal the nature and motive of an adversary, which could be used to take informed decisions for threat intelligence. The trivial metric of the pyramid is at the bottom called the hash value. Hash values provide unique reference for specific malware or to the payload that is used for the attack. Hash values can be changed, for example, a minor change to the payload changes the hash. So, it is not worth keeping track of them as new values are more often continually generated. This means that attacks using hash values are easily identified and tackled, so, the possibility of a system compromise is very low. In any cyber-attack incident, IP addresses are very basic indicators for identifying an attacker. It is hard to hide IP addresses during a cyber-attack event. For an attacker, it is very easy to change the IP address after and attack or masquerade before an attack takes place. In practice it is not feasible to pursue every single IP address that has tried to breach a system.

To get a domain name however, the adversary must have registered with a hosting company. It is relatively easy to trace back to the origin of the domain; although attackers could be disguised. On the other hand, domain names can be changed at anytime. Since domain name suers have to register, it is more difficult to change domain names in comparison to IP addresses. The next indicator is the network artefact, which could differentiate the malicious activities of the adversary from that of legitimate users. Hosts that are involved in a cyber attack, often contain a great deal of information about the attack. Host artefacts are the indicators of malicious activities performed within the host. These can be used to distinguish the activities of the legitimate user and the adversary. One of the difficulties in terms of IoC is the tools used by the adversary to make an attack. These tools which are used to deploy or plant the payload can be software or hardware based in in effect a combination of both new or customised tools can be a great challenge for the analyst. Therefore, tools could be a difficult IoC. The final and top component of the pyramid is Tactics, Techniques and Procedure (TTPs). This is the level where the behaviour of the adversary can be identified from the malicious software or the payload.

B. Hunting Maturity Model (HMM)

This is a cyber-threat hunting model, that identifies an organisations threat hunting ability including quantity and quality of threat data collection. HMM also indicates way

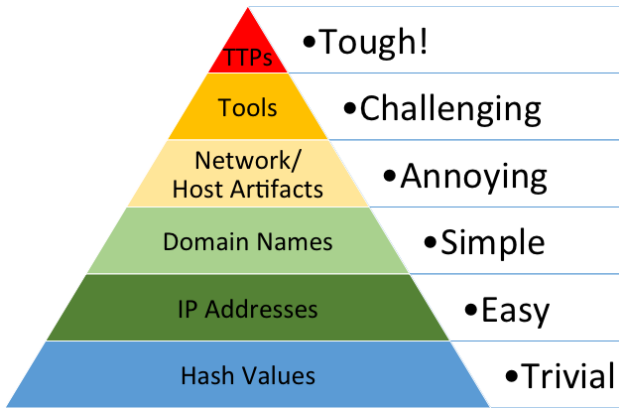


Fig. 1. Pyramid of Pain (Extracted from [9])

TABLE I
HUNTING MATURITY MODEL

Level	Maturity	Comment
Level 0	Initial	Depend on automated alerting
Level 1	minimal	Incorporate threat intelligence
Level 2	Procedural	Follow data analysis produced by others
Level 3	Innovative	Create new data analysis technique
Level 4	Leading	Automate the successful data analysis procedure

of analysing and visualising data [19]. This hunting model consists of five levels of maturity. The first level, which is level 0, is where organisations mainly rely on third-party automated alerting systems. In this level, very little or no data is collected. The maturity levels go up depending on how organisations collect data, analyse data and incorporate them into cyber threat analysis. In this context, the highest level means that the organisation uses very high levels of routine data collection and automated systems for data analysis. Figure 2 the HMM, which is linear in nature. The main idea behind the HMM is that it requires continuous improvement.

