Cyber Attack Modelling using Threat Intelligence

An investigation into the use of threat intelligence to model cyber-attacks based on elasticsearch and honeypot data analysis

Hamad Al-Mohannadi

School of Electrical Engineering and Computer Science

University of Bradford

This dissertation is submitted for the degree of

Doctor of Philosophy

August 2019

Abstract

Hamad Al-Mohannadi "Cyber Attack Modelling using Threat Intelligence"

Keywords: Cyber-attack, Cyber-attack modelling, Cyber threat intelligence, Elasticsearch, Honeypots, Cloud Services, Attack awareness, Object-based model

Cyber-attacks have become an increasing threat to organisations as well as the wider public. This has led to greatly negative impacts on the economy at large and on the everyday lives of people. Every successful cyber attack on targeted devices and networks highlights the weaknesses within the defense mechanisms responsible for securing them. Gaining a thorough understanding of cyber threats beforehand is therefore essential to prevent potential attacks in the future. Numerous efforts have been made to avoid cyber-attacks and protect the valuable assets of an organisation. However, the most recent cyber-attacks have exhibited the profound levels of sophistication and intelligence of the attacker, and have shown conventional attack detection mechanisms to fail in several attack situations. Several researchers have highlighted this issue previously, along with the challenges faced by alternative solutions. There is clearly an unprecedented need for a solution that takes a proactive approach to understanding potential cyber threats in real-time situations.

This thesis proposes a progressive and multi-aspect solution comprising of cyber-attack modeling for the purpose of cyber threat intelligence. The proposed model emphasises on approaches from organisations to understand and predict future cyber-attacks by collecting and analysing network events to identify attacker activity. This could then be used to understand the nature of an attack to build a threat intelligence framework. However, collecting and analysing live data from a production system can be challenging and even dangerous as it may lead the system to be more vulnerable. The solution detailed in this thesis deployed cloud-based honeypot technology, which is well-known for mimicking the real system while collecting actual data, to see network activity and help avoid potential attacks in near real-time.

In this thesis, we have suggested a new threat intelligence technique by analysing attack data collected using cloud-based web services in order to identify attack artefacts and support active threat intelligence. This model was evaluated through experiments specifically designed using elastic stack technologies. The experiments were designed to assess the identification and prediction capability of the threat intelligence system for several different attack cases. The proposed cyber threat intelligence and modeling systems showed significant potential to detect future cyber-attacks in real-time.

Declaration

I, Hamad Al-Mohannadi confirm that this thesis contains my own work and has never been submitted for any other academic award. Any information derived from other material has been properly referenced.

Hamad Al-Mohannadi

August 2019

To my loving family and parents.

Acknowledgements

First and foremost, I would like to thank Almighty Allah for blessing me with immense patience, strength, and knowledge to enable me to finish this study.

I would like to thank His Highness Sheikh Tamim Bin Hamad Al Thani for giving me an opportunity to study abroad with a scholarship and giving the young Qataris a brighter hope for the future. I am also indebted to the Qatar Embassy in London for their support during my study.

I am incredibly grateful to my supervisor Prof. Irfan Awan for his professional support and belief that gave me the confidence to continue my research in the right direction. I appreciate all his contributions of time and ideas to make my PhD experience productive and stimulating. It has been an honour to be his PhD student.

This research was carried out with the scholarship provided by the Government of Qatar. I would like express my appreciation to the Qatar Government and all the staff related to this project directly or indirectly for giving me this scholarship, opportunity and approval for conference travel grants.

A big thanks to Dr Andrea Cullen and Dr Jules Pagna Disso for their expert technical insights in cybersecurity and critical analysis of my work during the entire period of this study.

My time at Bradford was made enjoyable in large part due to the friends and colleagues who became a part of my life. I am grateful for the time spent with my colleagues in the research lab discussing ideas and having coffee.

Last but not least, I would like to thank my parents for their unconditional love and prayers throughout this journey that allowed me to reach this level successfully. I am grateful to my wife and children for making this journey much easier for me and inspire me to set my goals high.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACL – Access Control Lists

AMI – Amazon Machine Images

APIs – Application Programming Interfaces

APT – Advance Persistent Threat

AWS – Amazon Web Service

BSIMM – Building Security in Maturity Model

DDoS – Distributed Denial of Service Attack

DMZ – Demilitarised Zone

ELK – Elasticsearch, Logstash and Kibana

FERPA – Family Education Rights and Privacy Act

GDPR – General Data Protection Regulation

GLBA – Gramm-Leach-Bliley Act

GPS – Global Positioning System

HIPAA – Health Information Probability Act

HMM – Hunting Maturity Model

IaaS – Infrastructure-as-a-Service

IDS – Intrusion Detection System

IoC – Indicator of Compromise

IP – Internet Protocol

IPS – Intrusion Prevention System

ISO – Information Security Officer

MDM – Mobile Device Management

MULVAL – Multihost, Multistage, Vulnerability Analysis

NETSPA – A Network Security Planning Architecture

NOC – Network Operation Centre

OWASP – Open Web Application Security Project

PCI – Payment Card Industry

PE – Portable Executable

PoP – Pyramid of Pain

RoI – Return on Investment

SIEM – Security Incident Event Management

SOC – Security Operation Centre

SSH – Secure Shell

TCP – Transmission Control Protocol

TTPs – Tactics, Techniques, and Procedures

TVA – Topological Analysis of Network Attack Vulnerability

VPN – Virtual Private Network

Chapter 1

Introduction

1.1 Motivation

The Internet has become a significant part of contemporary society, it is utilised widely whether we are at home or across the world. The over reliance on the Internet has meant that being connected to a wireless network has become an essential part of modern life. From connecting loved ones, to Global Positioning System (GPS) mapping and routing from almost everywhere on the earth, to multi-million-pound transactions, and advanced medical technology the Internet has many uses. However, there are also many pitfalls associated with the advancement of the Internet, particularly the compromising of privacy, identity, data infringements, and malware attacks on devices.

With increased Internet usage the situation in the cyber world is becoming much more complex, with the security of legitimate networks being under threat daily. There are a significant number of researchers working on cyber threat analysis to predict the model of attack for any given system. Some of these cyber-attacks have been defined as cyberwar [1] [2] and provide some initial guidelines of a future defence of cyberspace. The defence mechanism is mainly concerned with the understanding of the network, nature of the attackers, the motive of the attacker, method of attack, security weakness of the network etc., to mitigate any future attack. Furthermore, it is also essential to model attacks earlier to ensure the network is secure.

In order to protect a network, it is essential to understand the behaviour of the malware that is found in an infected device. The primary purpose of malware analysis is to respond to an intrusion on the network [3] [4]. The analysis can be done statically or dynamically, which can help cybersecurity teams to understand the weaknesses of the network [5]. An approach such as honeypots can help in preventing networks from sustaining an intrusion, as they can help in acquiring attack data by mimicking the real network [6]. Honeypots data can also be classified in terms of network traffic for a better understanding of the intrusion type. Honeypots data analysis provides more information on the attack and the motive of the attacker. This also helps to build real-time intrusion detection systems [7]. In order to detect an intrusion, it is essential to learn about the weakness of the network. It is also necessary for the cybersecurity team to understand the motive of the attacker e.g. "what data could be targeted?" and "why the attack happened?" [8] [9]. The attack modelling and analysis has been emphasised in a recent article by the Bank of England, which describes how threats can be modelled to mitigate the cyber-attack in an organisation [10].

An important aspect of threat hunting is Cyber threat analysis. Hunting maturity depends on the ability of data collection and how this data is analysed [11]. The most valuable source that can be used in order to identify a cyber threat is data that could be historical or live. Threat data is collected using honeypots. These are then analysed to understand threats before they occur [12]. Furthermore, the data also contains the cybersecurity incident that has occurred, analysing such data indicates that security incidents do not just happen as zero-day attacks [13], in fact they can be quite frequent and have a certain pattern. In depth data collection and analysis could lead to many elements of Indicator of Compromise (IoC). Identifying a cyber threat before it occurs is a complicated process for network administrators or security personnel, furthermore it is quite challenging if it is a production system. Therefore, it is essential to find a system that could act as a real system to the attacker and is able to collect valuable information about attack events. Honeypot data analysis is one of the ways to hunt for cyber threats, as it is a method of developing an understanding of any cyber-attack. More specifically, a Secure Shell (SSH) honeypot is analysed while the session is running and the data is visualised using a visual analytical technique [14].

Honeypots are able to produce a considerable amount of log data, which records each event that has occurred with the time-stamp. A Log analysis using Elastic Stack is used to enhance threat intelligence, which then helps in increasing the network security by making appropriate measures. It is not easy for general-purpose data analysis tools to analyse such large amounts of data. Honeypots and honeynets are popular unconventional cloud-based security services that allow security personnel to collect data and analyse them to learn more about cyber-attack Sokol et al., [15] collected data from honeypots to hunt cyber-attack patterns. Honeypot data collected by Moor et al., [16], captured the IP addresses of attackers for further analysis. There are a number of Intrusion Detection Systems (IDSs), Intrusion Prevention Systems (IPSs), firewalls, etc., available on the market to protect networks, hosts and applications that form part of an organisation's information and technology assets. These tools can be automated for finding and reducing threats, however, automating cybersecurity is not the complete solution for protecting valuable assets within an organisation. The automation requires an analysis of the activities of the intruders to providing better protection.

1.2 Research Problem

Cyber-attack modelling and threat intelligence are techniques that work together to protect the network and data. It is essential to use appropriate cyber-attack modelling techniques in order to better understand a cyber-attack in an organisation's network, and to then handle this cyber-attack efficiently. On the other hand, cyber threat intelligence can work as a preventive measure to protect a network from being attacked. It is crucial that both techniques work together to prevent the cyber-attack as this gives a better understanding of the nature of the attack, which can be valuable knowledge in order to develop cyber threat intelligence. To the best of our knowledge, there are no such techniques that can merge these two disparate techniques to work together to build better protection. In this context, we have identified the following problems in the area of cyber-attack modelling and threat hunting -

- Employees are considered as the first line of defence of an organisation therefore, it is important to understand the security threat to the organisation. Research shows that there is limited or no awareness of cyber threat amongst IT employees of an organisation.

- Cyber-attack modelling is one of the most useful techniques, which is used to understand cyber-attacks. Some methods are used to identify the adversary's artefacts, this technique is used primarily to analyse the post-attack situation, so is not effective for early detection. Furthermore 'Attack' modelling techniques are not fully utilised to check vulnerabilities and protecting networks.

- Cyber threat intelligence is prevalent within the cybersecurity industry. Many techniques are used to understand the threat before it happens. This means that threat intelligence mainly works as a prevention mechanism. There are many techniques, including big data and log analysis, however there are very few, or no research projects that have utilised the Elasticsearch on event log from honeypot in order to actively hunt for cyber threats within the network.

- There is little or no correlations between the cyber-attack modelling and threat intelligence have been identified by researchers. Cyber-attack modelling, such as attack graph, cyber kill chain and diamond model work during and after the attack takes place. These techniques also elucidate the strengths and weakness of a network. Moreover, cyber threat intelligence is a data-driven approach, which aims to build a defence from data using prediction.

1.3 Aims and Objectives

This study aims to identify the current level of cyber-attack modelling and intelligence systems by investigating attack related data from honeypots. The main objectives of this research are to propose a threat intelligence technique to provide better security to an organisation's network. The following describes in detail the research objectives -

- The aim is to understand the current knowledge that IT employees possess about security or attack factors by using quantitative and qualitative data analysis. Also, a suggestion will be made to improve the knowledge of IT employees on the subject of systems and security.
- Detect attacks and analyse them using attack modelling techniques in order to understand the cyber-attack and the adversary. Critically analyse three well-known cyber-attack modelling techniques and compare the results with our proposed cyber-attack model.
- Develop cyber threat intelligence using proactive threat intelligence techniques. This analysis involves the setup of honeypot on the cloud, next data will be collected from the honeypot, and finally analyse the data to find the attack event using Elasticsearch technology because of the volume of the data set.
- Introduce a new threat intelligence technique using attack, behaviour, and pattern. The main idea here is to identify attack events from the honeypot log data. In order to do this, data will be analysed to determine the behaviour and find a pattern of attack

in order to prevent a future attack, using both quantitative and qualitative research methods.

1.4 Contributions

The key contributions of this research are -

1. ***Cyber Threat Awareness*** - Threat awareness is one of the fundamental issues in cyber threat modelling and intelligence amongst IT employees as they are at the front line of defence and weakness. Therefore, Cyber threat awareness has become the initial tool used to analyse the knowledge possessed by IT employees within an organisation. The benefits of this are two-fold: it helps to identify the understanding of the importance of cyber threat amongst employees, and aims to understand how attacks are handled using attack modelling techniques. In order to do this, a range of IT employees were surveyed who may or may not have handled cyber-attacks directly. The responses received were very informative, they showed a clear lack of knowledge amongst IT employees on the subject of cyber-attacks. This data led us to propose a number of security factors that would can aid the enhancement of employee knowledge and in turn offer better protection. In chapter 3, we present a detailed discussion of the cyber threat awareness process and discussion.

2. ***Cyber-attack analysis with modelling techniques*** - An evaluation study of three prominent cyber-attack modelling techniques was carried out, this provided an in-depth analysis of these models using a case study scenario. In order to compare, this study, we have proposed a cyber-attack modelling technique called an object-based model, which considers direct and indirect objects in the attack surface. We evaluate and compare the object-based model with the three models to understand the characteristics of those techniques. The result leads to propose a number of criteria that an attack

modelling technique should provide in the event of an attack, details of this analysis are provided in Chapter 4.

3. ***Cyber threat intelligence*** - Cyber threat intelligence is a conceptual model for understanding the nature of cyber-attacks, motive or behaviour of an attacker and the pattern formed by several attack events. To analyse the threat intelligence model, we have collected log data from the cloud-based honeypot, where the honeypot behaves like a real system. The log data is analysed and visualised using well-known Elastic stack. The analysis exposed a number of insights into the cyber-attack. The result shows that using honeypot data analysis could improve the security of an organisation instead of using a production system. Chapter 5 provides further details about threat intelligence techniques.

1.5 Research Scope

The survey and interviews were conducted amongst IT employees who work directly or indirectly on a computer. This implies that conducting a survey on those employees who do not work directly or in directly on a computer is out of the scope of this work.

Cyber-attack modelling techniques were evaluated only using Portable Executable (PE) i.e., '.exe' file. Other types of malware or executing files are out of the scope of this research.

In order to analyse cyber threat intelligence, we have collected cloud-based honeypot data. Data from another source such as firewall, IDS, IPS, etc., could provide a different result. So, non-cloud honeypots, honeynets, firewall, IDS and IPS are not within the scope of this research.

Finally, this research does not produce any security software. However, the threat intelligence model can be used in protection systems such as IDS and IPS, which can add a layer of security to an organisation's network.

1.6 Thesis Structure

The rest of this thesis is organised as follows -

- **Chapter 2:** Literature Review

  This chapter discusses a range of cyber-attack modelling techniques that are used for handling a cyber-attack. It also discusses a variety of cyber threat intelligence techniques and honeypot systems.

- **Chapter 3:** Cyber Threat Awareness

  The survey has been conducted through a questionnaire distributed amongst the IT employees of different organisations. This chapter discusses the data collection and analysis to understand the knowledge level of IT employees in the organisation. Furthermore, the chapter also suggests the protection mechanism that each of the organisations must apply.

- **Chapter 4:** Cyber Threat Modelling

  This chapter analyses an attack scenario for an IT employee in an organisation using three well-known attack modelling techniques. Moreover, a new attack modelling technique is proposed and analyses the same attack scenario. The result is compared with the three modelling techniques specified earlier. Finally, this chapter proposes a number of criteria for future attack modelling techniques.

- **Chapter 5:** Cyber Threat Intelligence

  This chapter covers design, implementation and analysis of cyber threat intelligence techniques. The chapter starts by discussing the new threat intelligence technique using attack, pattern and behaviour. The threat intelligence technique is also presented using the formal definition. The honeypot data collection and analysis are discussed in details. Finally, this chapter evaluates the results.

- **Chapter 6:** Conclusion and Future Work

  This chapter summarises the thesis contributions and highlights the benefits and limitations of the proposed models. Finally, it presents the future enhancements of cyber-attack modelling and threat intelligence techniques.

1.7 Publications

1. Almohannadi, H., Al Hamar, J., Awan, I. (2019). Analysis of Adversary Activities using Cloud-Based Web Services to Enhance Cyber Threat Intelligence. *Springer Journal of Service Oriented Computing and Applications*.

2. Al-Mohannadi, H., Awan, I., Al Hamar, J., Al Hamar, Y., Shah, M., & Musa, A. (2018, August). Understanding Awareness of Cyber Security Threat Among IT Employees. In *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 188-192). IEEE.

3. Musa, A., Almohannadi, H., & Alhamar, J. (2018, August). Malware Propagation Modelling in Peer-to-Peer Networks: A Review. In 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW) (pp. 198-202). IEEE.

4. Almohannadi, H., Awan, I., Al Hamar, J., Cullen, A., Disso, J. P., & Armitage, L. (2018, May). Cyber Threat Intelligence from Honeypot Data Using Elasticsearch. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)* (pp. 900-906). IEEE.

5. Al Hamar, J., Chamieh, J., Al-Mohannadi, H., Al Hamar, M., Al-Mutlaq, A., & Musa, A. S. (2018). Biometric of Intent: A New Approach Identifying Potential Threat in Highly Secured Facilities.

6. Al-Mohannadi, H., Mirza, Q., Namanya, A., Awan, I., Cullen, A., & Disso, J. (2016, August). 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). IEEE.

7. Namanya, A. P., Mirza, Q. K. A., Al-Mohannadi, H., Awan, I. U., & Disso, J. F. P. (2016, August). Detection of malicious portable executables using evidence combinational theory with fuzzy hashing. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 91-98). IEEE.

8. Namanya, A.P; Mirza, Q.K.A; Al-Mohannadi, H; Cullen, A; Awan,I (2016): Towards Building a Unified Threat Analysis and Management Framework; *(UKPEW) & Cyber Security Workshop (CyberSecW)*

9. Munir, R., Ahmed, B., Al-Mohannadi, H., Mufti, M. R., Namanya, A. P., & Awan, I. (2016). Performance Security Trade-off of Network Intrusion Detection and Prevention Systems.

Chapter 2

Literature Review

2.1 Introduction

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-based detection systems that allow or reject traffic according to the rule-sets. On the other hand, cybersecurity 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 sophisticated approach than the traditional rule-based detection system [17].

To detect or handle a cyber-attack, it is essential to learn about the weaknesses of the network. It is also necessary for the cybersecurity team to understand the motive of the attacker, what data could be targeted and why the attack happened. Proper planning is necessary to deal with a cyber-attack. The attack modelling and analysis has been emphasised in a recent article by Bank of England, which describes how threats can be modelled to mitigate the cyber-attack in any organisation [10]. Attack modelling techniques are important to understand, explore and validate security threats in the cyber world [18]. Building Security in Maturity Model (BSIMM) [19] usages attack modelling techniques in the BSIMM framework of cybersecurity. The goal of BSIMM is to use customised knowledge to handle an attack in an organisation.

Pursuing cyber threats is not a new concept. The threats are usually modelled using various modelling techniques, which is addressed by many researchers [20] [21]. 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 [22] [23], Attack Vectors [24], Attack Surface [25], Diamond Model [26], the Open Web Application Security Project's (OWASP) threat model [18] and the Kill Chain[27, 28]. These modelling techniques can be used individually or in conjunction with other models. In [20] a number of cyber-attack modelling techniques developed to handle cyberattacks 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 [11].

In this chapter, we discuss a wide range of tools and techniques of cyber-attack modelling and threat intelligence.

2.2 Attack Modelling

Modelling a cyber-attack, which has not happened yet can save time, money and other resources for an organisation. However, cyber attack modelling techniques can help understanding cyber-attack and mitigate them in order to protect organisation's network and data. This section focuses on reviewing the three main attack modelling techniques called the Diamond Model, the Kill Chain and the attack Graphs for cyber-attack modelling. Modelling a cyber-attack or predicting threat is a crucial issue for securing any corporate network. The goal of this review is not to compare those techniques but to understand the mechanism to model cybersecurity threats to provide more security for a system.

2.2.1 Diamond Model

The Diamond Model is one of the novel models for cyber intrusion analysis described by Caltagirone et al. [26] where an adversary attacks a victim depending on two key motivations; capability and infrastructure of a victim, rather than using a series of steps like the Kill Chain. This model consists of four basic elements such as adversary, infrastructure, capability and victim. An adversary is an actor (or set of actors) who attacks a victim after analysing their capability against the victim. Initially, the adversary starts with no knowledge of the capability of the victim. After examining the capability of a victim, the adversary may find that he/she is more capable than the victim to attack or not.

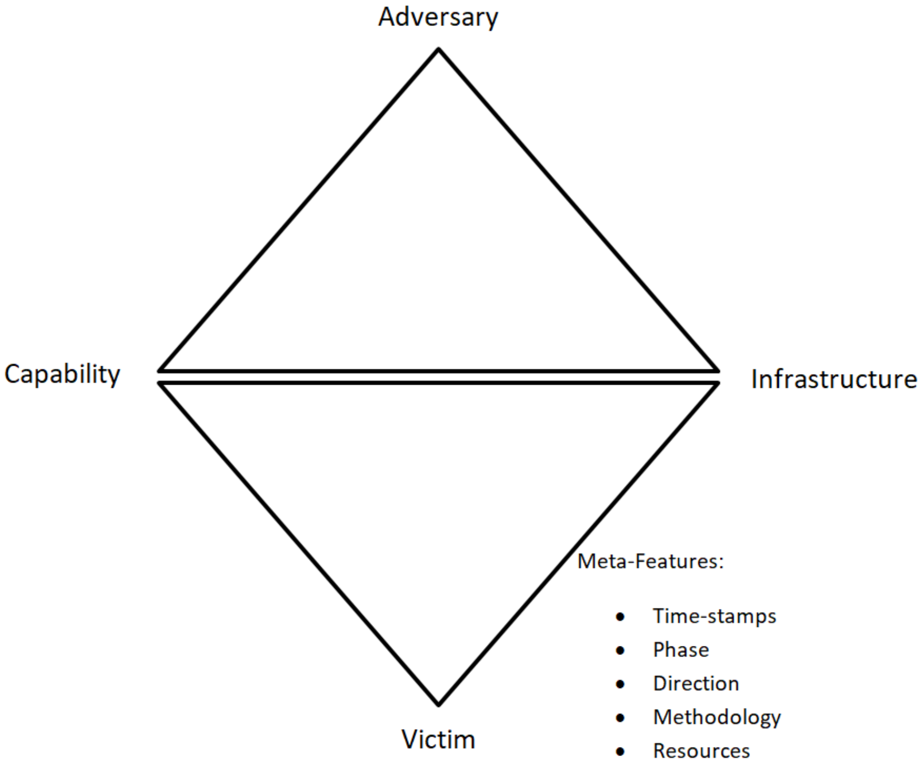


Fig. 2.1 Diamond Model

This model is important when dealing with more advanced attackers such as those who have already gained some control over the network. The adversary also analyses the infrastructure of his/her technical and logical ability to command and control any victim's network. The Diamond Model is also associated with some meta-features such as time-stamps, phases, results, directions, methodology and resources. In the event of an attack, the Diamond Model identifies phases in a time-stamp [26]. Components of the Diamond Model can be found in Figure 2.1, which illustrates that the adversary looks for an opportunity to attack a victim depending on the capability or the infrastructure.

2.2.2 Kill Chain

The Kill Chain for intrusion is one of the models, which focuses on cybersecurity. It is a structured attack since the attacker progresses the attack in an ordered chain according to a plan [28]. The Kill Chain technique is described by the US Department of Defence to attack a target [29], where they have defined the Kill Chain with stages such as, find, fix, track, target, engage and assess. The Kill Chain has been applied in other areas, including cybersecurity. In cybersecurity, [30] it is used to describe some attack steps within a countermeasure framework. Additionally, taxonomy based threat intelligence using Kill Chain to get step-by-step functional process of cyber-attack such as banking trojans, which is highly beneficial on cyber threat mitigation [31]. It can also address incomplete or incomplete alarm generated by Security Incident Event Management (SIEM) system through remote log analysis [32].

The research has led the Kill Chain to have seven steps of attack, which can be described below as -

Step 1 Reconnaissance: The attacker gathers information before an attack. The information could be collected from the Internet, which is publicly available.

Step 2 Weaponization: The attacker creates a malicious payload to send to the victim. The payload could be a virus, a trojan or an executable file that can perform some action on the victims' machine or on the network.

Step 3 Delivery: Attacker sends the malicious payload to the victim using some means of communication. The attacker may send the payload via email as an attachment or a link that will download the payload.

Step 4 Exploitation: In this stage, the actual exploitation takes place. If the victim has to download the payload into his/her computer, the main exploitation starts. This is the stage where the attacker needs the aid of the victim. Also, this is one of the phases where the chain can be killed.

Step 5 Installation: Install malware on the infected or victims' computer. To infect the victim's computer, the payload may need to be executed by the victim, or it can be automatically executed. This is also the phase where the chain can be broken.

Step 6 Command and control: Through the installed malware, the attacker creates a command and control channel to access the internal assets of the victim. In this phase, the attacker has successfully gained control of the victims' machine.

Step 7 Action on objectives: Attackers achieve their goal on the victim's computer or network that is infected. This could be the gateway of the attack. The attacker may progress towards valuable data from the database through the web server.

The Kill Chain is divided into two major phases called left of exploits or hack and right of exploit or hack. In Figure 2.2, we can explore the Kill Chain steps, where we can see that the left part allows the victim to kill the chain. If the attacker managed to move on to the right part of the Kill Chain, it would be difficult for the victim to stop such an attack or reduce loss.

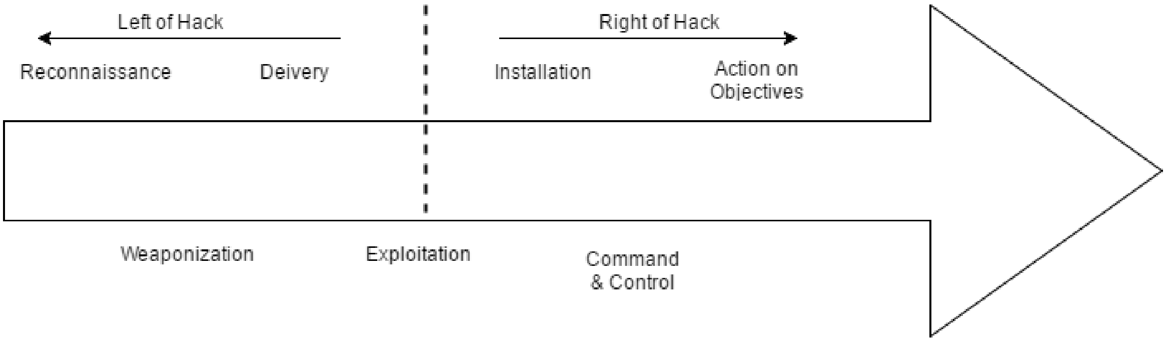


Fig. 2.2 Kill Chain Attack Modelling

On the left side is the initial phase where an adversary will try to gain control of the system. For example, if an adversary wants to gain access to a system; he/she will start reconnaissance, i.e., investigating the victim's network, profile and other profiles available on the Internet [33]. Let us assume that the adversary has managed to gain the email address of the victim. The email may become the gateway to enter the victim's system or network. The adversary will be weaponised in various ways. Conventional weapons could be trojanised PDF, DOC, BNP, etc., files sent to the victim as an attachment [34]. The sender tries to be as legitimate as they can, they may also put a link to click. The email subject, ID and domain name should look like the real one. Once the email is delivered, the adversary waits for the victim's response. In this situation, the victim may download the attachment or click the link, or delete the mail on suspicion. Exploitation plays a vital role in this situation as, if the victim clicks or downloads, the Kill Chain will advance, i.e., malware will be executed on the victim's machine.

In this context, it is clear that the left-exploits/hack is the most important for cyber defence for any organisation or individual. From the left-hack analysis, we can find a number of insights into such attacks. It can indicate the 'attackers' pattern of attack, sending IP, email domain, location and other relevant information, which can be collected for defending the victim. By understanding the pattern of the attacker, the organisation can educate employees on how to deal with such an attack. In most situations like this, a Kill Chain can be broken at an early stage. Understanding the kill chain can help organisations protect themselves from potential cyber-attacks.

2.2.3 Attack Graph

Attack Graphs are conceptual diagrams used to analyse how a target can be attacked. This is important to analyse cyber threats on a computer system or network. An attack Graphs is a tree-structured graph, which has multilevel children with a single root. This was originally introduced by Schneier et al. [23] to develop a tool for valid defence by analysing network vulnerabilities. The graph essentially consists of nodes and can be complex in nature when dealing with a specific attack. It may contain thousands of nodes with a number of different paths. Generating Attack Graphs is computationally complex [35], especially in the case of large networks. There are a number of Attack Graphs generating tools and techniques such as TVA (Topological Analysis of Network Attack Vulnerability) [36], NETSPA (A Network Security Planning Architecture) [37] and MULVAL (Multihost, Multistage, Vulnerability Analysis) [38] as reviewed by Ou et al., describes the quantitative security assessment of enterprise networks [39]. These tools help to draw logical Attack Graphs to understand "why an attack happens?" rather than "how an attack happens?". The main idea of an Attack Graphs is the path from the attackers to the victim's network. Attack graph techniques help to detect intrusions and the vulnerability of the system.

An example of an Attack Graphs is given below –

Attack Graphs can be useful in many areas of computer network security including intrusion detection, forensic analysis, risk analysis and cyber defence. A network administrator uses an Attack Graphs to identify,

- The vulnerability of the system

- How an attack can happen
- A set of actions that will prevent an attacker in achieving their goal

The main advantage of an Attack Graphs is that it helps to identify any potential attack on the network. The analysis helps to identify necessary steps if there is any weak point in the network. Using this technique, it is possible to calculate Return on Investment (RoI) for security. Organisations mainly avoid security or vulnerability checking because it is expensive [36]. On the other hand, if the cost is too high for the company, it is unlikely that the company will go for an expensive option. So, companies need a clear vision of investment in cybersecurity.

Generating an Attack Graphs is a challenging task, as there are hundreds of nodes that can be involved in the graph, which makes it challenging to identify a valid attack threat. Also, there are a number of uncertainties involved in the Attack Graphs technique. To deal with this uncertainty, some researchers use the Monte Carlo algorithm [35] as it can deal with uncertainty and has statistical dependents. Other algorithms, such as breadth fast and depth fast algorithms, are also used to create the graph. There are numerous theoretical works that have been done in the area of Attack Graphs generation. Some researchers use Ontology involving entities, properties, relationships and rules to model cyber-Attack [40].

2.3 Threat Intelligence

Cyber threat intelligence is the process of gathering and analysing attack related data to track, trace, identify and predict adversaries' intention and activities in a corporate network. It also deals with the action plan to extend decision making to mitigate the loss of data and valuable resources.

Cyber threat intelligence is a complicated process for an organisation's network administrator or security personnel. Threat hunting aims 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 [41], a low interaction client-based honeypot, emulates only essential features of target clients. This is a client honeypot (HoneyC), which can detect client-side attacks. 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 session honeypots are used for experimental purposes. Honeypot is one method of developing an understanding of any cyber-attack. More specifically, an SSH honeypot is analysed while the session is running and the data is visualised using a visual analytical technique [14]. Besides, honeypots are used to mitigate Advanced Persistent Threat (APT), which works with a combination of human and automated systems. The attacker in an APT situation does not jump into any attack without initially conducting reconnaissance and planning the attack.

Jasek et. al [42] used honeypots to detect cyber-attacks. Honeypots are an 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-attacks. Distributed Denial of Service Attack (DDoS) is a challenging threat to an organisation. Weiler [43] simulated the DDoS attack using honeypots to learn more about such a cyber-attack on network infrastructure.

Honeypots are also able to emulate mobile devices to understand the threat on a smartphone. The honeypots emulate a real phone and collect data to understand what kind of malware infects smartphones. Such honeypots are called Nomadic [44] and provide infrastructure to collect threat intelligence data. The monitoring is also carried out using visualisation techniques. A low-medium interaction honeypot called Dionaea is used to collect and analyse attack data to understand the trend of cyber-attack [45]. They also build individual attacker profiles by analysing collected data.

It has been noticed that many researchers do threat hunting by using honeypots data collection and analysis. On the other hand, honeypots produce a huge amount of data. It is not easy for general-purpose data analysis tools to analyse such large amounts of data. In this paper, we invoke Elasticsearch technology to analyse honeypots data as it gives the flexibility of searching on any data set size. It is well known that honeypots and honeynets are unconventional security tools that allow security personnel to collect data an analyse them to learn more about the cyber-attack. In [15], authors collected data from honeypots to hunt cyber-attacks patterns. Honeypot data collected by Moor et al. [16] captured the IP address of the attacker for further analysis.

2.3.1 Pyramid of Pain

The Pyramid of Pain (PoP) was introduced by Binaco [17], it analyses how an indicator of compromises behaves. The main purpose of the PoP is to establish the different levels of IoC for cyber defence. The pyramid indicates the level of difficulties for handling cyber threats for an organisation's security team.

Figure 2.3 shows the PoP, which gives levels of technical difficulties for both the adversary and the victim. PoP provides a simplified view of the adversary's activities. The adversary uses the PoP components for attacking a network. Besides, in a cyber-attack the adversary leaves some footprints, which could be the combination of the PoP components. So, analysing PoP could reveal the nature and motive of an adversary, which could be used to make an informed decision for threat intelligence. The trivial metric of the pyramid is at the bottom and is called the Hash Value. Hash values provide a 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. Therefore it is not worth following them around or setting rules to catch them as they could keep coming with a different value. This means that attacks with hash values are easy to identify and tackle, so, the possibility of a system

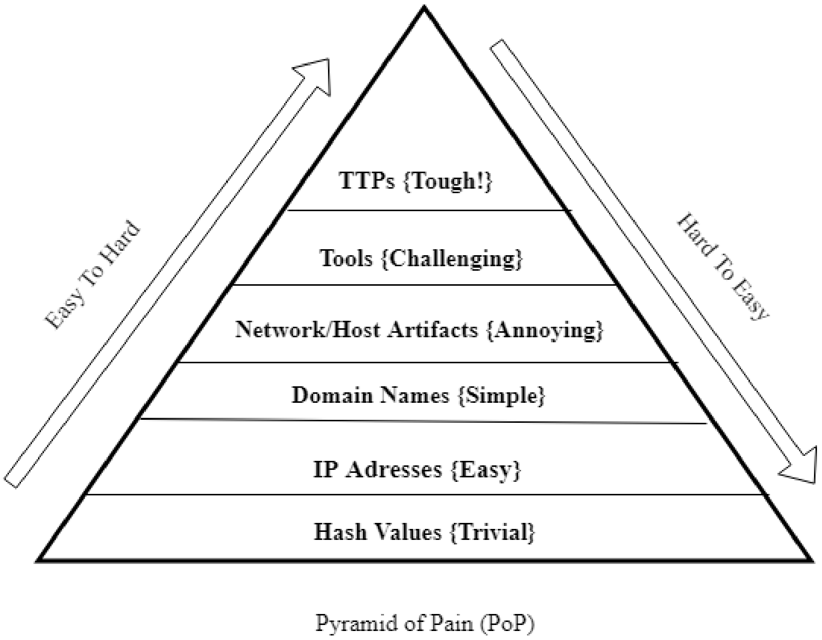


Fig. 2.3 Pyramid of Pain

Table 2.1 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

compromise is very low. In a cyber-attack incident, IP addresses are fundamental indicators for identifying an attacker. It is hard to hide the IP address during a cyber-attack event. For an attacker, it is straightforward to change the IP address after the attack or masquerade before an attack. In practice, it is not feasible to pursue every single IP address that has tried to breach a system. Table 2.1 shows how PoP behaves in practice.

The adversary must have registered with a hosting company to get a domain name. It is relatively easy to trace back to the origin of the domain, although they could be disguised.

The domain name can be changed at any time. Since they have to register, it is a bit harder to change the domain name compared to the IP address.

The next indicator is the network artefact, which could differentiate the malicious activities of the adversary from legitimate users. Hosts that are involved in a cyber-attack contain a lot of information about the attack. Host artefacts are the indicators of malicious activities performed within the host. These can be used to distinguish the actions 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 can be software, hardware or a combination of both, which are used to deploy or plant the payload. New or customised tools can be a great challenge for the analyst. The final and top component of the pyramid is Tactics, Techniques and Procedure (TTPs), which is used for cyber warfare [2]. This is the level where the behaviour of the adversary can be identified from the malicious software or the payload.

2.3.2 Hunting Maturity Model

The Hunting Maturity model is a cyber-threat hunting model that identifies an organisation's threat hunting ability, including the quantity and quality of threat data collection. Hunting Maturity Model (HMM) also indicates the way of analysing and visualising of the data [11]. 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 increase 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 uses automated systems for data analysis. Table 2.2 and Figure 2.4 show the HMM, which is linear in nature. The main idea behind the HMM is it requires continuous improvement of data collection and analysis.

Table 2.2 Hunting Maturity Model

Level	Maturity	Comment
Level 0	Initial	Depends 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

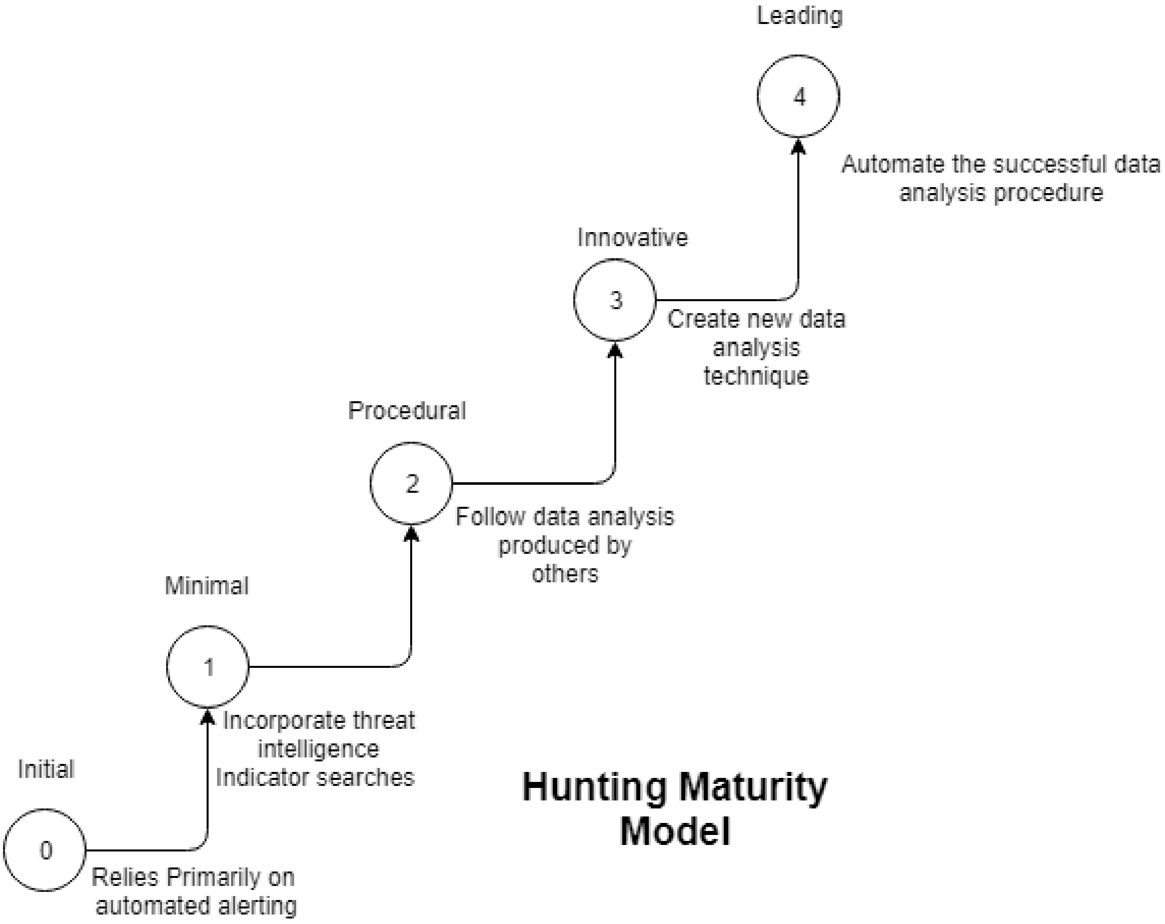

Leading
4
Automate the successful data analysis procedure
Innovative
3
Create new data analysis technique
Procedural
2
Follow data analysis produced by others
Minimal
1
Incorporate threat intelligence Indicator searches
Initial
0
Relies Primarily on automated alerting
Hunting Maturity Model

Fig. 2.4 Hunting Maturity Model

Hunting is not a one-off action; it is a process. Threat hunting 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 [11] introduced the hunting loop as shown in Figure 2.5. 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. Maver et al. used data-driven threat hunting from Sysmon logs, which indicates different level of cyber threat in and organisation [46].

The HMM process is conducted in the following steps -

- Data Collection - To hunt a real threat within the network or host, collecting data is the most important element, this could be collected from a range of different sources. This data could be syslog, honeypots data, network activity data, firewall data, server logs, etc., which could be used for creating a hypothesis. In most cases, data collection could then be automated, which could be fed to the analytic system or the visualisation software.

- Hypothesis Creation - To hunt real threat within the network or host, collecting data is an essential element. Data could be collected from different sources. This data could be Syslog, honeypots data, network activity 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 to the analytic system or the visualisation software.

- Tools and Techniques for Hypothesis Techniques - From the data collection to automation, there are many tools and techniques required to hunt a proper cyber threat. Basic log analysis tools or SIEM give a minimal level of flexibility for mature hunting. The hypothesis must be tested against the tools and techniques used for threat hunting. In most case, the advanced level of visualisation should aid and examine in order to create a new hypothesis.

- Pattern & TTP Detection - APT [42] or Zero Day Attack [13] are difficult to identify or predict in advance. Zero Day attack do not match any attack pattern, therefore, it is important to create patterns for identifying a normal attack and 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 is a key factor in such a situation. The threat hunting process from data collection to detection needs to be automated to manage cyber threat incident events efficiently.