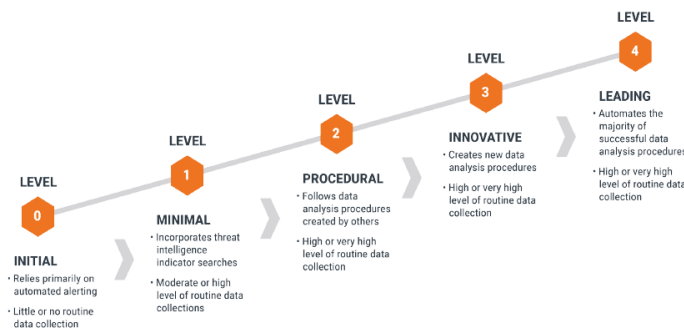


Fig. 2. Hunting Maturity Model (Adapted from [19])

Threat hunting is not a one off action, it is a process. It depends on many criteria such as the creation of a hypothesis, investigation of tools and techniques, identification of new patterns and enriched analytics. The Sqrrl Data [19] introduced the hunting loop as shown in figure 3. The loop components can be matched with the hunting maturity model to identify

the strength of the company's data collection analysis. If the process works for a hunting threat, it could be automated and shared with other team members for tackling similar types of cyber-threats.

The HMM process is conducted in the following steps:

- Data Collection - To hunt real threats within the network or host collecting data is the most important task. Data could be collected from different sources. These data could be different types such as syslog, honeypots data, firewall data, server logs etc., which could be used for creating a hypothesis. In most cases data collection could be automated, which could be fed into an analytic system or into the visualisation software.
- Hypothesis Creation - If existing alert based systems such as IDS, IPS, SIEM (Security Incident Event Management) or Firewalls do not find a real threat, it is important to review and analyse historical data to develop a new hypothesis. The hypothesis needs to be reviewed frequently within a typical network. Once a genuine threat is identified, the hypothesis needs to be reviewed and improved. In addition, a new hypothesis may be introduced depending on the cyber incident event.
- Tools and Techniques for Hypothesis Techniques - From the data collection to automation, there are many tools and techniques required to hunt a cyber threat. Basic log analysis tools or SIEM gives a minimal level of flexibility for mature hunting. A hypothesis must be tested against the tools and techniques used for threat hunting. In most cases advanced levels of visualisation should aid to test and create a new hypothesis.
- Pattern & TTP Detection - APT [3] or Zero Day Attacks [20] are difficult to identify or predict in advance. This is especially the case for a Zero Day attack as this does not match any known attack pattern. It is important to create patterns for identifying typical attacks and to keep looking for new and emerging threat patterns.
- Analytic Automation - Cyber threat hunting involves a number of tools and techniques. It is almost impossible to manage all these tools manually. Automation plays a key factor in such situations. The threat hunting process from data collection to detection needs instead to be automated for efficiently managing cyber threat incident events.

C. Matrix of IoC

The IoC can be put in a form of a matrix. Each of the indicators is evaluated using three criteria such as trace, identify and throttle. In the following we discuss three criteria against IoC for better understanding of those indicators.

- Trace - It is important to trace an attacker during their visit to a network or a host. Tracing a hash value is not

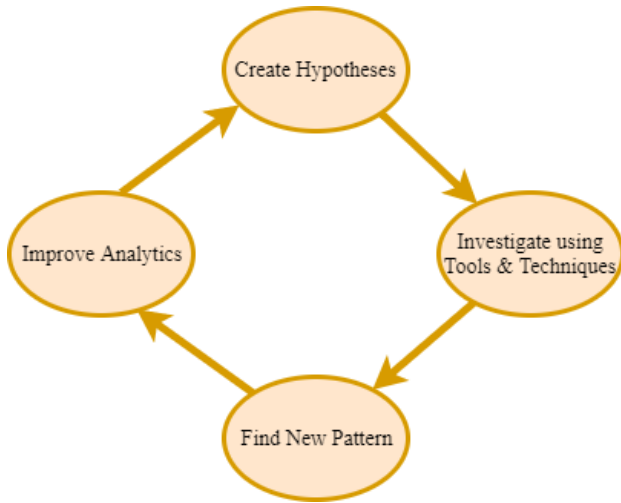


Fig. 3. Threat Hunting Loop (Extracted from [19])

entirely beneficial as values could be changed by the attacker 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 any subsequent attack is performed by the same attacker. An IP address is the key in making a connection between devices as each of the devices within the network must have an IP address. The attacker may change the IP address every time they do an attack, which is also true for domain names. On the other hand, the attacker may leave a network or host artefact although they have changed IP address or domain name. So, these networks or host artefacts are important elements to investigate further. Some users utilise same tool repeatedly to conduct a cyber attack, as changing the attack tool may require development and testing, which could be expensive. So, tools could be used to identify attackers whether they are using the same tool or not. In the top of the pyramid, the TTP is the most important and difficult indicator. Since TTP mainly expresses the skills and training of the attackers, they may improve their skill-set over time, which makes threat hunters thinking hard about attackers' next actions .