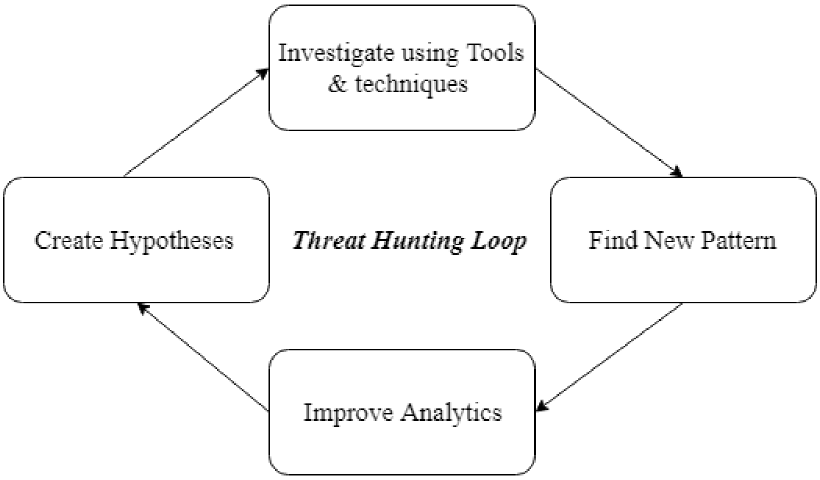


Fig. 2.5 Threat Hunting Loop

2.3.3 Matrix of IoC

The indicator of Compromise [47], which are considered as digital forensic artefacts that can be compromised during and attack, can be put in the form of a matrix. Each of the indicators is evaluated using three criteria such as trace, identify and response. After a cyber incident event, a threat hunter, investigates all the relevant IoCs to stop such attack. In the following, we discuss three criteria against the IoC for a better understanding of these indicators.

- Trace - It is important to find a trace of an attacker during his or her visit to the network or in a host. Trace of a hash value is not worth much as the attacker could change them during the next attack. The payload may be changed and the hash value will be different. Moreover, hash value could one way, which could not be tracked for the original data [48]. So, it will be challenging to identify if the attack was made by the same attacker. The IP address is the key to making a connection between devices as each of the network devices must have an IP address. The attacker may change the IP every time they make an attack, which is also true for domain names.

  Moreover, the attacker may leave a network or host artefacts although they have changed hash, IP or domain name. So, these network or host artefact are essential elements to investigate further. Some users may use the same tool repeatedly to make a cyber-attack as changing attack tools may require developing and testing, which could be expensive. So, the tools could be traced to identify if they are using the same tool or not. In the top of the pyramid, the TTP is the most critical and challenging indicator since TTP mainly expresses the skills and training of the attacker. An attacker may improve his or her skill set over time, which makes threat hunters think harder.

- Identify - Tracing helps threat hunters to track attackers. Any evidence left by the attacker could be used to identify the attacker in a future attempt. For example, the tracking system can be used to match identified components such as IP, hash values and domain name. Identifying the network or host artefact could help to analyse attack behaviour.

- Response - If the threat is traced and identified, it is required to prevent future events happening. For example, if an IP address is identified as a threat element from the analysis, it could be black-listed for any future events. If a quick response is made to the identified threat, the defence becomes offence [17].

2.4 Honeypot and Cyber Threat Intelligence

In the previous sections, we have discussed different techniques used by the organisations to identify cyber threat intelligence. We have also discussed analysis techniques that are used to understand the characteristics and behaviour of an attacker and attack patterns. Additionally, we have presented a discussion on different conventional attack modelling techniques that are quite commonly used by security professionals to handle a cyber-attack. In this section, we discuss recent research conducted that is relevant to our research along with their inherent strengths and weaknesses. Our work focuses on cyber-threat intelligence using honeypot data collection and analysis to provide further insight into attack events. We specifically discuss recent developments in this area.

Honeypots are used to collect cyber incident data and analyse them in order to understand cyber-attacks [16], [45], [49], [15]. A recent honeypot-based data analysis is conducted by Nawrocki et al. [50] provides an extensive survey on honeypot technology. The literature has identified that honeypots are tools that can record attacks or resource compromise data [15], which complement the traditional attack detection system [51].

Additionally, cloud-based honeypots are emerging with different types of services [52], where users are allocated with virtual machines instances. Each of the studies has its own methodology to approach the cyber threat handling by increasing true positive and reducing false positive alerts. Honeypots data analysis is different from the traditional threat intelligence systems as this can identify attack source, attackers and attack types without interacting with a real systems.

Understanding and predicting cyber-attack is a challenging task. This requires active threat hunting using big data analysis. Honeypot technology provides a safe way of collecting cyber threat-related data. It is important 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 it to understand the attack profile such as motivation of the attacker, breadth or depth of an attack, sophistication, concealment, attack resource, vulnerability and tools used for the attack [53]. A number of researchers have used honeypot technology to understand various types of attacks such as malware attack, botnet activity, phishing and spreading spam.

Honeypot could be deployed within or out of the network. However, it is not safe to deploy honeypot in a network where there is a possibility of getting to a production system. Honeypot can also be deployed in the cloud [52]. Cloud is a secure option as the cloud providers handle all the security matters. Cloud honeypots can be used as an additional security service for cloud users, which could be considered as Infrastructure-as-a-Service (IaaS).

Honeypots are classified according to 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. Server-side honeypots could be used in various ways such as web application, SSH, VoIP, Bluetooth, USB and Sinkhole honeypots. On the other hand, client honeypot consists of client-side application such as a web browser and has the ability to connect to remote server or services. The advantages of using honeypot are to collect data that it can play both client and server roles [50].

There are a number of honeypots taxonomies available for classification. The primary taxonomy is based on the type of attack resources and level of interaction. The type of attack resources describes that if the honeypot is server-side or client-side. On the other hand, the level of an interaction represents weather the honeypot is a real resource i.e., high-interaction or emulated i.e., low interaction honeypot. However, the combination of both low and high-interaction honeypot is called hybrid honeypot [54]. Kippo is a medium-high interaction honeypot and an SSH honeypot, which can be deployed to the cloud as Unix operating system. 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 [55]. Furthermore, honeypots can be used in the production system to protect company's valuable assets. The honeypot that are deployed in the production network to enhance network security by preventing, detecting and responding cyber-attack in real-time [56].

Other systems like Sandboxes [57] are mainly used to analyse malware in isolation. They can be deployed in a physical or virtual environment. Both honeypot and sandbox are deployed when there is any malicious activities within the network. However, they are very different in operation, for example, Honeypots are mainly deployed to identify malicious activities, where the sandbox is used to do a deep and automated analysis of the infection process. Honeypots may work like IDS as they can detect attack but cannot take action. Since honeypots do not interfere with the network activity, it cannot be used as the replacement of IDS [55].

2.5 Summary

In this chapter, several aspects of cyber threat intelligence have been discussed. Discussion started with various types of attack modelling techniques. Numerous attack modelling techniques such as Diamond Model, Attack Graphs and Kill Chain have been discussed in detail in order to understand the current state of cyber-attack modelling. We have explored both old and modern attack modelling techniques to understand the application of best practice in cyber-attack modelling by the organisations.

Furthermore this chapter also discusses the threat intelligence process that is used currently within the industry to tackle the cyber threat. There are a number of techniques that use data collection and analysis for predicting future cyber-attacks to the organisation network. Also, it discusses how honeypot technology can support in active threat hunting in the organisation's cyberspace. Using this fundamental knowledge as a reflection point, the next three chapters evaluate and analyse cyber threat intelligence. This allows for the development of the analytical approach that contribute to the thesis.

Chapter 3

Cyber Threat Awareness

3.1 Introduction

Protecting the network from external attackers is one of the priorities by the organisations. However, the main issue identified by many researchers in cyber-attack is the employee within the organisation as they are the front line of the defence of an organisation for protecting the network. They are also the biggest threat to the organisation's system and data. Cyber attack modelling mainly concerns with the cyber attacks, which are handled by security professionals such as Security Operation Centre (SOC) or Network Operation Centre (NOC). However, employees in the organisation, who use computers for daily business could become a threat at any time due to a lack of enough knowledge about security and vulnerability. To build a better defence system, an organisation must have an understanding of the skills of the employees. Also, it is essential to know the security awareness of IT employees. There are a few research have been conducted in this area. However, this is not enough to tackle the current volume of an attack made to organisations' resources.

In this chapter, we examine the awareness of cyber-security threat among all IT employee focusing on their Knowledge, Monitoring and Preventing against cyber-attacks. We have collected data from IT employees from various organisations who are responsible for handling cybersecurity events. The data is collected using an interview and questionnaire. This chapter also aims to advise security awareness, which could be adapted depending on the organisation's need and strategy.

3.2 Motivation

Information technology is changing the way we do business and communication. Organisations are increasingly depending on information technology to improve their product and quality of services. It is hard to secure personal and organisation's data because of vulnerabilities issue. European Union has published new data protection regulation called General Data Protection Regulation (GDPR)[1], which aims to protect user data. GDPR is a guidelines, is not enough for the organisation to protect their assets, they need to train people [58] and build cyber defence [1].

Even a small organisation maintains mailing service to communicate with employees, clients and stakeholders. Malicious emails can ruin the reputation of an organisation. Such an attack is called a phishing attack where an attacker sends spam emails to employees to an organisation pretending to be a genuine one [59]. It is tough for an employee to decide whether to click the link or not. However, appropriate training on information security awareness could support employee decisions. Information security awareness gives the user more understanding of the importance of the best practice. It is vital to provide training to all the employees within an organisation [60]. Since IT employees are at the front line of cyber-attack, they could cause more harm than the outsiders. Simple negligence of an IT employee can cost an organisation not only money but also valuable data [61]. The threat from the employee may not always be intentional; it could be because of the lack of enough knowledge about cyber risk and consequences. So, it is crucial to understand the level of consciousness that IT employees have regarding cyber threat in a corporate

[1] https://www.eugdpr.org/

network. Organisations should also look for security awareness services, which can help both employees and organisation understand the weakness of the network system [62].

Honeypot is used to detect, identify and gather information within the company to reduce cyber risk from the employee, i.e., insider threat [63]. In many occasions, cyber-attack happens to an organisation's network because of the lack of knowledge among the employees. Shaw et al. [64] studied on security awareness and training effectiveness [65]. They have identified that there are several barriers in organisation for security awareness, such as budget, computer skills and general security awareness. Cybersecurity awareness delivery method is important within an organisation. The research suggested that combined security awareness is better than individual delivery [60]. Most of the organisations provide basic training to the employees. These training are mostly online-based or basic introduction. Security awareness is beneficial [66], which could give the user more understanding about cyber threat. Change of behaviour is important within an organisation in conjunction with cybersecurity training. A phishing attack is very common to any organisation to understand user awareness, which mostly used by the attackers using email. Nalin et al. investigated phishing threat on human behaviour [67], using mobile phone game prototype, which shows a significant improvement on participants to avoid phishing attack. User security awareness exercises is helpful using internal phishing attack [68].

3.3 Research Method

We have surveyed by distributing questionnaires (see Appendix A) to various cyber threats handling teams. The main goal of the cyber-attack survey is to support threat hunting as the knowledge and experience of the cybersecurity team is essential to mitigate cyber-attack. These teams include several different cybersecurity professionals, who work directly or indirectly in the cybersecurity team. SOC personnel mainly deal with direct cyber-attack to an organisation's network. Some other people in the organisation do not handle or tackle cyber-attack directly but support to mitigate problems.

Data collection was performed with the full consent of the participant. In this research, we did not collect any personal data from the participants. Also, the aim of the data analysis will focus on the knowledge and awareness of IT employee about the cyber threat in their network. We sent those questionnaires to the people who are directly involved in cybersecurity or networking.

We have distributed about 50 questionnaires among the participants in different organisations. There were about 22 IT subject matter expert respondents to the survey representing: SOC, NOC, System Admin, Database admin, Network engineer, Application developer and System architect were involved in the study. The response is about 44%, which is acceptable for data analysis as questionnaires. The researchers also have profile into two categories: SOC and Non-SOC team. Since combating this risk is the responsibility of all not only SOC but also the Non-SOC team. There were 17 respondents from the non-SOC team who are IT Subject matter expert from a different domain and only five from the SOC team who deal with daily security threats. These expert IT employees, who are expert in their area, have answered the following questions in the questionnaire -

1. In terms of your security duty, do you have a defined checklist for your daily task?
2. What the common attacks do you handle normally?
3. Do you recognize any of the terms during a cyber-attack?
4. Which of the following IoC is the most/least difficult to trace in your environment?
5. What are the common alerts do you handle daily?
6. In case of repetitive attacks, what action do you take?
7. Do you have any procedure to follow in case of attacks?

8. Do you use any (firewall, IDS, IPS, router etc.) log data to understand activities in the network?

9. Do you have an operation center to monitor all attacks?

10. Is your workstation/Server implemented using a “managed” client/server architecture, or in a stand-alone to push the policy configuration and update?

11. Do you have Demilitarised Zone (DMZ) for external and firewall for internal cross-site?

12. Does it help in isolating or preventing the attack?

These questions are designed as multiple choice question for the convenient of the participant and data analysis. Participants also have the option to answer differently if the desired answer is not in the list.

We have also use interview technique to SOC and cyber security researchers. The following questions were asked to participants during the interview sessions -

- What do you do in a normal day?

- What kind of attack do you handle normally?

- Do you recognise any of the components from the following during a cyber-attack?

  1. Hash values
  2. IP Address
  3. Domain Name
  4. Network Artefact
  5. Host Artefact
  6. Attack tools (e.g., used same tool or different one for each of the attacks)
  7. Other special techniques

Which one is the most or least difficult to trace for supporting the alert system?

- What kind of alert you get?
- If you see same attack happening in your network, what action do you take?
- Do you have any procedure to follow?
- Do you use any (firewall, router etc.) log data to understand activities in the network?
- Do you use any operation centre to monitor all attacks?
- Is all the machine (Computers) have the same configuration and update in the operation room?
- Are the firewalls are flat or cross site?

3.4 Cyber-Attack Survey

We have surveyed by distributing questionnaires to various cyber threat handling teams. These questionnaires aim to understand the knowledge and awareness of IT security staff. It is essential to know how IT security staffs react during an event of cyber-attack.

We have conducted research on various teams that includes many different cybersecurity professionals, who work directly or indirectly in the cybersecurity team. SOC personnel mainly deal with direct cyber-attack to an organisation's network. There are some other personnel in the organisation who do not handle or tackle cyber-attack directly but supports to mitigate such problems. The survey has examined the awareness of cybersecurity threat among employees within the organisation. We mainly focus on three domains of cyber threat intelligence such as

- Knowledge - To identify how much knowledge of an IT employee has for cyber security related tools, techniques etc.

- Monitoring - To understand how the cybersecurity team perform cyber incident monitoring.
- Prevention - To understand how IT employees prevent cyber-attack events.

3.4.1 Survey Result

The result indicates that there is a gap of knowledge between Security operation team and other IT subject matter expert. Interestingly, only 68.1% of SOC team and less than half (48.4%) of the non-SOC team are knowledgeable about IoC. Likewise, only 65% of SOC teams are aware or use Access Control Lists (ACL) and 41.9% of non-SOC. Moreover, only 75% of the SOC team and 45.2% of the non-SOC team are reviewing user privilege access activity.

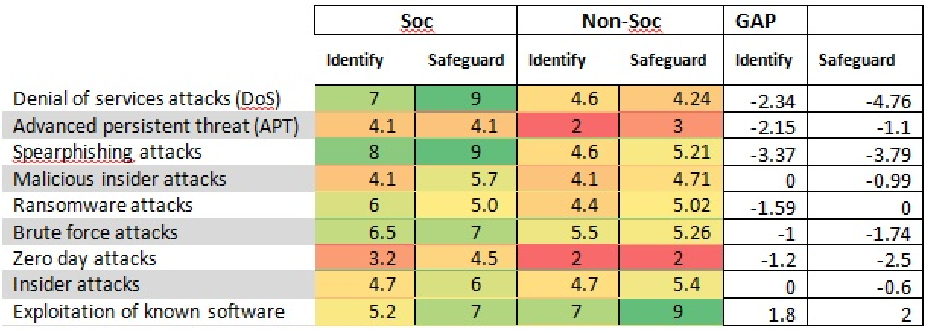
Table 3.1 indicates that while the SOC team have a better knowledge of cybersecurity threat, the non-SOC team show less positive results where most of their score are below 50%. Figure 3.1 intends to explore the capability of the IT employee to identify and safeguard from cybersecurity threats. The data in Figure 3.1 shows that SOC team are generally capable of protecting from cybersecurity threats if they can identify it. Moreover, the more significant cyber threats such as Zero-day attacks, malicious insider attacks and APT are hard to identify. These types of attacks are advanced and require highly skilled hackers to identify unknown world-wide vulnerability in the victim IT Infrastructure and then plan to get access to the network. However, security experts recommend having disaster recovery, including a business continuity plan to reduce the impact of such an attack. Result also shows a significant gap between SOC and non-SOC, which need to be narrowed.

Figure 3.2 shows the methods used by the responders to ensure their ability to identify security threats.

The result shows that each member of the IT domain is using some methods to identify the threat or check the health of their system. Furthermore, only 32.80% of SOC team and

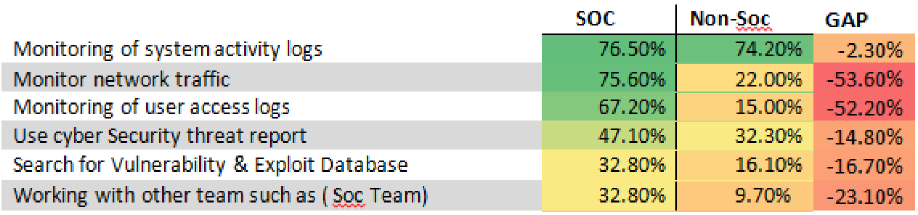
Table 3.1 Knowledge of IT Employee

Knowledge Elements	**SOC**	**Non-SOC**	**Difference**
Antivirus/malware	84.9%	90.3%	5.4%
Firewalls	78.2%	90.3%	12.1%
Indicator of compromise (IoC)	68.1%	48.4%	-19.7%
Data encryption (data in transit)	59.7%	61.3%	1.6%
Data encryption (data at rest)	61.3%	48.4%	-12.9%
Patch and vulnerability management	61.3%	41.9%	-19.4%
Intrusion detection system (IDS)	100.0%	41.9%	-58.1%
Intrusion prevention system (IPS)	100.0%	45.2%	-54.8%
Mobile device management (MDM)	56.3%	35.5%	-20.8%
User privilege controls	75.0%	45.2%	-29.8%
Access control lists (ACL)	65.0%	41.9%	-23.1%
Network traffic monitoring tools	85.0%	45.2%	-39.8%
Web security gateway	80.3%	32.3%	-48.0%
Multi-factor authentication	41.2%	32.3%	-8.9%



	Soc		Non-Soc		GAP	
	Identify	Safeguard	Identify	Safeguard	Identify	Safeguard
Denial of services attacks (DoS)	7	9	4.6	4.24	-2.34	-4.76
Advanced persistent threat (APT)	4.1	4.1	2	3	-2.15	-1.1
Spearphishing attacks	8	9	4.6	5.21	-3.37	-3.79
Malicious insider attacks	4.1	5.7	4.1	4.71	0	-0.99
Ransomware attacks	6	5.0	4.4	5.02	-1.59	0
Brute force attacks	6.5	7	5.5	5.26	-1	-1.74
Zero day attacks	3.2	4.5	2	2	-1.2	-2.5
Insider attacks	4.7	6	4.7	5.4	0	-0.6
Exploitation of known software	5.2	7	7	9	1.8	2

Fig. 3.1 Capability to Identify or Safeguard IT infrastructure

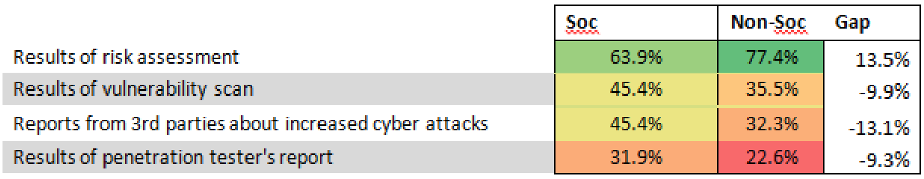


	SOC	Non-Soc	GAP
Monitoring of system activity logs	76.50%	74.20%	-2.30%
Monitor network traffic	75.60%	22.00%	-53.60%
Monitoring of user access logs	67.20%	15.00%	-52.20%
Use cyber Security threat report	47.10%	32.30%	-14.80%
Search for Vulnerability & Exploit Database	32.80%	16.10%	-16.70%
Working with other team such as (Soc Team)	32.80%	9.70%	-23.10%

Fig. 3.2 Methods Used to Identify Threats

16.10% of the non-SOC team are searching in Vulnerability & Exploit Database, is a great risk since it could open a gateway for a hacker to access your network or system. Finally, getting a SOC and non-SOC team working together is a challenge which organizations need to overcome by setting up a daily report, weekly meeting to discuss the latest challenges related to cybersecurity threats. Figure 3.3 demonstrates the process of risk assessment, vulnerability scan, penetration testing knowledge for the SOC and non-SOC staffs. The survey result does not reflect good awareness among those employees. We have identified that only 31.9% of the SOC team members have good knowledge of penetration test result, whereas only 22.6% of non-SOC staffs are aware of that report. So, this implies that in the area of prevention and proactive response need to improve among the relevant departments who are responsible for mitigating cyber-attack.

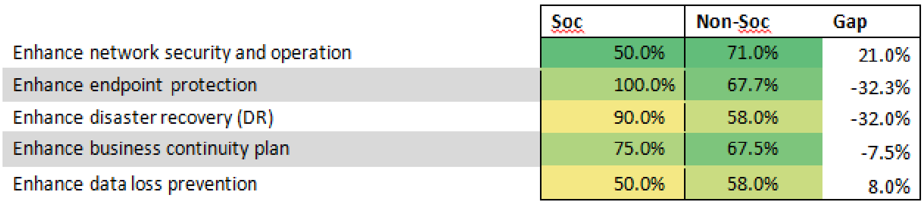
Figure 3.4 demonstrates the survey result of network security and protection, endpoint protection, disaster recovery, business continuity plan and data loss prevention. The result shows that both the SOC and non-SOC staffs have some level of knowledge of these areas of



	Soc	Non-Soc	Gap
Results of risk assessment	63.9%	77.4%	13.5%
Results of vulnerability scan	45.4%	35.5%	-9.9%
Reports from 3rd parties about increased cyber attacks	45.4%	32.3%	-13.1%
Results of penetration tester's report	31.9%	22.6%	-9.3%

Fig. 3.3 Prevention and Proactive Responses

enhancements. On the other hand, in some areas such as network and security operation and data loss prevention, knowledge of SOC is about 50%, which is very low for the team.



	Soc	Non-Soc	Gap
Enhance network security and operation	50.0%	71.0%	21.0%
Enhance endpoint protection	100.0%	67.7%	-32.3%
Enhance disaster recovery (DR)	90.0%	58.0%	-32.0%
Enhance business continuity plan	75.0%	67.5%	-7.5%
Enhance data loss prevention	50.0%	58.0%	8.0%

Fig. 3.4 Areas of Enhancement

The result shows that there is a considerable gap in knowledge among those employees regarding the cyber attack. Each of the IT team has a different level of knowledge about the cyber attack. For example, generally, the SOC team has got more knowledge of cyber-attack than the non-SOC team. It has been noticed that in the case of multi-factor authentication, both SOC and Non-SOC team scored very low as 41.2% and 32.3% as a demonstration of knowledge. Software patching is one of the essential factors for reducing cyber threat and vulnerability. Software patch should be updated on the time of the released by the vendor. Also, it is crucial for IT security team to understand patch and vulnerability updates. From the survey, it has been noticed that only 61.3% of SOC staffs know patch and vulnerability management.

3.5 Security Assessment Services

From the above survey result, it has been noticed that there is a huge gap in knowledge and awareness among IT employees within an organisation. It is important that organisation take necessary steps to build knowledge and awareness about cyber security among the employees. So, we propose some common assessment scenarios that aim to support organisations to keep up to date with cyber security knowledge.

This section describes the of the most common assessment scenarios. These can be customized in many ways to meet organisation's needs. Each type of assessment takes varying amounts of time and is impacted by the number of targets (applications, servers, networks, etc.). The exact type of assessment should be determined in the initial meeting.

3.5.1 Network-Based Attack & Prevention

Network-based attack always could cost a lot for an organisation. One of the prominent network-based attacks is the denial of service attack, which can be prevented by using design decision [69]. Organisations need to check the vulnerabilities of the network to prevent cyber-attacks. Penetration testing includes components of application vulnerability assessment, host vulnerability assessment, and security best practices. This type of test can be performed with or without detailed prior knowledge of the environment. When it is performed without prior knowledge additional steps will be taken to enumerate hosts and applications and to assess the ease with which any outsider could exploit publicly available information or social engineering to gain unauthorised access [70].

An attack and penetration test will answer questions like -

- How vulnerable is the network, host, and application(s) to attacks from the Internet or Intranet?
- Can an intruder obtain unauthorized access to critical resources?
- Are social engineering techniques effective?
- Are operational controls effective?

This would involve the Information Security Officer (ISO) acting as an attacker and looking at the system as an outsider. The ISO would look for -

- Remotely exploitable vulnerabilities
- Patch levels (OS and Apps)
- Unnecessary services
- Weakness of encryption
- Weakness of authentication

3.5.2 Host Based Assessment

This is an assessment of the health and security of a given workstation or server. Automated scanning tools (e.g. Nessus[2]) are the primary vehicle for this type of assessment. An additional hands-on inspection may also be necessary to assess conformance to security best practice. This assessment will answer questions like:

- Is patching up to date?
- Are unnecessary services running?
- Are anti-virus/anti-malware signatures up to date?

This would involve the Information Security Officer acting as a System Administrator and auditing the system and applications looking for -

- Locally exploitable vulnerabilities
- Patch levels (OS and Apps)
- Access rights
- Security best practices

[2] https://www.tenable.com/

3.5.3 Application

This is an assessment of the functionality and resilience of the compiled application to known threats. This assessment focuses on the compiled and installed elements of the entire system, e.g., how the application components are deployed, communicate or otherwise interact with both the user and server environments. Application scanning tools, as well as manual testing with and without application credentials, are used to perform this assessment. Typically some host, network, and general information security practices are assessed as part of an application vulnerability assessment.

This assessment will answer questions like -

- Does the application expose the underlying servers and software to attack
- Can a malicious user access, modify, or destroy data or services within the system

This would involve the Information Security Officer auditing an application (typically web based) and looking for vulnerabilities like -

- SQL Injection
- Cross Site Scripting
- Cross Site Request Forgery
- Improper data sanitization
- Buffer overflows (limited)
- Misconfigured or weak authentication

3.5.4 Compliance

This would involve the Information Security Office auditing (or assisting in the coordination of an audit if the ISO is not trained to conduct the specific audit) systems for compliance with specific regulations:

- GDPR
- HIPAA - Health Information Portability and Accountability Act
- FERPA - Family Education Rights and Privacy Act
- GLBA - Gramm-Leach-Bliley Act
- PCI - Payment Card Industry

3.5.5 Physical Security Assessment

This assessment typically involves interviews with key staff, documentation review, and an on-site visit to assess appropriate physical and environmental controls for safeguarding computing resources.

This assessment will answer questions like -

- Are there appropriate physical access controls in place for securing servers and desktop machines
- Are appropriate environmental controls in place to sustain critical computing infrastructure
- Are systems left logged in while staff are away

3.6 Summary

We also recommend security assessment service to support the Information Security Officer. Security assessment service provides a guideline to prevent network, host, application and physical security. This is the combination of both human and machine, which need to follow rules and regulations. Also, they need to maintain the physical security of the network and an individual hosts. So, it is important to do periodic penetration testing across the network to find the vulnerability.

The result shows that IT employees need to improve their knowledge in many aspects of cyber threat. It indicates that there is a gap of knowledge between Security operation team and other IT expert which need to be narrowed. SOC team are generally capability of safeguarding from cybersecurity threats if they can identify it. The Methods Used to Identify Threats were mainly through monitoring tools, and less attention was given to Security threat report, vulnerability assessment, and communication and cooperation with the SOC and non-SOC team. Moreover, participants, including the SOC and non-SOC operate show insufficient knowledge on the practice of risk assessment, vulnerability assessment and penetration testing. Both participants agreed on the need for enhancing data loss prevention. The employees need to be considered for more training on the current state of cyber threat on the organisation. Moreover, the employees who handle personal data must be updated with GDPR and relevant data protection acts.

The following chapter discusses a phishing attack scenario and analyse this using well-known cyber-attack modelling techniques. Additionally, the goal is to identify how a victim's (employee) action can help the organisation to keep valuable asset secure.

Chapter 4

Cyber Attack Modelling

4.1 Introduction

Cyber-attack is a sensitive issue in the world of Internet security. Business organisations across the globe are providing enormous effort to secure their data. They are using various types of tools and techniques to keep the business running, while adversaries are trying to breach security and send malicious software such as botnets, viruses, trojans etc., to access valuable data. Every day the situation is getting worse because of new types of malware emerging to attack networks. It is essential to understand those attacks both before and after they happen in order to provide better security to our systems. Understanding attack models provide more insight into network vulnerability; which in turn can be used to protect the network from future attacks. In the cybersecurity world, it is difficult to predict a potential attack without understanding the vulnerability of the network. So, it is important to analyse the network to identify the top possible vulnerability list, which will give an intuitive idea to protect the network. Also, handling an ongoing attack poses a significant risk on the network and valuable data, where prompt action is necessary. Proper utilisation of attack modelling techniques provides advanced planning, which can be implemented rapidly during an ongoing attack event.

In this chapter, we introduce a new attack modelling technique, Object-based model. The aim is to analyse various types of existing attack modelling techniques to understand the vulnerability of the network; and the behaviour and goals of the adversary. The ultimate goal is to handle cyber-attack in efficient manner using three well-known cyber-attack modelling techniques as well as the Object-based model to evaluate and compare with those models.

4.2 Proposed Model - Object-Based Attack Model

In an organisation's network, each of the components such as a server, workstation, router, mobile phones, tablet computers, software, etc., devices are considered as an object. This model concerns the protection of valuable asset of the organisation by identifying critical threats associated with components of an enterprise environment. An object encapsulates attributes and behaviour.

- **Attributes** - Physical and logical characteristics of an object. For example, a computer has name, IP address, MAC address, geographic location and Operating systems. The attribute of an object could be represented as an exposer to the public or private network.
- **Behaviour** - The behaviour of an object is equivalent to the functionalities. An object may have some functionalities such as
  - Create: Object has the ability to create another object. For example, an application can create a database connection object.
  - Read: An object may have the ability to read a message from another object. A server must have the ability to read the request message.
  - Write: An object may have the ability to write. For example, accept or reject a request.

  - Delete: An object may delete another object. Antivirus software can quarantine malware.
  - Modify: An object may have the ability to modify another object. An administrator may change a user's or group policy.
  - Accept: An object may accept request from another object. A server may accept user request for a new connection.
  - Reject: An object may reject a request from another object. A server may accept user request for a new connection.
  - Connect: An object may have the ability to connect to another object. Two devices can be connected either physically or logically.
  - Request: An object may originate a request to another object. An attacker may request for a new connection.
  - Response: An object may have the ability to respond to a request. Server responses to the request from the clients.

- **Criticality** - Some objects may be important than others. For example, keeping the unused port open can be dangerous as it could invite attackers.
- **Attack Surface**- Attack surface of an object that an adversary can insert or retrieve data [25].

An object is a generic term as there could be two types of objects as illustrated in Figure 4.1, are described as below -

Direct Object: A direct object could be a physical or virtual device that is directly impacted by an attack. An attacker may get an IP address, which could be a workstation, server or any other device. The attacker can attack to that object directly using a port. So, a direct object is directly exposed to the attack surface.

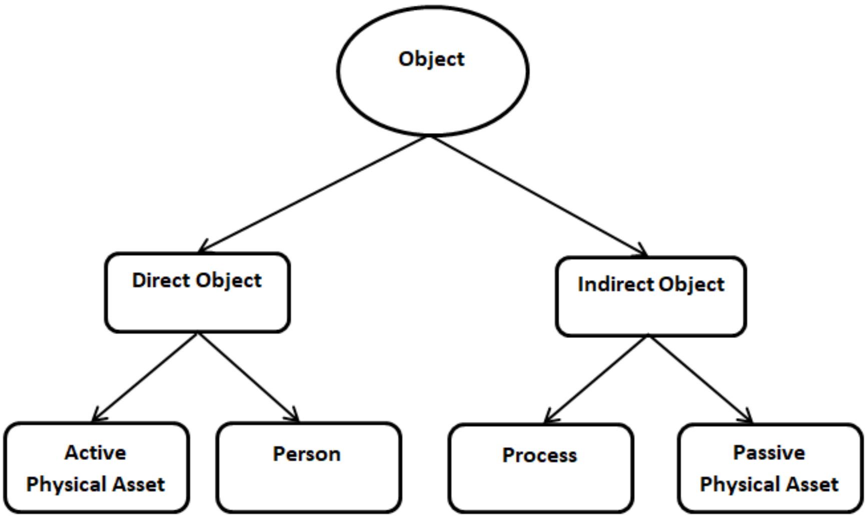


Fig. 4.1 Object-based Attack Model

Indirect Object: An indirect object, which is not directly in the attack surface, can be affected if the direct object is being attacked. For example, a server is a direct object. On the other hand, a payment processing system is an indirect object because it depends on the direct object such as a server. A person or a system could process the payment. In this context, the person is the direct object and the process is the indirect object since the process needed to be executed by the person.

This model considers three main aspects of the cyber security including -

1. **People** - A person is considered as a direct object, which can be exploited directly in an attack event. Generally, a person is an employee of an organisation who uses a computer for his/her day to day activities. As an object, a person has the attributes and behaviours. A person can have all the behaviour including read, write, delete, modify, accept, connect, request and response. So, a person could be a vulnerable object because he/she could be the reason for an attack. A person could be involved in the following activities or operations.

- Management - People are involved in management of hardware and software. For example, managing a router in an organisation is sensitive. A simple mistake could result severe consequence on the entire network.
- Configuration - The configuration of a hardware or a software product involves a person. For example, configuring a firewall is critical, i.e., for a simple configuration mistake by the System Administrator could open back door for adversaries.
- Phishing - Phishing attacks mainly depends on the human interruption to be successful. Emails are the main source of phishing attack to an organisation. For example, an adversary can send an email, which looks like send by someone from the company to make some transaction.
- Social Engineering - Social engineering is more advanced than the phishing attack, which also involves people (i.e., employee of a company) to gain access to valuable data.

2. **Process** - A process is an indirect object. When an attacker attacks a system, he/she needs to interact with a direct object such as human or computer software. Processes could be considered as the weakness of cyber security so, he following criteria should be considered to secure processes.

   - Improve simultaneous transmission of a system especially in payment process.
   - Process could be part of change management process.
   - There should be a recovery plan for the process.

3. **Technology** - It is considered as both a direct and an indirect object. In an organisation's network, each of the devices and software is considered as technology, which is the whole infrastructure of the organisations' network infrastructure. Technology could be vulnerable because of the following reasons -

- Update - Required to keep up to date all times.
- Unpatched system - System must be patched as soon as they release.
- Default Settings - Default settings must not be used.
- Vulnerabilities - Regular vulnerability analysis must be performed.
- Brute Force attack - System meta data must be protected along with business data from brute force attack.

4.2.1 Advantages of Object-Based Model

Advantage of this object based threat model are as follows -

- This model protects each object based on its real threat and does not worry so much about the strength of the attacker.
- This model protects attacks paths by analysing the attacks surface of indirect objects.
- Object-based model helps to understand the weakness of both direct and indirect object.
- People are given more emphasise as they are one of the important direct object.

4.2.2 Disadvantages of Object-Based Model

Disadvantages of the object-based model are as follows -

- This model require a serious threat analysis surface which can be a lengthy process.
- This model require attack data analysis for attack presentation.
- People, process and technology must coordinate with each other to be successful with this model.

4.3 Case Study - Cyber-Attack Analysis

In this section, we use a case study to analyse these three well-known attack modelling techniques. The goal of this analysis is to identify how each of the attack models behaves when encountering a cyber-attack. The case study has been written in the light of a real phishing attack scenario.

The case study involves a victim, who works for a company; and a regular user of a computer. The victim receives an email which has an attachment of a Portable Executable (PE) file with an extension of '.exe'. The email was sent to the company mail with a professional subject line. The sender email has the feeling of a legitimate email address and gives a sense of urgency. The email require the user to download and execute the file in urgent basis as this is the important update of the system. Moreover, the email emphasise that the update is important otherwise it could cause potential cyber threat. So, the user downloaded and executed the file.

In the mean time, the phishing alert was reported within the company about the same email received by other employees. So, the victim reported the attacked to the security department. The company's security team has identified an intrusion on the employee workstation or personal computer, which could be a potential security threat to the organisation's network as a whole. The workstation has been isolated from the network and has invoked action from a cyber incident expert to investigated the case. The investigator uses different types of attack model techniques to model the attack to enable them to prevent or detect future attacks.

In the following sections, we have analysed this scenario using attack modelling techniques called Diamond Model, Kill Chain and Attack Graph. Also, we proposed a new model and analyse this scenario, which emphasis on the human as an important actor. We compare the results, evaluate the scenario in various view points. Finally, we suggest a list of criteria that could be useful to model cyber-attack modelling techniques in protecting and mitigating cyber-attack.

4.3.1 Case Analysis with Diamond Model

In the Diamond Model, four main components are the adversary; capability; infrastructure; and the victim. The above attack scenario initially identified two of the components of the attack are the adversary and victim. We analyse each of the main components for this case by asking some questions. These questions will be answered as we investigate the case using this model. The investigator uses different types of attack model techniques to model the attack to enable them to prevent or detect future attacks.

Victim:

1. *What was the profile of the person being attacked? (Is it a general attack or is it a targeted attack?)*
   It is important to identify the profile of the person attacked, which gives a general idea of the attack whether the person is a general member of the public or someone perceived as significant. A targeted attack can use an individual who is a general worker to get to more high profile individuals or the information needed.

2. *Is the detail of the person being attacked publicly available (e.g. via the Internet)?*
   We may like to know why this person is being attacked? Are any details online and accessible to the general public? Is there any sensitive information online to attract an attacker? The information, mainly the email address of the person, is not in the public domain.

3. *What was attacked? (Email or anything else)*
   It could give an initial idea of why the attack had taken place by identifying what was attacked. The attacker could attack the victim's email, personal data, bank details, etc., which could be the initial step of his/her attack. The attack used the corporate email, which is easy to guess or found in the public domain.

4. *Was the person attacked at work, i.e., inside the corporate network?*

   The attack took place in the corporate network as the victim has opened the email in the corporate network. Generally, a corporate network is more secure than a private one. Also, it is assumed that the work email will be accessed at work. So, this gives more opportunity to an attacker to attack a corporate network.

Capability:

1. *Did the attacker use any Malware?*

   A Malware attack could be dangerous for any corporate or personal network. Notably, a Malware attack could ruin the company's reputation or even close the business. The attacker used a malware through a .exe file in a Windows 2007 computer.

2. *If yes, was the malware trying to exploit a vulnerability?*

   There are thousands of different types of malware in the cyber world. If we can identify, the nature of the malware, it could give an idea about the goal of the attack or the motivation of the attacker [5]. The payload is a portable executable file, which is commonly used within a Windows PC environment.

3. *How sophisticated is the malware?*

   Since '.exe' files are Windows operating system's native executable file; the user usually does not mistrust the file. Generally, it is dangerous as anyone can execute such a file within a computer with little or no user authentication.

4. *Did the malware requires human intervention?*

   The malware that requires human intervention; e.g., the user needs to click and allow an install or to execute the malware or the malware may come with an email attachment; and the user needs to click to open it and then the malware executes. A '.exe' file needs user intervention to be executed. Also, the user must have some privileges on the system to execute such files.

5. *Did the attacker uses a known hacking tool?*

   Attackers usually use hacking tools to originate an attack. It is important to know if the attacker used any known hacking tool to model or understand the attack. It is unknown if the Adversary has used any hacking tool. In this case, it is possible that the attacker has used a hacking tool to create malware. It is possible that the attacker has another hacking tool to use for command and control if the installation is entirely successful.

6. *Did the attack use stolen credentials?*

   It is common to steal credentials before initiating any cyber-attack. The attacker used only one email address, which could or could not be stolen.

7. *Is that malware part of a campaign?* No, this malware was not a part of a campaign, it is just an executable which could potentially harm a corporate network to steal data.

Infrastructure:

1. *What IP was used?*

   It is important to identify the IP address of the attacker, which will tell a story about the attack.

   The IP address used in this attack is 191.234.4.50:80 and 176.31.128.232:1443. We can also see a port 80 is used, which indicates an email address. On the other hand, port 1443 is one of the main Transmission Control Protocols (TCPs), which is used to transmit the file.

2. *Can the IP be traceable? (Was it a VPN, proxy, etc.)?*

   The IP needed to be traced to find the source of the attack. The IP that has been used and can be located using a simple trace command in the Windows command prompt.

3. *How many IPs were involved?*

   There were two IP addresses used in this attack event.

4. *Was a domain name used?*

   Yes, an email must contain a domain name. So, the domain name used in the email was not any legitimate domain.

5. *Can we track that domain name?*

   Yes, we can track the domain name.

6. *Can we establish a history for the domain name?*

   Yes, domain history can be identified from the attacker's email address.

7. *Can we find an email related to the attacker?*

   The email address of the attacker was found in the incoming email to the victim. The email address helps to identify the domain name and other relevant details of the attacker.

8. *Are they using a botnet?*

   No, they were not using any botnet; it was a .exe file for the Windows operating systems.

9. *Are we able to find a command and control?*

   The attack was identified and appropriate actions have been taken in the early stage. The attacker did not get much time to use any command and control for this attack. Usually, command and control occur after the malware been installed in the system. In this case, the action has been taken just after the installation of the executable file.

10. *Is there anything in the malware sample that will tell us about the infrastructure?*

    The malware comes with many details; e.g. the malware was a '.exe' file, which is a Windows-based executable file. To create an executable file such as '.exe' requires a

Windows operating system. So, it is clear that the attacker has used a Windows-based infrastructure to make this attack. Since the attacker has used two IP addresses, it can be assumed that the attacker has a well-structured infrastructure.

Adversary:

1. *Can we identify an email from the attacker?*

   To identify an email from an adversary is not an easy task for regular users. The user trusts the email is coming from a legitimate email address.

2. *What physical infrastructure was being used (hardware, operating system, Software version)?*

   It is a challenging task to identify what kind of infrastructure has been used by the attacker as we cannot physically see any of his activities. Despite this barrier, some components of the infrastructure can be worked out, such as a .exe is generated by Windows Operating Systems, TCP is the standard protocol for most Internet communications.

3. *What sort of network was being used (Wi-Fi, private, public servers online)?*

   It could not be identified what kind of network has been used by the attacker during the attack.

Using the Diamond Model, we have identified several insights into the case study. It is essential to ask several questions and try to find answers during the investigation. This process gives an obvious idea of the attack. In the previous section, we have a number of questions to investigate the case, which essentially gave more in-depth insight. In this section, we discuss the findings of the investigation for each of the attack modelling techniques.