- Identify - Tracing helps threat hunters to identify the foot-print of an attacker. Any evidence left by the attacker could be used to identify the attacker in a future attempt. For example, tracking systems can be used to match identified components such as IP address, hash values and domain name. Identifying the network or host artefact could help in analysing attack behaviour.
- Response - If a threat is traced and identified, it is required to prevent future events from happening. For example, if an IP address is identified as a threat element from data analysis, it could be black-listed for any future events. If quick response is made to the identified

TABLE II
PYRAMID OF PAIN IN PRACTICE

Criteria	Trace	Identify	Response
Hash Value	Easy	Easy	Easy
IP Address	Easy	Easy	Easy
Domain Names	Easy	Easy	Easy
Network Artefacts	Medium	Medium	Medium
Host Artefacts	Medium	Medium	Medium
Tools	Hard	Hard	Hard
TTP	Hard	Hard	Hard

element, the defence becomes offence [9].

III. THREAT HUNTING ANALYSIS

Cyber threat analysis is the key to threat hunting. Cyber threat hunting is a process of searching potential cyber threats through the network by analysing relevant data-sets. Data analysis could be performed by using existing automated tools or alternatively be performed manually. In an organisation, cyber threat hunting maturity depends on the ability of data collection and analysis [19]. Data could be historical or live, depending on the most valuable source to identify cyber threats. Threat data could be collected by using honeypots and analysed to understand threats before they occur [21]. Data also contains details of a cyber security incident that has happened. Analysing such data gives an indication that most of security incidents do not occur as zero-day attacks [20], they are quite frequent and in most cases have patterns. Appropriate data collection and analysis could lead to many elements of IoC.

To maximise hunting, we have installed two low-interaction honeypots called Kippo and Dionaea [22] on Amazon cloud services. We have collected over 500MB of Kippo log data. The log data consists of all the login attempts onto the honeypots. The log also comes with timestamps that indicate when the event took place.

Before we analyse the honeypots log data, we propose a cyber-threat intelligence model, which will help us to understand the collected data efficiently. In the following subsection, we have defined the conceptual model and derived a formal definition of that proposed model.

A. Threat Intelligence

The following is the conceptual diagram of cyber-attack recognition. There are three main components:

- Attack
- Behaviour
- Pattern

An attack is a systematic approach by an attacker to gain access in to a system, a network or a host. An attack is originated from anattacker on to a system, which can be recorded using data collection. The behaviour of the attacker can be identified from the data collected if the same attacker attempted several attacks. In any event, an attacker is a human or a machine. For both cases the behaviour is an indicator of

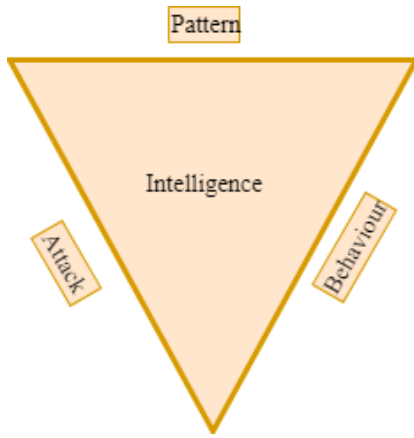


Fig. 4. Cyber-attack Concept

the method used. So, there is a good relationship between human behaviour and cyber attacks [23].

Generally, system log data collection is performed for most of the system. These data can be considered as big data as the data has velocity, verity and volume [24]. If we consider 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 occur repeatedly over time, which could form pattern. The aim of using data analysis is to identify such a pattern. Data can be analysed more intelligently and efficiently by using big data analysis techniques.

The main idea of the triangle in Figure 4 is that a cyber incident data-set contains attack data, which can be analysed using data analysis. Attack data can be separated from normal incidents and presented in a more readable format.

So, in this context the attack performed by the attackers reveals the behaviour of the attackers. By adding intelligence such as data analysis, to these two components we can identify attack patterns. Attack patterns could be the key to preventing future cyber-attacks.

So, in this context the attack performed by the attackers reveal the behaviour of the attackers. By adding intelligence such as data analysis, to these two components we can identify attack pattern. Attack pattern could be the key to prevent future cyber-attack.

B. Problem Analysis

An initial conceptualisation of the cyber-attack is describe as follows.

A cyber-attack is originated by an adversary from a remote location. An attack will have one of two outcomes: a) successful, which means the victims system was compromised or, b) unsuccessful, which means the victims system was not compromised.

An attack can be defined as a set of actions $\{a_1, a_2, a_3, \dots, a_n\}$ taken by an adversary by using some tools and techniques to access valuable assets. 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 stands for nodes and edges E for a path. In the event of an unsuccessful attack the path remains a single direction. On the other hand, if the attack is successful, the system is compromised and, the path becomes bi-directional. The Internet consists a heterogenous topology, however we are only interested in the abstract edges and vertices. Our main interest in the vertices are that they identify the attackers and victims machines. In a victims machine or network, the data, which we will call *assets* could be in three different stages.

Assets $X = \{X_r, X_p, X_m\}$, 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. We define a *node* function which for an *asset* and time-stamp returns the *node*. The *node* is represented by the symbol \perp , which implies that an *asset* is on the move. Formally, $node : X \times T \rightarrow V \cup \{\perp\}$. So, for the resting *asset* $x \in X_r$, where $node(x, t)$ is constant, i.e., the value does not depend on the time-stamp. On the other hand, X_p and X_m are dependent on the time-stamp.

Let assume that in the event of a cyber-incident, an attack starts at time t and lasts for Δt . Given the time-stamp, we have formalised cyber-incident as follows -

attack - an attacker comes to the contact of the victim's system at time t and leaves at time δt . The elapse time may vary depending 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.

IV. EXPERIMENT SETUP

This section describes the experiment setup using the ELK¹ stack. The ELK stack consists of Elasticsearch, Logstash and Kibana, which helps to present data, create visualisation and a dashboard for any size of data. One of the advantages of using elasticsearch technology is that scalability is not issue. ELK can handle any size of data and search is faster. To support the ELK we used Filebeat to get multiple files to the elasticsearch. Figure 5 illustrates the architecture of the experiment. We have collected more than 500MB of honeypot log data for more than one year through an Amazon Web Services (AWS) cloud. We setup two honeypots called Kippo and Dionea, where kippo is a low-high and Dionea is medium honeypot. Both of the honeypots appear as real operating systems, which attracts many attackers. These log data contains a time-stamp and a date of any events. Events are recorded if anyone tries to interact with the honeypot. The log data is huge in size, which is very difficult to analyse simply by looking at the log files. So, we adapted the ELK in order to find the meaning of the log data. The main advantage of ELK 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

¹<https://www.elastic.co/>

relevant 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. Elasticsearch is used by many organisations such as: Wikipedia 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 a meaning within data [25].

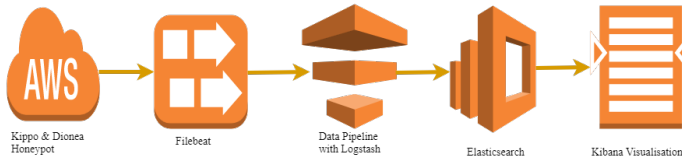


Fig. 5. Experiment Setup with ELK Stack

A. Data Analysis using Kibana

In Kibana, we performed a number of keyword searches using elasticsearch. The main goal was to find attack events in our honeypots. We have identified a number of events from the log data analysis. The Figure 6 illustrates the attack events in the Kippo honeypot. We identified eight keywords, which can be recognised as events that occurred in the honeypots. It should be noted that all the events are not attack-related. It has been recognised that six of the keywords are attack-related and the rest such as remote error and connection lost, are not related to any cyber attack. In Figure 7 attack data is