In this case study, the adversary has sent an email to an employee of a company. The email had an attached .exe file and sent to attack the victim's computer or the network. The

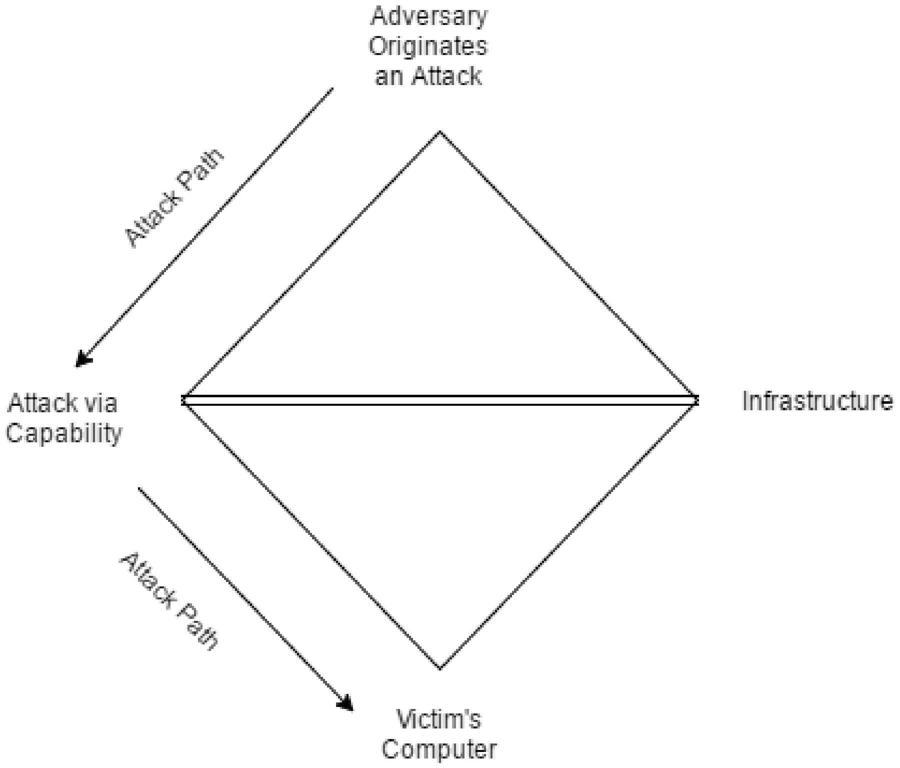


Fig. 4.2 Attack Path with in the Diamond Model

Diamond Model allowed us to identify the structure of the attack. For example, the attacker has used the victim's email address and sent an email, which looks like a legitimate one. It could be suggested that the infrastructures are the same for both the victim and the adversary; however, their capabilities differ substantially. The adversary has taken this opportunity to attack an employee of the company, not an administrator. A network administrator has knowledge in identifying '.exe' files from an unknown source, which indicates that the adversary has done some research on the company or the employee and acquired the knowledge of their infrastructure and capability. So, it is clear that the attacker has used the capability measure of the Diamond attack model to manipulate the victim. In Figure 4.2 illustrates the general view of the attack using the Diamond Model.

However, Diamond Model can be extended to analysis this case as shown in Figure 4.3 that an attacker mainly targets to have financial gain. So, they use infrastructure and

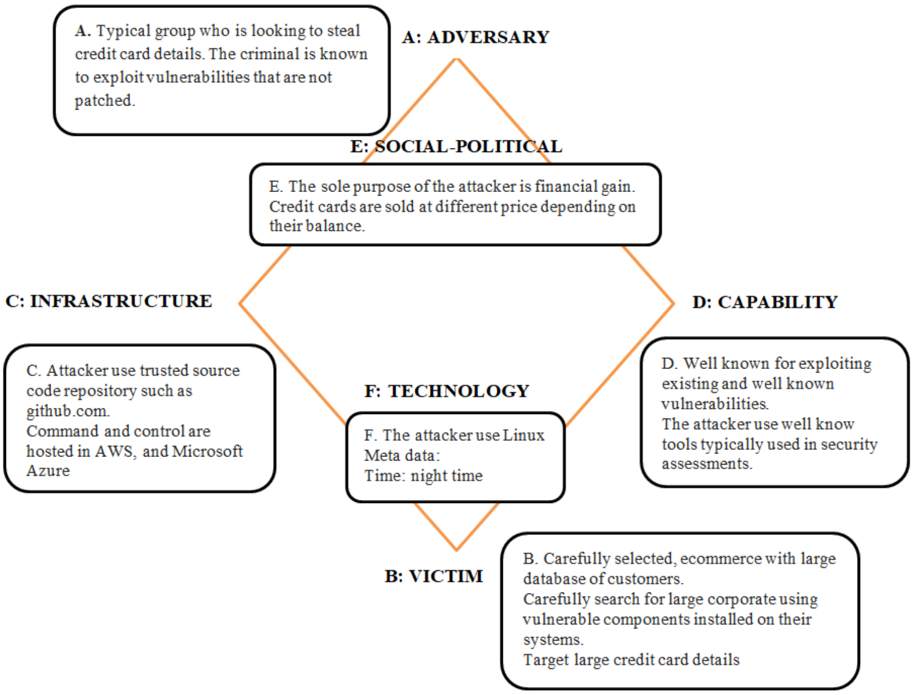

A. Typical group who is looking to steal credit card details. The criminal is known to exploit vulnerabilities that are not patched.
A: ADVERSARY
E: SOCIAL-POLITICAL
E. The sole purpose of the attacker is financial gain. Credit cards are sold at different price depending on their balance.
C: INFRASTRUCTURE
D: CAPABILITY
C. Attacker use trusted source code repository such as github.com. Command and control are hosted in AWS, and Microsoft Azure
F: TECHNOLOGY
F. The attacker use Linux Meta data: Time: night time
D. Well known for exploiting existing and well known vulnerabilities. The attacker use well know tools typically used in security assessments.
B: VICTIM
B. Carefully selected, ecommerce with large database of customers. Carefully search for large corporate using vulnerable components installed on their systems. Target large credit card details

Fig. 4.3 Attack Analysis with Diamond Model

capability to attack victim. However, the attacker also use social and political benefit by stealing credit cards. They also use technological advantages such as they use Unix operating systems, meta data, time of the day etc., to get action on target.

4.3.2 Case Analysis with the Kill Chain

In this section, we analyse an attack using the Kill Chain to understand more about the attack. As previously stated, the Kill Chain has seven steps, i.e., the attacker needs to go through those steps to achieve their goals. In this case study, the attacker did not attack the victim directly without doing any research. An attacker in Kill Chain model has to go through the user where user need to take an appropriate decision as shown in Figure 4.4 In the following we have used the earlier case study to analyse the Kill Chain -

Step 1 Reconnaissance: The attacker has gathered lots of information from the Internet and other media in which he or she can reach the goal. It is possible that the attacker's

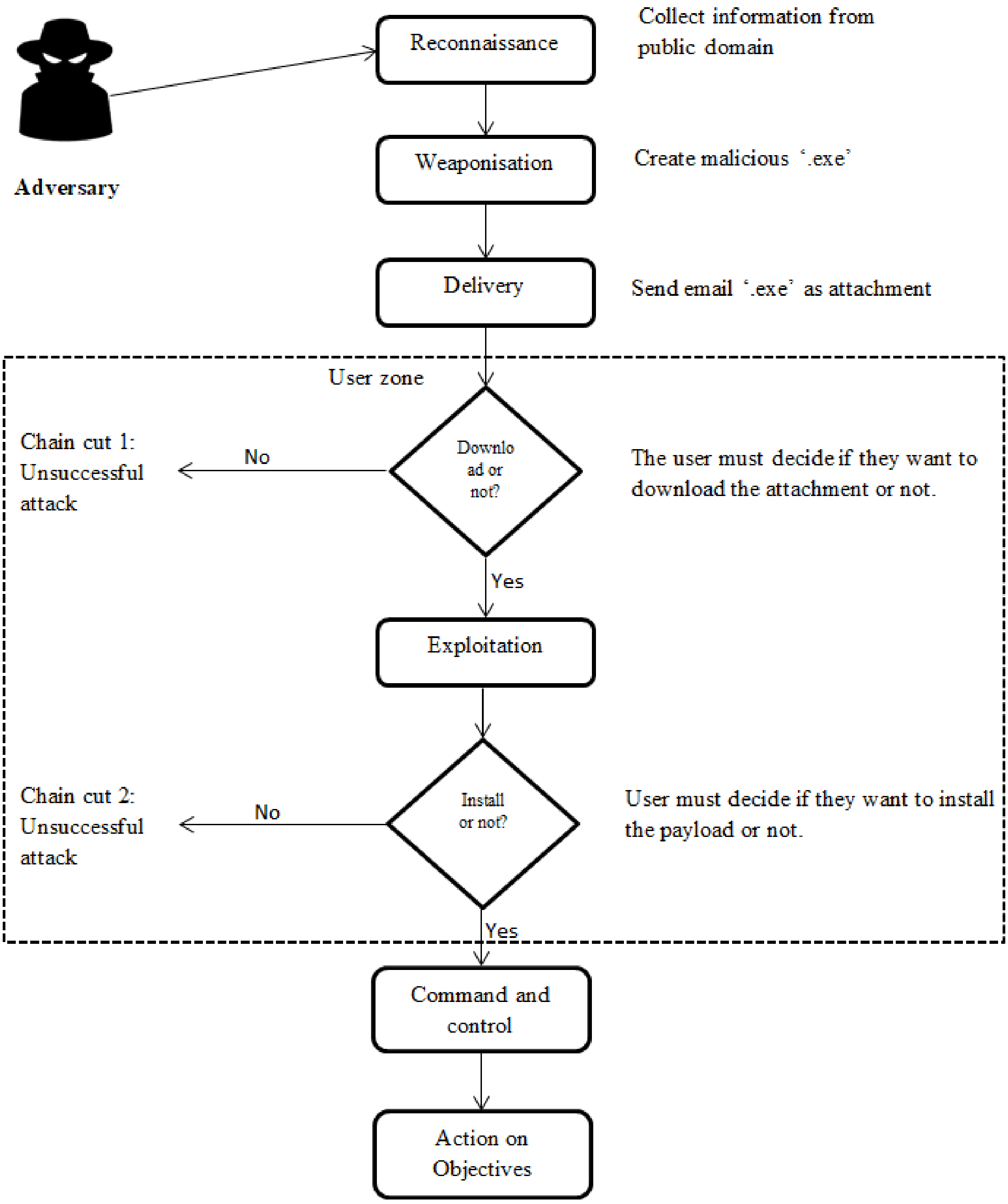

Adversary
Reconnaissance
Collect information from public domain
Weaponisation
Create malicious '.exe'
Delivery
Send email '.exe' as attachment
User zone
Download or not?
No
Chain cut 1: Unsuccessful attack
The user must decide if they want to download the attachment or not.
Yes
Exploitation
Install or not?
No
Chain cut 2: Unsuccessful attack
User must decide if they want to install the payload or not.
Yes
Command and control
Action on Objectives

Fig. 4.4 Case analysis with Kill Chain

primary goal is to access information about the company. One to achieve this is to collect information about an employee of the company. We can say that the attacker has collected an employee email address as an entry point.

It is almost impossible to detect the reconnaissance, but worth understanding by the defender before becoming a victim. By understanding this phase, it will be easier to protect the network from any attack.

Step 2 Weaponisation: The attacker creates a malicious payload to send to the victim. The payload is mainly the '.exe' file, which is a PE file. A PE can be executed in Windows operating systems just by clicking. So, this is an easy way to weaponise for the attack.

This is the phase that is difficult to recognise by the victim. Although it is not recognisable, it can be identified by the malware analysis or other investigation. Detecting the weaponiser can play an important role in securing a network or understanding a cyber-attack.

Step 3 Delivery: An attacker sends the malicious payload to the victim using some means of communication, which in this case is an email. So, the attacker has sent PE through email the victim may execute the file.

This is the phase in which the victim has the opportunity to protect themselves from any potential attack. In the case under study, the victim receives the file, which is not under the control of the attacker anymore. It depends entirely on the victim if they want to execute it or not. If the victim does not execute the file (PE), the adversary will not be able to move on to the next step for exploitation.

Step 4 Exploitation: In this stage, the actual exploitation happens. The file has been downloaded onto the victim's computer and is ready to be installed. After downloading a file in a computer, the file stays in the computer's memory. In most case, it requires human intervention to execute the file. However, most of the operating systems notify if the user want to install the particular software.

Step 5 Installation: The file has been installed by the victim, which makes the attacker's job easy. In this step, the victim should be more careful before installing any unwanted software. Modern Windows operating systems always ask before installing any software from an unknown source. So, to prevent any cyber-attack, the victim should check the credentials of the software.

Step 6 Command and control: Through the installed malware, the attacker creates a command and control channel to access the internal asset of the victim. The attacker may get control over the DNS, websites, social networks or other methods. The command may gather information from the infected computer or its network. The method of data collection could be keystroke monitoring, password cracking, screen capture, sneak into valuable documents etc., to capture valuable information.

Step 7 Action on objectives: The attacker achieves their goal on the victim's infected computer or network.

From the victim's side, it is not possible to see how the attacker attempts an attack. It could only be anticipated from the Kill Chain method that the attacker has taken those steps. If we go reverse order to the chain from the attack, it could become clear that the attacker took time to research before making any attack. The attacker also tries different types of methods such as social engineering by sending an email. The attack could be cut off from Step 3, which is Delivery by understanding the malicious activities of an attacker. If the attacker managed to go to Step 7, which is the goal of the attack, they could do whatever they plan.

Once the attack is recognised, it is possible to understand all the steps of the attack and find the pattern of such attacks.

4.3.3 Case Analysis with Attack Graph

In the Attack Graph technique, the attacker can attack using many different paths to get control of the victim's network. The attacker could be anywhere on the Internet when originating an attack. In our case study, the attacker attacks using social engineering techniques to get access to the victim's computer. The Attack Graph for the attack scenario can be summarised as below -

- Adversary stays on the Internet and creates a PE. The file has been designed to attack a particular network. The file contains the goal or the motive of the attacker, including the attack information.
- The adversary researches online into the organisation to be attacked. The information gathered is available in the public domain. The adversary usually takes time to get information for making any valid attack.
- The adversary identifies an employee email address as an entry point. An attacker always needs an entry point that will help in achieving their goal.
- An email is sent to the employee with a PE with an assumption that the employee will open and download the file, which is a plan of the adversary, which needs victims intervention to be successful.
- If the employee executes the PE, the adversary can exploit the workstation of the employee. This exploitation could get more information about the network as well as the web server or other connected devices,(e.g. flash drives), attached to the workstation.

- Gradually the attack can be extended to the ultimate goal such as the manipulation of a database server using the web server or a DDoS attack.

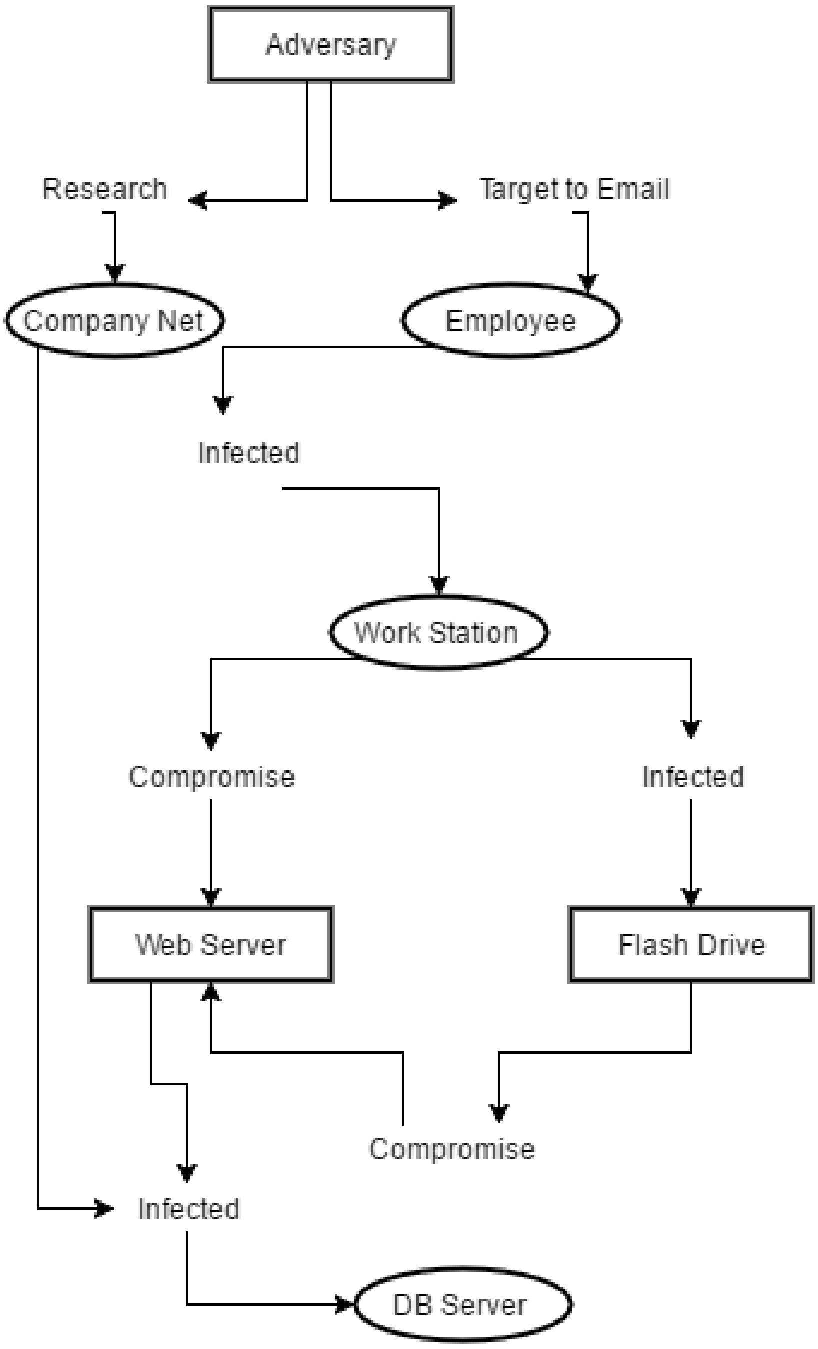


Fig. 4.5 Analysing Cyber-attack with Attack Graph Technique

In general, to model an attack, the Attack Graph technique requires the deployment of a graph that shows many attack paths. These paths could hold information that can be used to prevent future cyber-attacks within a high-profile network. In figure 4.5, we have modelled the attack for our case study, in which we can see a number of attack paths to reach the victim's database server. There could be a number of paths from the adversary to the victim, and it depends on how they secure the victim's infrastructure. This path is very challenging as the company security system may not be compromised and an alert could be generated so that the cybersecurity team can increase the security to prevent a potential cyber-attack. The attacker has selected the other path, which is via an employee. In this path, the attacker has to go through many steps, once in the company network, but gives more probability to success on attack goal. Analysing each of the paths could lead to the vulnerabilities of any organisation's network. So, in this method, we have identified the victim employee worked as a gateway, which helps the adversary to proceed further with the attack.

4.3.4 Case Analysis with Object-Based Model

In the Object-based model, there are three main components including people, process and technology. We analyse the attack scenario using the object model to understand how the attack can be modelled. In Figure 4.6, we illustrate the case analysis using the object-based model, which shows how three main components people, process and technology can model the attack event.

People: An employee (people) could be a victim or a door to the attack in this model. The attacker has attacked a worker to get access to the system of interest, such as a database server or web server. The attacker gets the email address of the worker, which is not in the public domain. The attacker used the email of the victim, which is a corporate email address. The attack has happened to the person's corporate network. He/she has opened the email assuming that this is from a trusted source. Although the network system is secure, it is hard

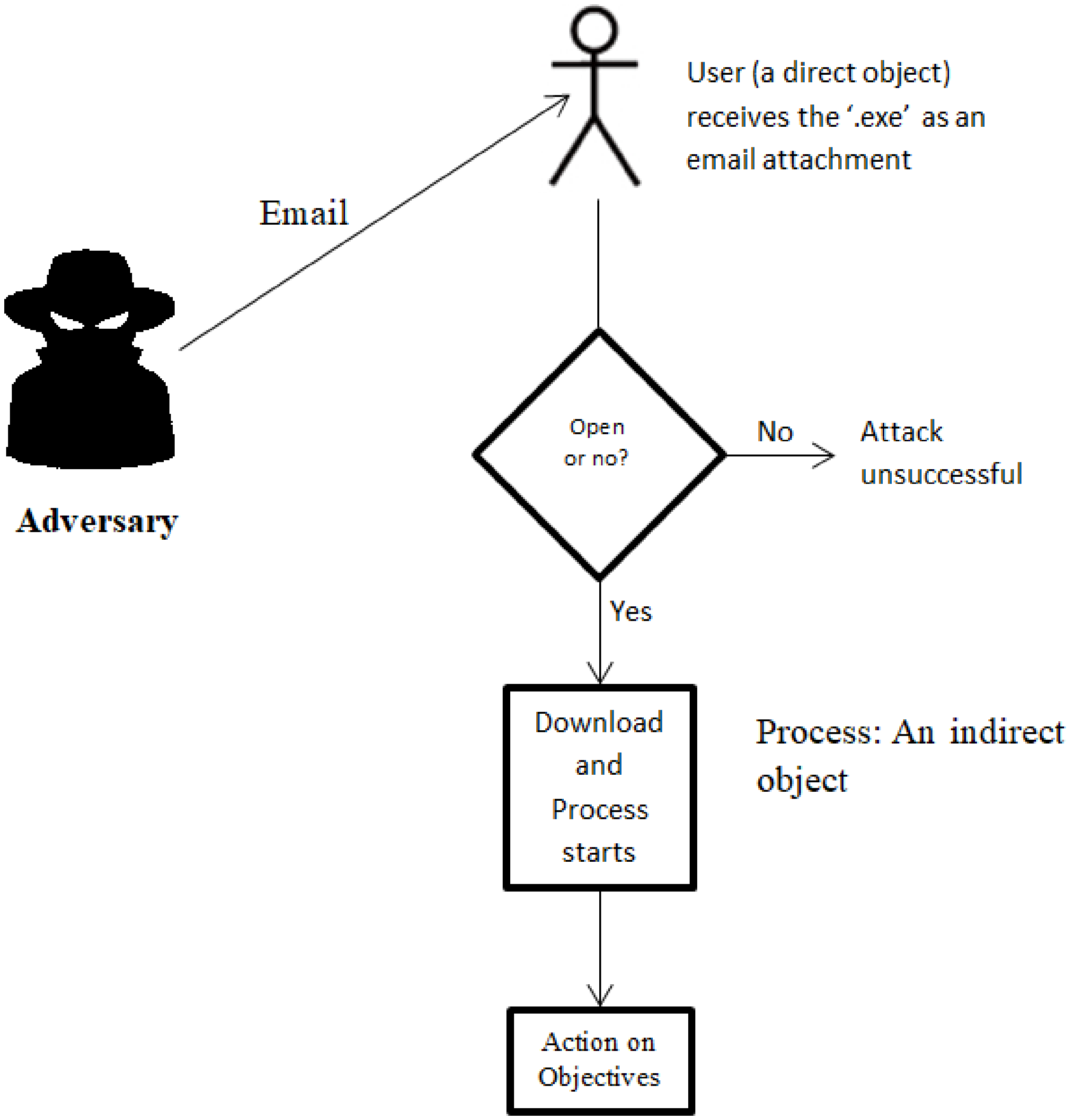


Fig. 4.6 Case analysis with object-based model

to attack directly. So, the attacker sent the '.exe' file to the person. Now it depends on the person whether to open or not.