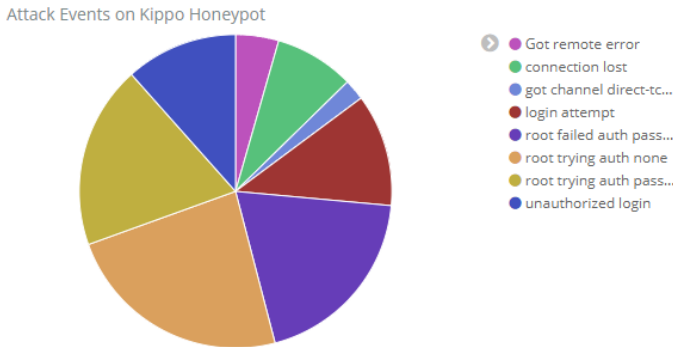


Fig. 6. Kippo Honeygot Log Event Visualisation using Kibana

summarised according to the keywords found in the log data. It shows that each of the attacks happen within the honeypot. We know that attackers of the honeypot do not have any idea that they are interacting with a honeypot system. This can indicate the frequency of an attack in relationship to a legitimate system.

The result has been summarised in table III to identify the statistics of those events that occurred. We identified that root trying auth none occurred some 3,839,723 times which, is about 23.57% of the total number of events found up to this point of data collection. Since the honeypots are Linux

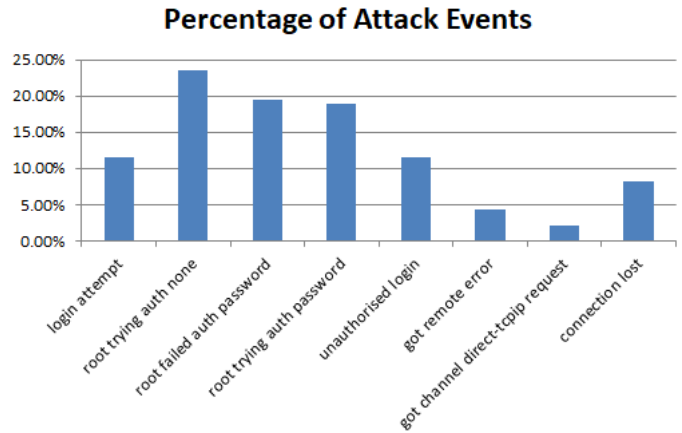


Fig. 7. Attack Events

TABLE III
ATTACK EVENTS ANALYSIS

Event Name	No of time occurred	% of occurring
login attempt	1, 889, 046	11.6%
root trying auth none	3, 839, 723	23.57%
root failed auth password	3, 172, 791	19.48%
root trying auth password	3, 172, 791	18.91%
unauthorised login	1, 889, 046	11.6%
got remote error	726, 436	4.46%
got channel direct-tcpip request	351, 466	2.16%
connection lost	1, 342, 279	8.24%

machines, the attackers try to access root. The second event is the root failed auth password, which occurred 3,172,791 times; or a total of 19.48%. This is another attack event where attackers are trying to access the machine by using some sort of brute force attack. The frequency of attacks indicate that in any moment of time, 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.

V. CONCLUSIONS

In this paper we have introduced a new threat intelligence model, which indicates that attack, behaviour and patterns are a relevant and important concern to all organisations. It is equally important in our understanding of a cyber attack to understand the behaviour of attackers. Consequently, attack pattern can be identified from the attack and behaviour. The model works only when there are a significant number of network incident related data for analysis. We have analysed cyber threat intelligence by using honeypot data collected from

AWS. The data is analysed using an ELK stack for log data visualisation. It is worth noting that ELK uses elasticsearch, which helps to identify various types of cyber incident events. It has become apparent that honeypots are constantly being targeted by attackers. Most of the attacks are similar in kind as 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. The main advantage of using honeypot data for threat intelligence is that there is no side effect on the production system. This kind of analysis could help to build future IDS and IPS for production.

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. These attack data could be analysed by using appropriate tools to find attack patterns. The resulting patterns could be used to train IDS and IPS systems to automate future processes. These attack patterns could be used to implement cyber threat hunting techniques for a better understanding of cyber attacks. APTs are one of the issues in a cyber security environment [26]. In this case, a group of attackers use full planning advanced infrastructure and capability to attack a corporate network. This is a growing and challenging issue for businesses and governments alike. Another dimension of this research could be to further develop the model for analysing APTs using honeypots data and attack modelling techniques.