Process: During the attack event, in the phase the recipient of the '.exe' file has the option either to download or not. If the file is download and clicks to execute the exploit starts, which could be malware installation, stealing data, ransom and other activities. Firstly, it is essential to know if the file is malware. If it is a malware, it is hard to know what type of malware. However, this is a '.exe', it is Windows based executable file, which can only be executed in Windows platform. On the other hand, the user is using a computer with Windows Operating systems, which could pose a danger if the file is being executed.

Technology: In the object-based model technology includes all the hardware and software. In technology, there are direct and indirect objects. For instance, an email address is a direct address since the attacker can send an email directly to any address. However, to execute a file, it requires another direct object, which is a person. In this, the process can be executed, which is an indirect object, requires a direct object (a person) to run.

The attacker could use a malware tool to create the '.exe' file. The technology they used and targeted are Windows Operating systems.

4.4 Discussion

Using these four models to analyse the same cyber-attack is useful. Each of the methods is unique and presents the same attack in very different ways. In this section, we discuss the insight achieved following this analysis using three attack modelling techniques.

The Diamond Model has only four components with several subcomponents, which shows that the adversary has more capability than the victim. The victim has executed the PE on the machine to allow the adversary to achieve the goal. On the other hand, the adversary is not as strong as the PE in this aspect, as the file cannot be executed without any user intervention. The attack could be prevented if the victim could identify the executable file as malicious earlier. The victim's computer is a part of the victim's infrastructure, which may be as stronger as the adversary's one; however, the knowledge of the victim played as a weak point in this situation. So, in this case, it is clear that the victim may have the same strength in the infrastructure as the attacker but weaker in capability.

The Kill Chain method gives more details of the attack using seven steps. These steps are like a chain, which can be broken from the victim's side as soon as an incident is reported.

The victim can not control the first three steps of a Kill Chain. From step 3, the victim can kill the chain if a threat is recognised from its payload. The attackers motive can be identified by analysing the malware. On the other hand, reverse engineering the Kill Chain could help the security team to identify the behaviour and motivation of the attacker. These indications could help the network to be more secure by finding the weak points.

The Attack Graph is generated to identify the vulnerability of the system, understand how an attacked happened and to set actions to prevent further attacks. The Attack Graph gives multiple paths that an attack has occurred in any network. This technique can also help to identify potential threats by analysing network vulnerability. Analysing network vulnerability using an Attack Graph technique is expensive as generating a graph is difficult. There are many tools available for creating a graph. In our case analysis, we have illustrated a graph using the attack scenario. This graph shows the current path of the attack as well as relevant branches that could be infected.

Finally, the Object-based model allows identifying a cyber-attack in the object level. This model provides an opportunity to understand the attack surface of an organisation's network to take appropriate action. For example, in the object-based model, 'people' are the direct object, who play a vital role in the defence mechanism. In this case study scenario, an attacker targeted a human and sent the payload as '.exe' file. The human user has two choices: a) execute, or b) not execute. On the other hand, a workstation could work as a direct object, which required to be secured using the software update, patching, and antivirus software. However, in an organisation situation, it is hard to stop attachment to be received. So, the employee (a person) need to be aware of the threat of executing a '.exe' file. In this context, regular training on the cyber threat is vital to all the employees within an organisation.

Although the attack modelling techniques are different from each other, they share some common attributes such as adversary, victim, network, attack plan, create payload, delivery of the payload installation and execution. One of the important parts of a cyber-attack is that the attacker does not attack any high-profile network without doing proper research. The attacker plans before making an attack, which is done using collected data on victim's infrastructure and capability. To make an attack the attacker needs to send a payload, in our case the '.exe' file, which is a ubiquitous way of attacking any Windows workstation. Also, a social engineering method is used to deliver the payload to the victim. So, the chance of success is fifty-fifty, because once the file is delivered, it is upon the victim whether to execute or not. The Attack Graph shows if the attacker makes a successful attack, there will be more alternative ways will be opened for making further attack such as DDoS. On the other hand, the object-based model identifies a direct object (user) plays the role of a gatekeeper in this situation.

4.5 Evaluation

This section evaluates the attack modelling techniques using some set criteria. These Criteria has been identified from the analysis of some case studies using these attack modelling techniques. During the analysis, we have identified a number of questions to model the case study, which gave detailed insights of the cyber-attack. The detailed discussion also indicates some of the evaluation criteria. In the following section, we discuss the identified criteria, also illustrated in Table 4.1. We also discuss the support that is offered by these attack modelling techniques in the event of an attack.

1. **Attack Presentation**

   It is important to represent the attack promptly using any of the attack modelling techniques both before and after an attack happens, which will give a general view of the attack and will help the investigation team to take necessary actions.

   The presentation of the attack using the Diamond Model is relatively easy it comprises of only four main components called the Adversary, Capability, Infrastructure and

Table 4.1 Attack Modelling Criteria

Criteria	**Diamond Model**	**Kill Chain**	**Attack Graph**	**Object-Based**
Attack Presentation	Easy	Easy	Hard	Hard
Attack Analysis	Medium	Medium	Medium	Medium
Ease of Modelling	Easy	Easy	Hard	Easy
Identify vulnerability	Hard	Hard	Easy	Easy
Prevent attack	*X*	*X*	*X*	Medium
Support Attack Handling	Easy	Easy	Medium	Easy
Identify Attacker	Medium	Medium	Easy	Medium
Protect Network	*X*	*X*	*X*	Medium
Recognise Threat	Easy	Hard	Medium	Easy

Victim, which need to be considered for a quick investigation. So the attacker used the capability of the Diamond Model to attack the victim. To make a successful attack, the attacker must carry out a detailed investigation into the target organisation and the infrastructure. In the Kill Chain method, the attacker also has to study about the capability and infrastructure of the victim in the reconnaissance phase to create a suitable payload. In the Attack Graph method, attackers look for the vulnerability of the victim's network, which does not explicitly present attack easily. Also, the Object-based model deals with the direct and indirect object, which mainly works on the defence level. However, the object-based model helps to analyse after the attack scenario by analysing objects being exploited.

To summarise, attack presentation using these four models give different views of a given attack situation.

2. **Attack Analysis**

   Analysing attack is one of the essential tasks that security professionals do after a cyber-attack event occurred. A proper analysis of an attack scenario ensures better protection to the network. So, an attack must be analysed and recorded and apply the knowledge to build a better protection system.

Diamond Model analyses the attack scenario using adversary, capability, infrastructure and victim. In this model, the victim's vulnerability is defined by the capability and infrastructure of an adversary. However, as detail analysis is done using the attack time-stamp, phase, result, direction, methodology and resources. A time-stamp indicates the time of an attack event as well as the attacker's activity in the victim's domain. A time-stamp explains each of the phases of an attack.

The Kill Chain model analyse is on the high-level components, which mainly depends on exploitation, installation, command and control, and action on objectives. In the Attack Graph model, the analysis occurs on the victim's side of the attack path, not on the detail level of the attacker's activities. On the other hand, the Object-based model analyses each of the objects, including the user to understand the attack event.

3. **Ease of Modelling**

   Cyber-attacks are very sensitive that must be tackled promptly after the attack happens to reduce the impact of the attack. It is essential to model the attack immediately after the attack to understand the nature and effects of the attack. So, the attack modelling techniques must ensure flexibility when following an attack event.

   From the above discussion, we can see that the simple Diamond Model consists of only four components that are very easy to use for quick modelling. If we need to model the attack further, we need to use the meta-features of the diamond model. The Kill Chain also gives a view of the chain of the attack. It also indicates the point where a chain can be killed, which potentially make an attack unsuccessful. On the other hand, the Attack Graph gives detail of the attack from the attacker node to the victim, which could be complicated as it can involve hundreds of nodes. So, the Attack Graph does not seem so efficient for quick modelling; instead, it can help to identify multiple paths of an attack. Modelling the object-based model depends on the attack surface, including both direct and indirect object. Analysing the direct object level is easy as this is the entry point of an attack. However, if the attack is so deep in the indirect object level such as effect several services or process, it would be hard to model such attack using the Object-based model.

4. **Identify vulnerability**

   Identifying vulnerabilities in a network are the best way of preventing any cyber-attack. A cyber-attack can be reduced if the network is continually monitored for any vulnerabilities. It is essential to check and identify vulnerabilities to ensure the network is safe and secure.

   For example, using the Diamond Model can indicate whether the victim has a weaker or stronger capability or infrastructure. The security could be improved by analysing the network, which can prevent many potential cyber-attacks. The Kill Chain also gives an overall idea of a cyber-attack, including how a chain could be broken. The goal of the Diamond Model and the Kill Chain models attack after it happens. On the other hand, the purpose of the Attack Graph is to find the vulnerability of the system in advance. Like the attack graph technique, using the Object-based model it is possible to detect the vulnerability of the network, which can be done simply by analysing both direct and indirect objects.

5. **Prevent attack**

   It is necessary to find an efficient way to prevent a cyber-attack. Prevention could be direct or indirect. A direct prevention measure is taken when an attack is on progress whereas an indirect prevent work as a precaution to stop a future attack. Attack modelling techniques should facilitate a way of preventing cyber-attack.

   Among these three models, the Kill Chain allows the victim to stop an attack in good time. For example, in our case study, the victim needs to identify the threat from the phishing email to stop any further damage to the network or in the workstation. Other attack modelling methods, such as the Diamond Model and the Attack Graph indicates any specific prevent point during a cyber-attack. People play a central role in the object-based model as they are the first line of the attack and prevent. The object-based model also ensures the direct object is secure and up to date with patches and versions.

6. **Support Attack Handling**

   Supporting the security team to handle an attack situation is challenging in a large organisation. It is crucial that all the attack modelling techniques provide attack handling support during or after a cyber-attack.

   All four attack modelling techniques are designed to attack handling. The Diamond Model and the Kill Chain are designed to handle an attack. An analysis using the Diamond Model, the investigation team can identify the nature of the attack and the Kill Chain analysis can give the state of the attack. The attacker may attack using capability or infrastructure measure, which can be identified by the Diamond Model. The attacker may send a payload to the victim and waiting for the installation, which can be identified by the Kill Chain analysis.

   On the other hand, the Attack Graph supports before the attack. The object-based model can support the attack both before and after the attack. For example, it can ensure that all the direct objects are in the standard states before any attack event. After an attack event, the object-based model analyses all the objects to find the root cause of the attack and support to handle them.

7. **Identify Attacker**

   In a cyber-attack event, it is difficult to identify an attacker; it requires a proper analysis of the data or could require reverse engineering, which will allow understanding the motive of the attacker. The motivation of the attacker gives an idea of what kind of data he or she is looking for. Understanding the attacker could help to provide appropriate security in the network. The Diamond Model generally tells about the attacker's attack approach. In the Kill Chain model, we can reverse engineer the chain to understand the attacker or their motive.

On the other hand, the Attack Graph gives more detail path of the attack, which leads to the attacker. It also identifies multiple vulnerabilities of the network, which leads to the attacker's motive. The object-based model requires enormous analysis of the affected objects to find artefacts such as IP address, left by the attackers to identify them.

8. **Protect Network**

   The most important task in cyber defence is well managed. A managed network provides more security than an unmanaged one. An attack modelling technique may not directly protect a network but could advise protection mechanism.

   None of the attack modelling techniques provides any security to the network but indicates vulnerabilities of the network. The Diamond Model and Kill Chain help to protect the network during an attack in progress. On the other hand, the Attack Graph and object-based model help protecting network both before and after an attack by identifying vulnerabilities from the attack surface.

9. **Recognise Threats**

   One of the critical aspects of cybersecurity is to recognise potential threats. If a threat is identified, it is easy to protect the network. An attack modelling technique should provide an option to recognise potential cyber threat by analysing the vulnerability of the network.

   The Diamond Model gives a simple and clear idea of threat recognition. For example, a simple analysis such as testing the capability or the infrastructure could provide an intuitive notion of the network. If the network is weaker in any of those criteria, there is a threat. A Kill Chain analysis could give an idea of the penetration point of the network. For example, the network needs continues attention for new types of threat, malware or any zero-day attack, which could be predicted using the Kill Chain. The Attack Graph analysis could find multiple attack paths, from which threats could be identified. The object-based model analyses all the relevant objects, including incoming traffic, to recognise future threats.

4.6 Summary

In this chapter, we have analysed an attack scenario using four attack modelling techniques called the Diamond Model, Kill Chain, Attack Graph and Object-based model. First three are established models that were used for modelling several cyber-attacks. The Object-based model is our proposed modelling technique, which mainly deals with objects within the network. We also compare the result to identify if one model is better than the other. We have determined that these attack modelling techniques are unique in operation; however, share common goals such as supporting attack situation.

We have also proposed several criteria for attack modelling techniques. These criteria could help organisations to handle cyber-attack efficiently and effectively. In this case analysis, we have identified that cyber-attacks can be modelled using different techniques. Each of the techniques gives interesting insights into a cyber-attack. For example, the Diamond Model identifies how and why an attack happens, as we can see that an attacker attacks a victim depending on two main attributes called infrastructure and capability. The attacker will attack a victim if the victim's capability or the infrastructure is weaker. An attack event will not be successful if the victim is stronger than the attacker in infrastructure and capability.

On the other hand, the Kill Chain technique gives detail steps of an attack. Although the attacker considers the infrastructure and capability before originating any offence, the victim gets some opportunity to kill the attack chain, if an attack is identified in any of the early stages. So, it is important that the victim is aware of the attack chain. The Attack Graph indicates that how many ways cyber-attacks can happen. An Attack Graph technique finds multiple paths that can lead to a cyber-attack on any company network infrastructure. Each of the attack paths can represent some vulnerability information. Finally, the object-based model takes a little different approach as it considers people as one of the main components. Also, the object-based model deals all the direct and indirect object that can be affected by a cyber-attack. The Attack Graph technique can help in securing a complex corporate network from a potential cyber-attack to mitigate network vulnerability.

Chapter 5

Cyber Threat Intelligence

5.1 Introduction

Cyber attacks are continuously growing and becoming major concerns for all types of organisations. Organisations are putting several protection measures in place including regular penetration tests, setup IDS and 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 [71]. 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 APT, are becoming a significant issue [72]. The attacker in an APT event does not launch an attack without conducting reconnaissance and planning the attack. They are trained adversaries and use sensitive tools and techniques to target confidential information from high profile victims [28]. 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. Researchers used honeypot technology to understand various types of attacks such as malware attack, botnet activity, phishing and spreading spam.

This chapter exploits elasticsearch technique to analyse honeypot attack data as it gives flexibility of searching on any sized data set. This chapter also proposes a new threat intelligence approach to understand the attack pattern and behaviour of an adversary. The threat intelligence technique is evaluated following the collection of data. This is achieved by deploying honeypots as cloud services to find cyber-attack related events through the analysis of this data using the elasticsearch technique. The results are promising, thus demonstrating that honeypot data analysis could be used in cyber threat intelligence to support traditional protection systems.

5.2 Threat Intelligence Analysis

Cyber threat intelligence is a process of searching for potential cyber threats in the network by analysing the relevant dataset. Data analysis could be performed by using existing, automated tools or be performed manually. In an organisation, cyber threat hunting maturity depends on the ability of data collection and analysis [11]. 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 [12]. Data also contains details of a cybersecurity incident that has happened. Analysing such data gives an indication that most security incidents do not occur as zero-day attacks [13], they are quite frequent and in most cases have patterns. Appropriate data collection and analysis could lead to many elements of Indicator of Compromise.

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 sub-section, we have defined the conceptual model and derived a formal definition of that proposed model.

5.2.1 Threat Intelligence

Threat intelligence is the actions taken by the organisations to look for an answer for questions such as who is trying to login to the system, what password they are using, what time of the day the event happens, what type of connection request is made (e.g. SSH, HTTP etc.), why they are trying to access the system and more. The intelligence can also be queries that can be 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. The following is the conceptual diagram of cyber-attack recognition. There are three main components:

- **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 is similar events happen repetitively. Cyber attack pattern is the combination of the attack event and the behaviour of the attacker over time.

An attack is a systematic approach by an attacker to gain access into a system, a network or a host. An attack is originated from an attacker on to a system, which can be recorded using data collection. The behaviour of the attacker can be identified from the data collected

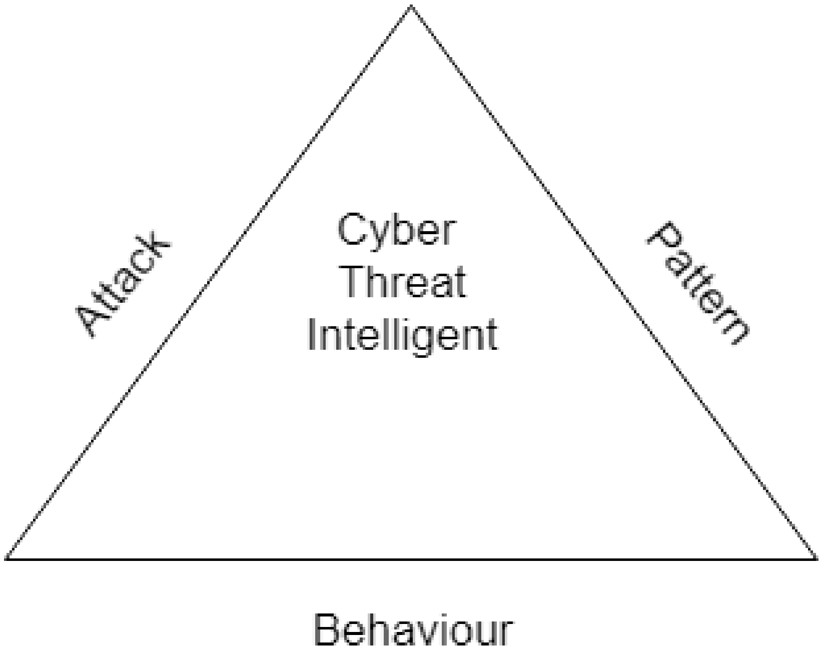


Fig. 5.1 Cyber-attack Concept

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 [73].

Generally, system log data collection is performed on most of the systems. These data can be considered as big data as the data has velocity, verity and volume [74]. If we think 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.

The main idea of the triangle in Figure 5.1 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 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 prevent future cyber-attacks.

We also illustrate a cyber threat intelligence system that allows analysing cyber threat data in real-time to generate an alert. In Figure 5.2 a new threat intelligence architecture is designed.

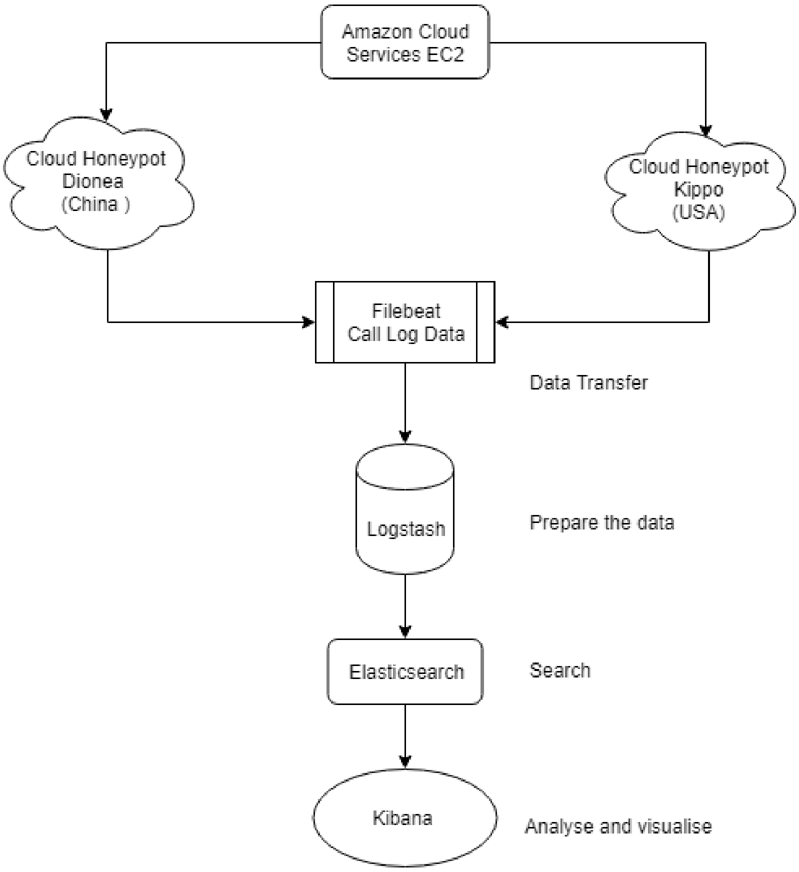


Fig. 5.2 Threat Intelligence Architecture

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.

5.2.2 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 $\{a_1, a_2, a_3, . . ., a_n\}$ 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.

An attack can be defined as a set of actions $\{a_1, a_2, a_3, . . ., a_n\}$ 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. 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.

However, if the attack is successful, the system is compromised and, the path 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 is 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 $X = \{X_r, X_p, X_m\}$, 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. The time-stamp starts from zero and lasts until the attack session remains.

Let us assume that in a cyber-incident event, an attack starts at time t and lasts for $\triangle t$. 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_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 advances 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 require '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 $Z = \{t_e((a_1 \times t_1), (a_2 \times t_2), (a_3 \times t_3), ..., (a_n \times t_n))\}$

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, domain name etc., 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, analyse them to identify attack pattern and behaviour of attackers.

5.3 Experimental Setup

This section describes the experimental setup using the Elastic stack, which consists of Elasticsearch, Logstash and Kibana knew as ELK[1] stack. The Elastic stack helps to present data, 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 any size of data and search is faster than other approaches. To support the Elastic stack

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

for data discovery, we used Filebeat to get multiple files to the elasticsearch. Figure 5.3 illustrates the architecture of the experiment.

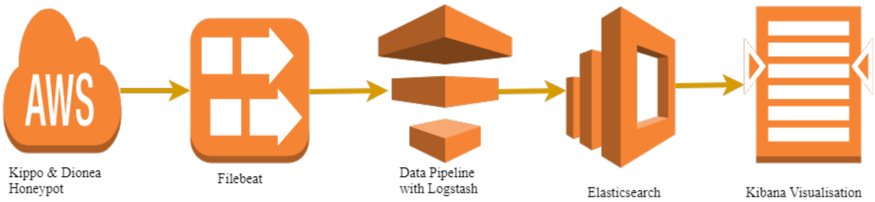
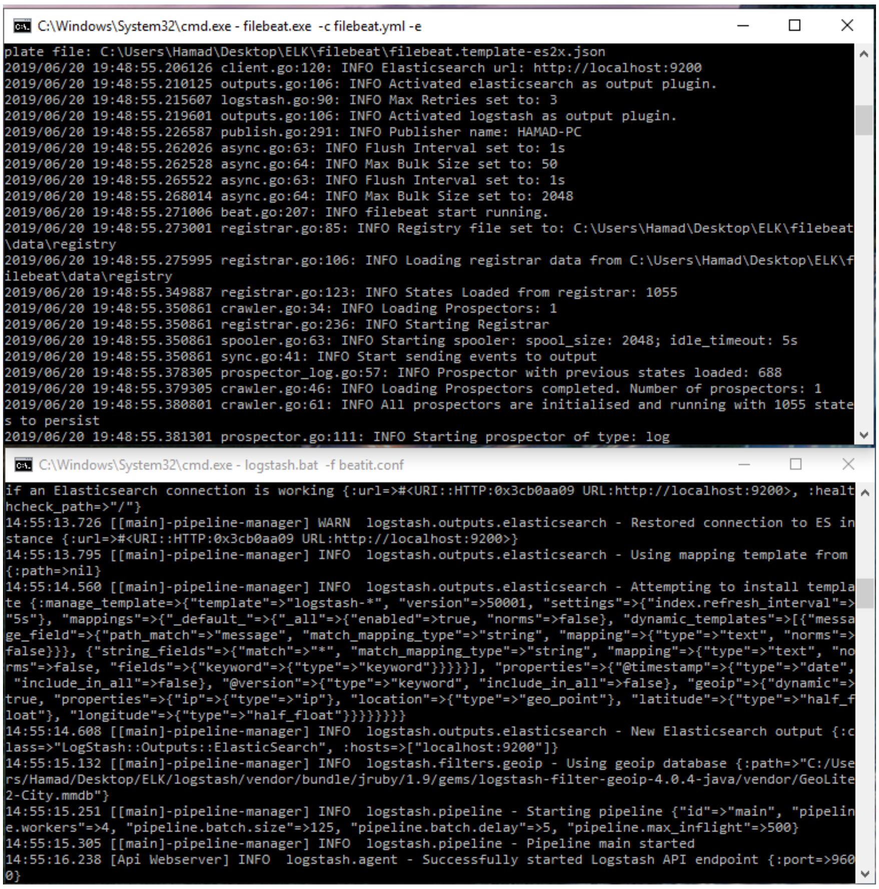


Fig. 5.3 Experiment Setup with ELK Stack

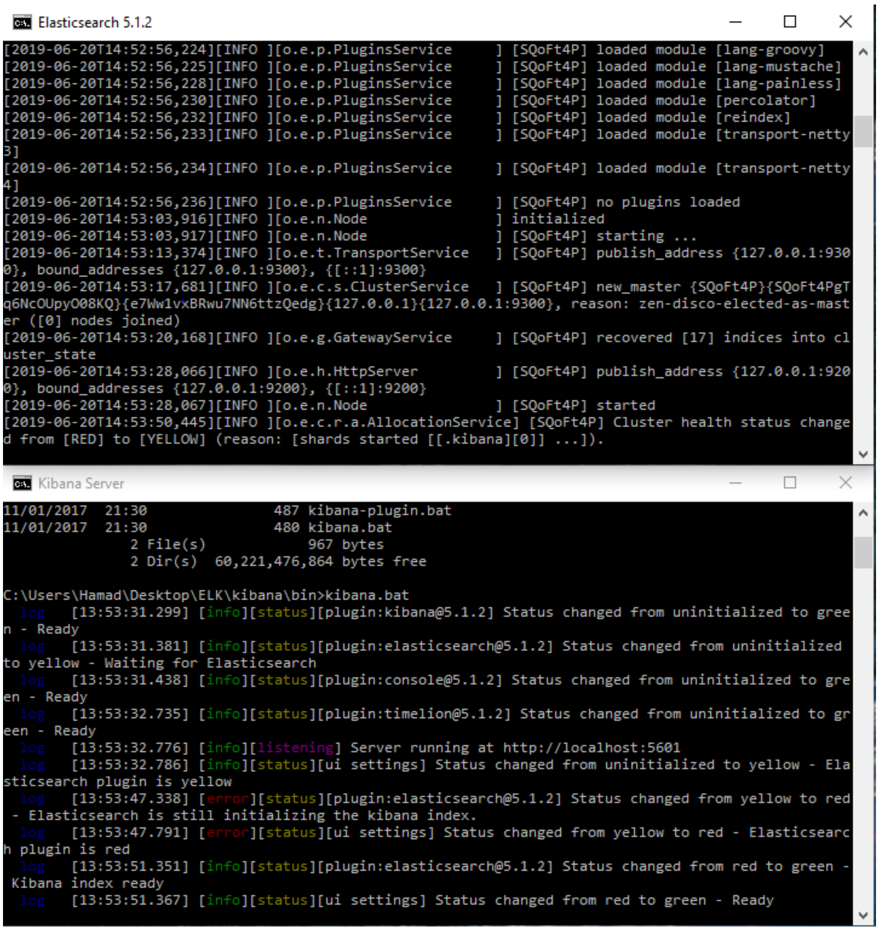
Furthermore, we have presented the experiment modules using Windows operating system in Figure 5.4 for shows the running modules of filebeat and logstash and in Figure 5.5 for elasticsearch and kibana. We briefly explain the Elastic stack and associated technologies that are used in this paper as follows -


C:\Windows\System32\cmd.exe - filebeat.exe -c filebeat.yml -e

```
plate file: C:\Users\Hamad\Desktop\ELK\filebeat\filebeat.template-es2x.json
2019/06/20 19:48:55.206126 client.go:120: INFO Elasticsearch url: http://localhost:9200
2019/06/20 19:48:55.210125 outputs.go:106: INFO Activated elasticsearch as output plugin.
2019/06/20 19:48:55.215607 logstash.go:90: INFO Max Retries set to: 3
2019/06/20 19:48:55.219601 outputs.go:106: INFO Activated logstash as output plugin.
2019/06/20 19:48:55.226587 publish.go:291: INFO Publisher name: HAMAD-PC
2019/06/20 19:48:55.262026 async.go:63: INFO Flush Interval set to: 1s
2019/06/20 19:48:55.262528 async.go:64: INFO Max Bulk Size set to: 50
2019/06/20 19:48:55.265522 async.go:63: INFO Flush Interval set to: 1s
2019/06/20 19:48:55.268014 async.go:64: INFO Max Bulk Size set to: 2048
2019/06/20 19:48:55.271006 beat.go:207: INFO filebeat start running.
2019/06/20 19:48:55.273001 registrar.go:85: INFO Registry file set to: C:\Users\Hamad\Desktop\ELK\filebeat
\data\registry
2019/06/20 19:48:55.275995 registrar.go:106: INFO Loading registrar data from C:\Users\Hamad\Desktop\ELK\f
ilebeat\data\registry
2019/06/20 19:48:55.349887 registrar.go:123: INFO States Loaded from registrar: 1055
2019/06/20 19:48:55.350861 crawler.go:34: INFO Loading Prospectors: 1
2019/06/20 19:48:55.350861 registrar.go:236: INFO Starting Registrar
2019/06/20 19:48:55.350861 spooler.go:63: INFO Starting spooler: spool_size: 2048; idle_timeout: 5s
2019/06/20 19:48:55.350861 sync.go:41: INFO Start sending events to output
2019/06/20 19:48:55.378305 prospector_log.go:57: INFO Prospector with previous states loaded: 688
2019/06/20 19:48:55.379305 crawler.go:46: INFO Loading Prospectors completed. Number of prospectors: 1
2019/06/20 19:48:55.380801 crawler.go:61: INFO All prospectors are initialised and running with 1055 state
s to persist
2019/06/20 19:48:55.381301 prospector.go:111: INFO Starting prospector of type: log
```

```
if an Elasticsearch connection is working {:url=>#<URI::HTTP:0x3cb0aa09 URL:http://localhost:9200>, :healt
hcheck_path=>"/"}
14:55:13.726 [[main]-pipeline-manager] WARN  logstash.outputs.elasticsearch - Restored connection to ES in
stance {:url=>#<URI::HTTP:0x3cb0aa09 URL:http://localhost:9200>}
14:55:13.795 [[main]-pipeline-manager] INFO  logstash.outputs.elasticsearch - Using mapping template from
{:path=>nil}
14:55:14.560 [[main]-pipeline-manager] INFO  logstash.outputs.elasticsearch - Attempting to install templa
te {:manage_template=>{"template"=>"logstash-*", "version"=>50001, "settings"=>{"index.refresh_interval"=>
"5s"}, "mappings"=>{"_default_"=>{"_all"=>{"enabled"=>true, "norms"=>false}, "dynamic_templates"=>[{"messa
ge_field"=>{"path_match"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>
false}}}, {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "no
rms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword"}}}}}], "properties"=>{"@timestamp"=>{"type"=>"date",
 "include_in_all"=>false}, "@version"=>{"type"=>"keyword", "include_in_all"=>false}, "geoip"=>{"dynamic"=>
true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_f
loat"}, "longitude"=>{"type"=>"half_float"}}}}}}}}
14:55:14.608 [[main]-pipeline-manager] INFO  logstash.outputs.elasticsearch - New Elasticsearch output {:c
lass=>"LogStash::Outputs::ElasticSearch", :hosts=>["localhost:9200"]}
14:55:15.132 [[main]-pipeline-manager] INFO  logstash.filters.geoip - Using geoip database {:path=>"C:/Use
rs/Hamad/Desktop/ELK/logstash/vendor/bundle/jruby/1.9/gems/logstash-filter-geoip-4.0.4-java/vendor/GeoLite
2-City.mmdb"}
14:55:15.251 [[main]-pipeline-manager] INFO  logstash.pipeline - Starting pipeline {"id"=>"main", "pipelin
e.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max_inflight"=>500}
14:55:15.305 [[main]-pipeline-manager] INFO  logstash.pipeline - Pipeline main started
14:55:16.238 [Api Webserver] INFO  logstash.agent - Successfully started Logstash API endpoint {:port=>960
0}
```

Fig. 5.4 Executing Elastic Stack for Data Analysis - Filebeat and Logstash



```
[2019-06-20T14:52:56,224][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [lang-groovy]
[2019-06-20T14:52:56,225][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [lang-mustache]
[2019-06-20T14:52:56,228][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [lang-painless]
[2019-06-20T14:52:56,230][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [percolator]
[2019-06-20T14:52:56,232][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [reindex]
[2019-06-20T14:52:56,233][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [transport-netty
3]
[2019-06-20T14:52:56,234][INFO ][o.e.p.PluginsService     ] [SQoFt4P] loaded module [transport-netty
4]
[2019-06-20T14:52:56,236][INFO ][o.e.p.PluginsService     ] [SQoFt4P] no plugins loaded
[2019-06-20T14:53:03,916][INFO ][o.e.n.Node               ] initialized
[2019-06-20T14:53:03,917][INFO ][o.e.n.Node               ] [SQoFt4P] starting ...
[2019-06-20T14:53:13,374][INFO ][o.e.t.TransportService   ] [SQoFt4P] publish_address {127.0.0.1:930
0}, bound_addresses {127.0.0.1:9300}, {[::1]:9300}
[2019-06-20T14:53:17,681][INFO ][o.e.c.s.ClusterService   ] [SQoFt4P] new_master {SQoFt4P}{SQoFt4PgT
q6NcOUpyO08KQ}{e7Ww1vxBRwu7NN6ttzQedg}{127.0.0.1}{127.0.0.1:9300}, reason: zen-disco-elected-as-mast
er ([0] nodes joined)
[2019-06-20T14:53:20,168][INFO ][o.e.g.GatewayService     ] [SQoFt4P] recovered [17] indices into cl
uster_state
[2019-06-20T14:53:28,066][INFO ][o.e.h.HttpServer         ] [SQoFt4P] publish_address {127.0.0.1:920
0}, bound_addresses {127.0.0.1:9200}, {[::1]:9200}
[2019-06-20T14:53:28,067][INFO ][o.e.n.Node               ] [SQoFt4P] started
[2019-06-20T14:53:50,445][INFO ][o.e.c.r.a.AllocationService] [SQoFt4P] Cluster health status change
d from [RED] to [YELLOW] (reason: [shards started [[.kibana][0]] ...]).
```

```
11/01/2017  21:30               487 kibana-plugin.bat
11/01/2017  21:30               480 kibana.bat
               2 File(s)            967 bytes
               2 Dir(s)  60,221,476,864 bytes free

C:\Users\Hamad\Desktop\ELK\kibana\bin>kibana.bat
  log   [13:53:31.299] [info][status][plugin:kibana@5.1.2] Status changed from uninitialized to gree
n - Ready
  log   [13:53:31.381] [info][status][plugin:elasticsearch@5.1.2] Status changed from uninitialized
to yellow - Waiting for Elasticsearch
  log   [13:53:31.438] [info][status][plugin:console@5.1.2] Status changed from uninitialized to gre
en - Ready
  log   [13:53:32.735] [info][status][plugin:timelion@5.1.2] Status changed from uninitialized to gr
een - Ready
  log   [13:53:32.776] [info][listening] Server running at http://localhost:5601
  log   [13:53:32.786] [info][status][ui settings] Status changed from uninitialized to yellow - Ela
sticsearch plugin is yellow
  log   [13:53:47.338] [error][status][plugin:elasticsearch@5.1.2] Status changed from yellow to red
 - Elasticsearch is still initializing the kibana index.
  log   [13:53:47.791] [error][status][ui settings] Status changed from yellow to red - Elasticsearc
h plugin is red
  log   [13:53:51.351] [info][status][plugin:elasticsearch@5.1.2] Status changed from red to green -
 Kibana index ready
  log   [13:53:51.367] [info][status][ui settings] Status changed from red to green - Ready
```

Fig. 5.5 Executing Elastic Stack for Data Analysis - Elasticsearch and Kibana

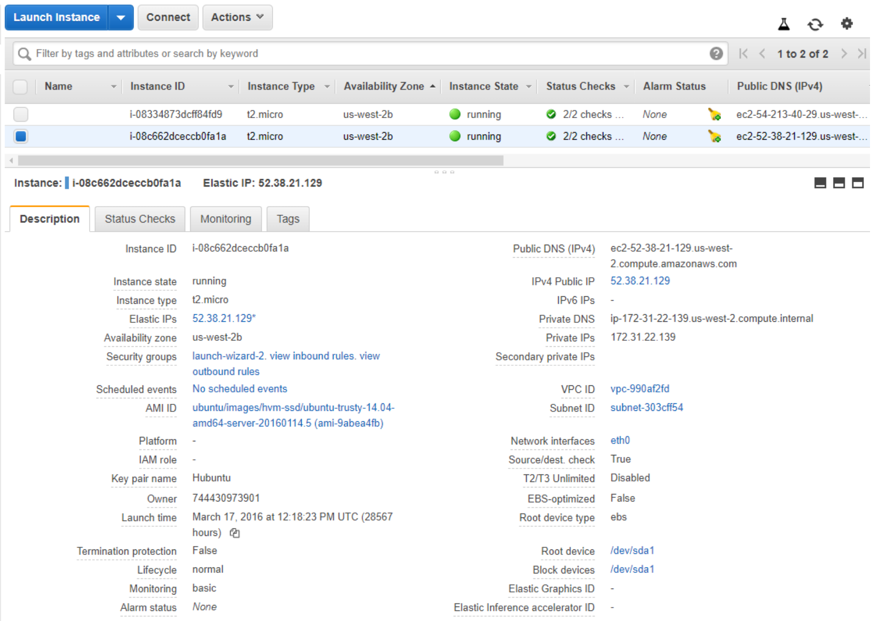
- Elasticsearch: Elasticsearch is one of the prominent search techniques that can search text from almost any format or platform. Elasticsearch is highly scalable and flexible [75]. This is also distributed and uses RESTful search technique, which ensures the exact search result. Elasticsearch is the 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 [76] on Amazon as cloud services as shown in Figure 5.6. There are two instances are running at the same time for collecting data. The operating system is Ubuntu, which is a variation of UNIX system. We have launched the instances on March 2016 and running for more than 28567 hours.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-08334873dcff84fd9	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-213-40-29.us-west-...
	i-08c662dceccb0fa1a	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-52-38-21-129.us-west-...

Instance: i-08c662dceccb0fa1a Elastic IP: 52.38.21.129

Instance ID	i-08c662dceccb0fa1a	Public DNS (IPv4)	ec2-52-38-21-129.us-west-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	52.38.21.129
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs	52.38.21.129*	Private DNS	ip-172-31-22-139.us-west-2.compute.internal
Availability zone	us-west-2b	Private IPs	172.31.22.139
Security groups	launch-wizard-2. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-990af2fd
AMI ID	ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-20160114.5 (ami-9abea4fb)	Subnet ID	subnet-303cff54
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	Hubuntu	T2/T3 Unlimited	Disabled
Owner	744430973901	EBS-optimized	False
Launch time	March 17, 2016 at 12:18:23 PM UTC (28567 hours)	Root device type	ebs
Termination protection	False	Root device	/dev/sda1
Lifecycle	normal	Block devices	/dev/sda1
Monitoring	basic	Elastic Graphics ID	-
Alarm status	None	Elastic Inference accelerator ID	-

Fig. 5.6 Honeypot Instances in Amazon Cloud

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 (Sample data can be found in Appendix B) about three years.

Both of the honeypots 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 log data is enormous, which is very difficult to analyse simply by looking at the log files. So, we adapted 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 [75].

5.3.1 Data Collection Technology

We have used various data collection technologies in Amazon cloud services. Amazon cloud services provide an intelligent and scalable framework for hosting honeypots. This subsection discusses the building blocks of the services that used to collect honeypot data for this experiment.

Amazon EC2, which is a cloud-based service, provides scalable computing space in the cloud. It provides a virtual server to deploy operating systems instances as virtual machines, which is known as Amazon Machine Images (AMI). AMI technology can provide preconfigured virtual machines such as Ubuntu desktop and server, and Windows desktop and server [77][78]. Amazon offers virtual machines and storages. The storage size could be agreed with the clients. This system also gives elasticity as it can expand and reduce the size and network bandwidth the service as required for a particular service[79]. EC2 also provides easy and fast setup to launch an application or operating system, which highly faults tolerant and resilient to failure [78][80].

5.4 Result and Discussion

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. Figure 5.7 illustrates the attack events in the Kippo honeypot. We identified ten keywords, which can be recognised as event 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.