REFERENCES

- [1] C. Seifert, I. Welch, P. Komisarczuk *et al.*, "Honeyc-the low-interaction client honeypot," *Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand*, 2007.
- [2] J. van der Lelie-jop and R. Breuk-rory, "A visual analytic approach for analyzing ssh honeypots."
- [3] R. Jasek, M. Kolarik, and T. Vymola, "Apt detection system using honeypots," in *Proceedings of the 13th International Conference on Applied Informatics and Communications (AIC'13)*, WSEAS Press, 2013, pp. 25–29.
- [4] N. Weiler, "Honeypots for distributed denial-of-service attacks," in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on*. IEEE, 2002, pp. 109–114.
- [5] S. Liebergeld, M. Lange, and C. Mulliner, "Nomadic honeypots: A novel concept for smartphone honeypots," in *Proc. Wshop on Mobile Security Technologies (MoST13), together with 34th IEEE Symp. on Security and Privacy*, 2013.
- [6] G. Kelly and D. Gan, "Analysis of attacks using a honeypot," in *International Cybercrime, Security and Digital Forensics Conference*, 2011.
- [7] P. Sokol, P. Pekarčík, and T. Bajtoš, "Data collection and data analysis in honeypots and honeynets," *Proceedings of the Security and Protection of Information. University of Defence*, 2015.
- [8] C. Moore and A. Al-Nemrat, "An analysis of honeypot programs and the attack data collected," in *International Conference on Global Security, Safety, and Sustainability*. Springer, 2015, pp. 228–238.
- [9] D. Binaco, "A framework for cyber threat hunting part 1: The pyramid of pain," 2015. [Online]. Available: <http://blog.sqrrl.com/a-framework-for-threat-hunting-part-1-the-pyramid-of-pain>
- [10] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 Workshop on New Security Paradigms*, ser. NSPW '98. New York, NY, USA: ACM, 1998, pp. 71–79. [Online]. Available: <http://doi.acm.org/10.1145/310889.310919>
- [11] B. Schneier, "Attack trees," *Dr. Dobbs journal*, vol. 24, no. 12, pp. 21–29, 1999.
- [12] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. R. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," in *USENIX Security Symposium*. San Francisco, CA, USA, 2011, pp. 65–76.
- [13] P. K. Manadhata and J. M. Wing, "An attack surface metric," *Software Engineering, IEEE Transactions on*, vol. 37, no. 3, pp. 371–386, 2011.
- [14] S. Caltagirone, A. Pendergast, and C. Betz, "The diamond model of intrusion analysis," DTIC Document, Tech. Rep., 2013.
- [15] X. Lin, P. Zavorsky, R. Ruhl, and D. Lindskog, "Threat modeling for csrf attacks," *2013 IEEE 16th International Conference on Computational Science and Engineering*, vol. 3, pp. 486–491, 2009.
- [16] U. S. J. C. of Staff, *Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Battlespace*. Joint Chiefs of Staff, 2000.
- [17] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, p. 80, 2011.
- [18] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, and J. Disso, "Cyber-attack modeling analysis techniques: An overview," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug 2016, pp. 69–76.
- [19] SQRRL, "A framework for cyber threat hunting," 2016. [Online]. Available: <http://sqrrl.com/media/Framework-for-Threat-Hunting-Whitepaper.pdf>
- [20] G. Portokalidis, A. Slowinska, and H. Bos, "Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation," in *ACM SIGOPS Operating Systems Review*, vol. 40, no. 4. ACM, 2006, pp. 15–27.
- [21] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation," in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. ACM, 2011, pp. 29–36.
- [22] T. Sochor and M. Zuzcak, *Study of Internet Threats and Attack Methods Using Honeypots and Honeynets*. Cham: Springer International Publishing, 2014, pp. 118–127.
- [23] M. Ovelgönne, T. Dumitras, B. A. Prakash, V. Subrahmanian, and B. Wang, "Understanding the relationship between human behavior and susceptibility to cyber attacks: A data-driven approach," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 4, p. 51, 2017.
- [24] M. Hilbert, "Big data for development: A review of promises and challenges. development policy review," *martinhilbert. net*. Retrieved, pp. 10–07, 2015.
- [25] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. " O'Reilly Media, Inc.", 2015.
- [26] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security*. Springer, 2014, pp. 63–72.