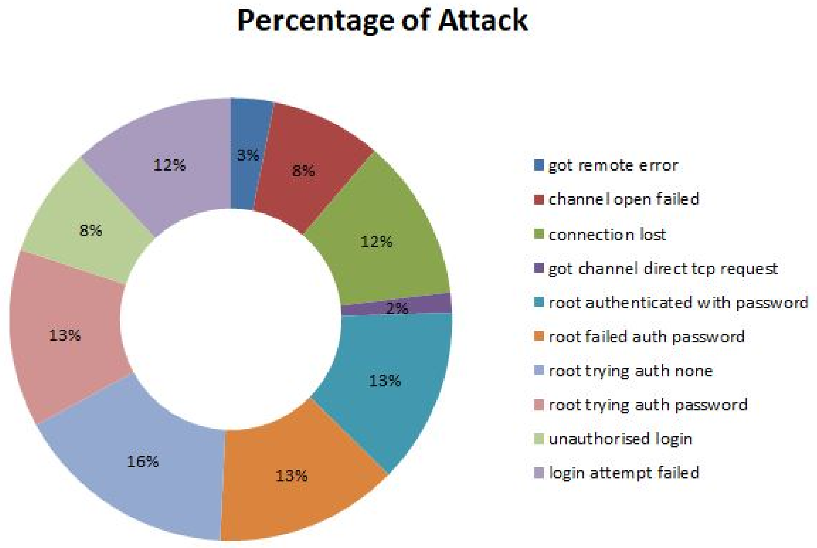


Fig. 5.7 Kippo Honeypot Log Event Visualisation using Kibana

Table 5.1 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	5,283,03	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%

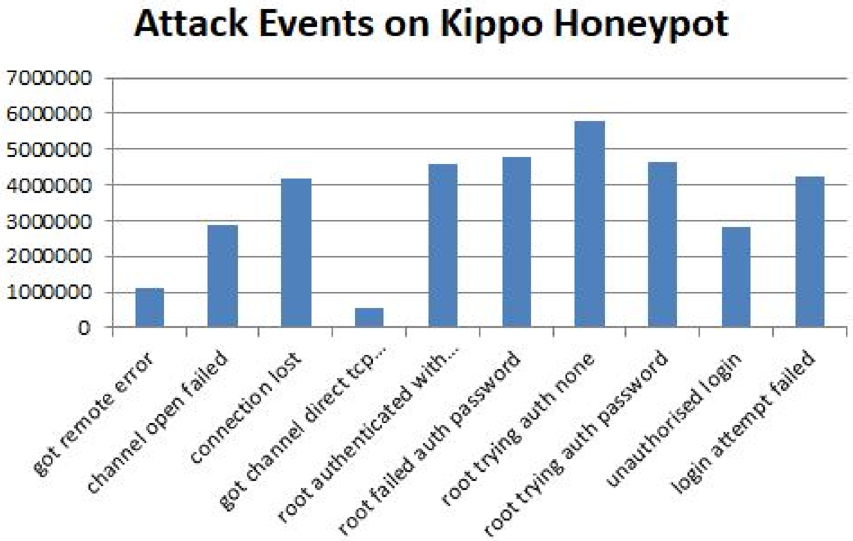


Fig. 5.8 Attack Events in Bar Chart View

Each of the keywords in Table 5.1 are identified from the honeypot data indicates cyber-attack events. Each of the events are presented by using Kibana visualisation in Figure 5.7 and number of attack events presented in Figure 5.8. 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 incident 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 been hit. This is one of the highest events, which is about 4,574,932 and 12.86%.
- channel open failed: This event happened about 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 have been failed. The percentage of this event is 11.94%, which is very high compare to other events happened to the honeypot.

The result has been summarised in table 5.1 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 identified 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 second event is the 'root failed auth password', which occurred 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.

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 5.2 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% compare to other 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/minute, i.e., about 8.39 attacks/seconds. The attack frequency

Table 5.2 Brute Force Attack

Password	**No of time occurred**	**% of occurring**
123456	60,099	37.35%
abc123	5,086	3.16%
admin	57,839	35.94%
asdf	3,848	2.39%
asdf.*	20,310	12,62%
password	8,149	5.06%
Password1	939	0.58%
qwer	4,619	2.87%

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 example of the username and password are [root/p123456789], [root/p0o9i8], [root/parr0lla789], [root/pass!@#], [root/onlyidc2010] and [root/nhs39f40201]. However, they have used PUTTY as a tool for SSH connection establishment. Figure 5.9 shows the frequency of the brute force attack by matching username and password.

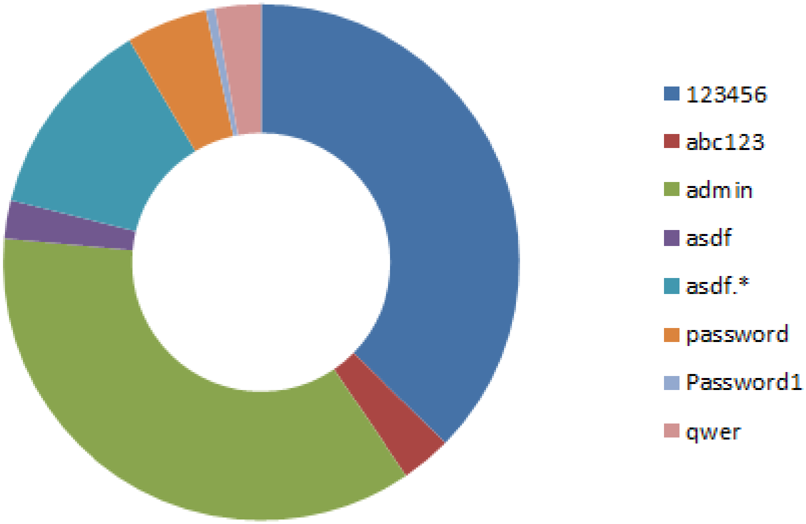
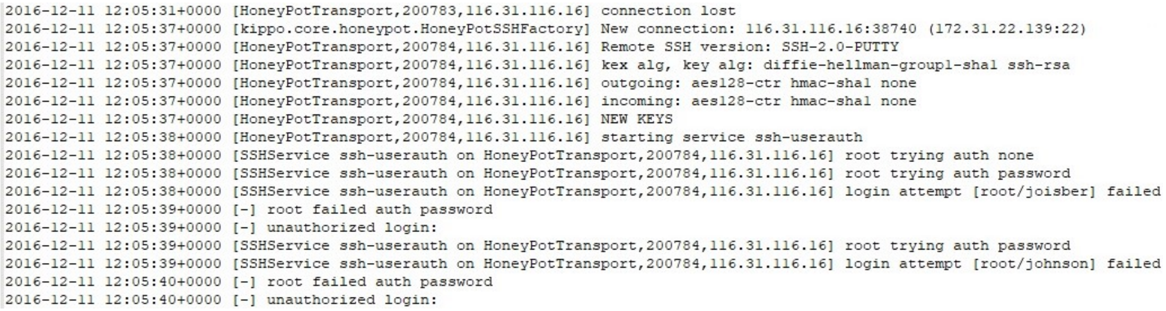


Fig. 5.9 Brute Force Attack on Honeypot

Several attack originated by the attackers start with a 'New connection'. They always use 'NEW KEYS' although they 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' 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 incident as the 'root failed auth password' since the username was 'root'. Also, this is considered as 'unauthorized login' for the attacker's IP address. Figure 5.10 is the snippet from the honeypot log, which shows the time-stamp for each of the steps taken by the attacker for a whole session.

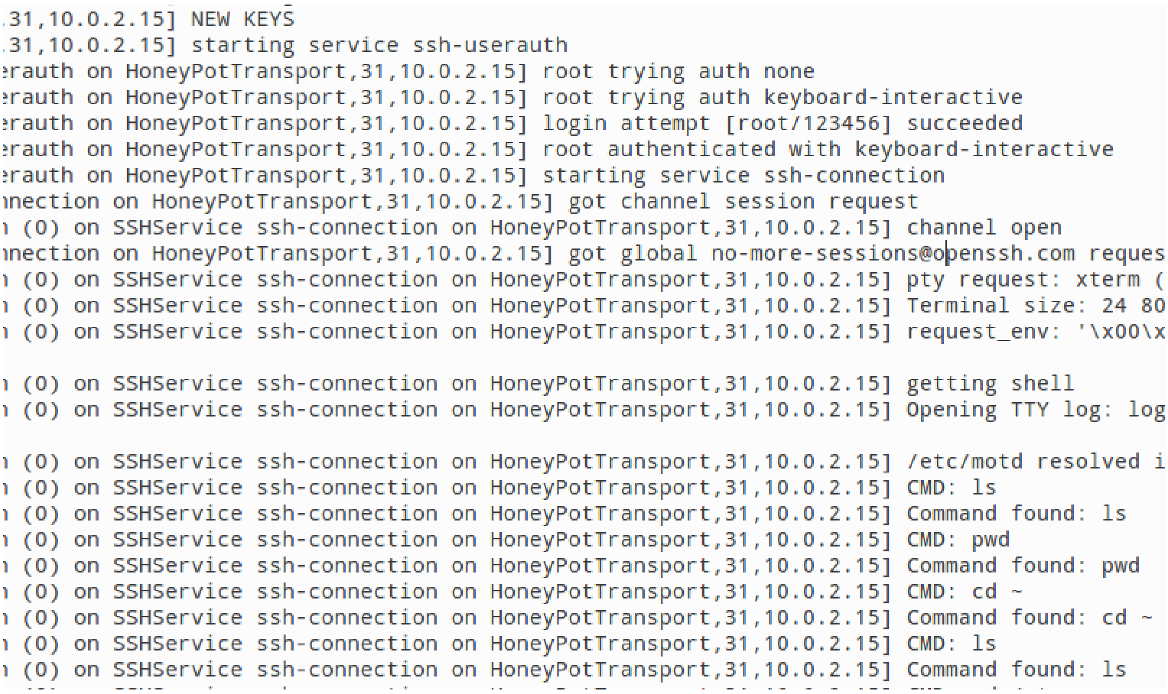


```
2016-12-11 12:05:31+0000 [HoneyPotTransport,200783,116.31.116.16] connection lost
2016-12-11 12:05:37+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:38740 (172.31.22.139:22)
2016-12-11 12:05:37+0000 [HoneyPotTransport,200784,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-11 12:05:37+0000 [HoneyPotTransport,200784,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-11 12:05:37+0000 [HoneyPotTransport,200784,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-11 12:05:37+0000 [HoneyPotTransport,200784,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-11 12:05:37+0000 [HoneyPotTransport,200784,116.31.116.16] NEW KEYS
2016-12-11 12:05:38+0000 [HoneyPotTransport,200784,116.31.116.16] starting service ssh-userauth
2016-12-11 12:05:38+0000 [SSHService ssh-userauth on HoneyPotTransport,200784,116.31.116.16] root trying auth none
2016-12-11 12:05:38+0000 [SSHService ssh-userauth on HoneyPotTransport,200784,116.31.116.16] root trying auth password
2016-12-11 12:05:38+0000 [SSHService ssh-userauth on HoneyPotTransport,200784,116.31.116.16] login attempt [root/joisber] failed
2016-12-11 12:05:39+0000 [-] root failed auth password
2016-12-11 12:05:39+0000 [-] unauthorized login:
2016-12-11 12:05:39+0000 [SSHService ssh-userauth on HoneyPotTransport,200784,116.31.116.16] root trying auth password
2016-12-11 12:05:39+0000 [SSHService ssh-userauth on HoneyPotTransport,200784,116.31.116.16] login attempt [root/johnson] failed
2016-12-11 12:05:40+0000 [-] root failed auth password
2016-12-11 12:05:40+0000 [-] unauthorized login:
```

Fig. 5.10 An Unsuccessful Attack Event

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 5.11 is the part of the successful attack, which shows that after the attacker logged into the honeypot, he/she is using UNIX commands.

Each of the attackers come in different ways using the different IP address and different. Some attackers keep changing their IP addresses and domain name. Some IP addresses were used only once that could not be found afterwards. Some behaviour of the attackers is



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

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

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

Fig. 5.11 A Successful Attack Event

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 identify how they operate an attack.

Since the attackers attack the honeypot system assuming that they are attacking to a real system. Each of the attackers, who attack to 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 password. It has also been noticed that the attacker does not change the IP address every time he or she attacks.

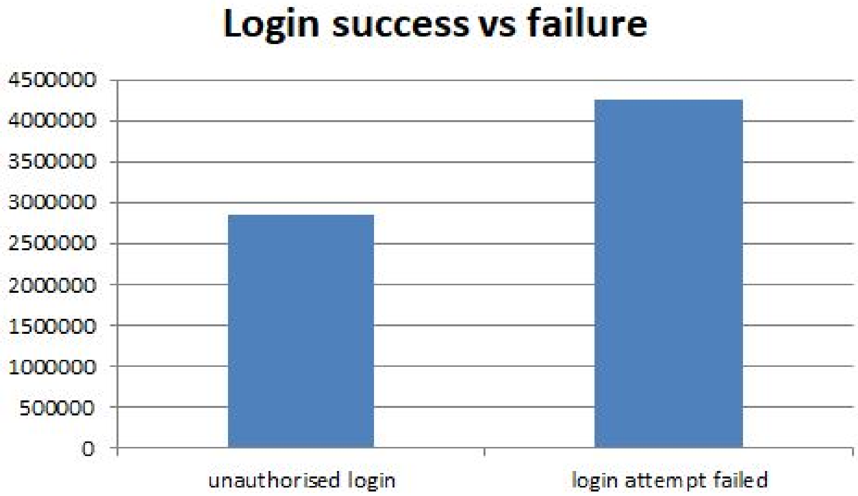


Fig. 5.12 Login success vs. failure

Figure 5.12, 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 as 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.

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 5.13 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) is 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.

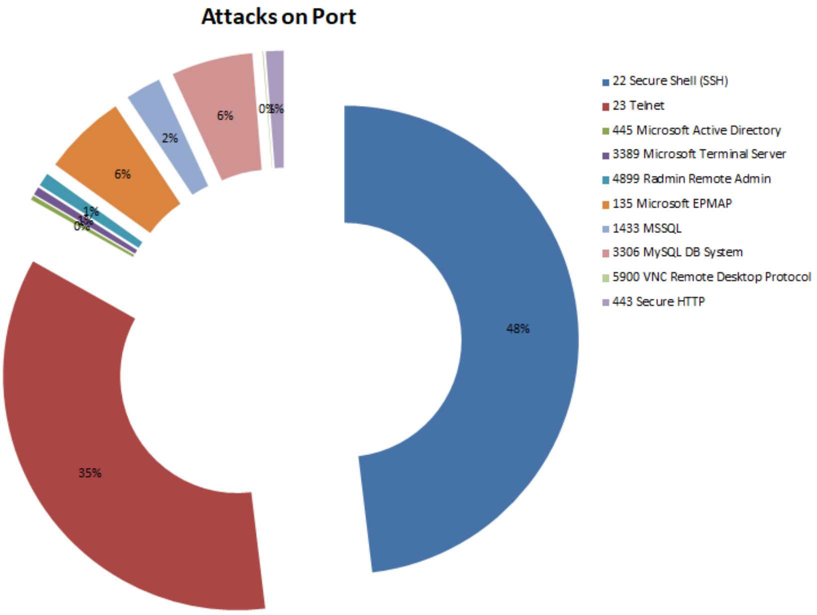


Fig. 5.13 Attack Using Port

We have identified the geographic location of the attackers from the IP addresses they used. Figure 5.14 illustrates that attackers form the USA hit about 51% followed by China, which is approximately 40%. There are other countries including Pakistan, Iran and Romania are in the list of originating attacks.

Moreover it has been noticed that once the attacker managed to get into the system they look various information. Figure 5.15 illustrates a number of commands executed by the attacker in the honeypot. 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’, ’ssh’, etc., are used several times. The user of commands provides insights

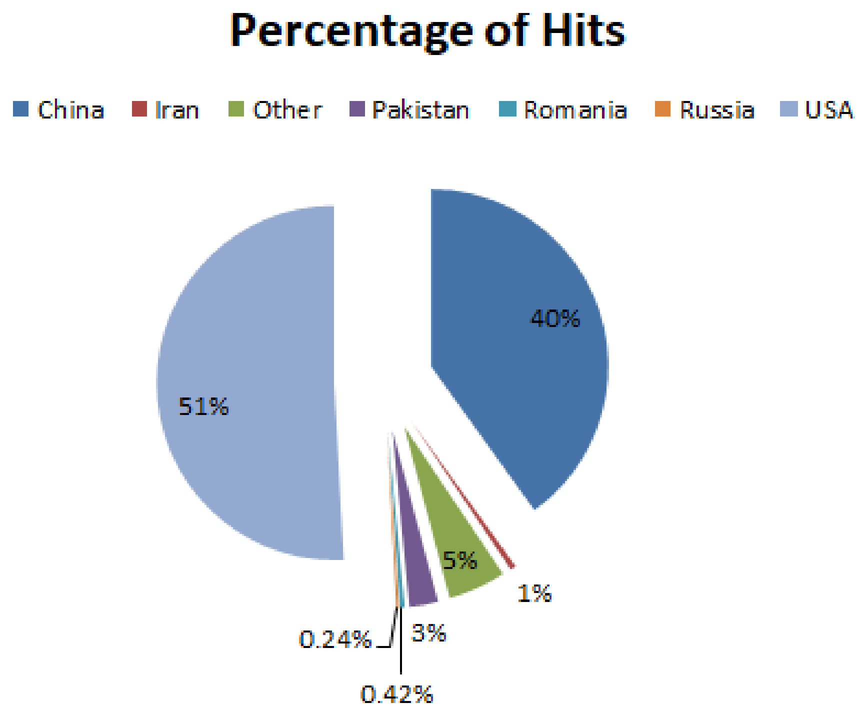


Fig. 5.14 Number of Attack from Geographic Locations

about the attacker’s motive. However, some attackers user random commands, which means they don’t have planning of the attack.

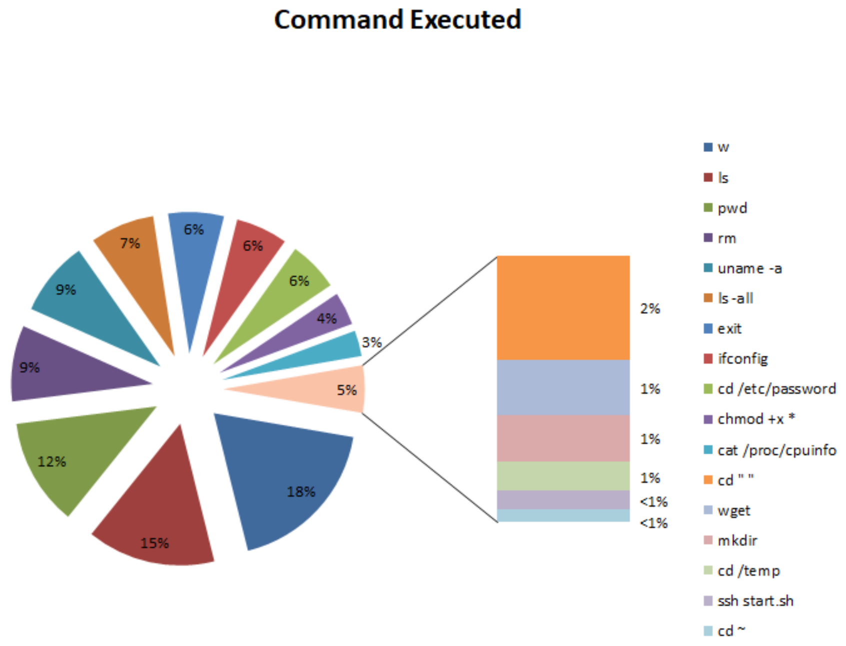


Fig. 5.15 Command Executed by the Attacker after Logging to the system

This experiment gives essential insights about 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. There were many SSH attack 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 attack is related to many artefacts. To find those artefacts, elasticsearch played a significant role, which helps to identify various types of cyber incident 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.

5.5 Summary

In this chapter 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 Elastic stack for log data visualisation. It is worth noting that Elastic Stack 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 system.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research has identified several aspects of cyber-attack modelling for threat intelligence. The main aim was to understand the cyber-attack using honeypot and elastic technology. However, it has been identified that using state-of-the-art technologies are not the only solution to mitigate the cyber-attack. In order to protect a system, it requires you to consider many aspects such as the knowledge of employees about the cyber-attack, attacker activity within the network and plan for attack handling.

This study shows that IT employee knowledge of cyber threat within an organisation is significantly poor. In this context, we have surveyed among a number of IT employees such as SOC, Non-SOC, NOC, Non-NOC, System administrator, Database administrator, Network engineer, Application developer and System architect to understand the cyber-attack or threat related knowledge among them. The results show that different IT teams have a different level of knowledge of the topic, i.e., some team has better knowledge on some issues than others. For example, the SOC team have more knowledge than the Non-SOC team about cybersecurity threat. Through the survey result we have suggested a number of security assessment services such as network-based attack & prevention, host-based, application and compliance. We also suggested physical security assessment, such as securing server and workstations.

The second major finding is the introduction of a new cyber-attack modelling techniques called the Object-based model, which considers all the components, including employees within two categories called direct and indirect objects. To analyse the performance of this model, we use a case study. The case study was analysed with an Object-based model and three prominent attack modelling techniques called the Diamond Model, kill Chain and an Attack Graph. In this case study, we use a Windows-based executable file called '.exe'. The file was sent to the user as an attachment to analyse how four models behave to tackle this cyber-attack situation. Each of the cyber-attack models provides a unique view of the attack scenario. For example, Diamond and Object-based model emphasise the employee or user as a victim, which supports our previous finding as an employee could be a threat. However, the Attack Graph and Kill Chain do not consider the human as a threat, but they share some standard features like the handling after an attack. Moreover, we have suggested some features that should be considered for future attack modelling.

The final contribution in this thesis is the proposed threat intelligence system, which mainly focuses on three components: a) attack; b) pattern and c) behaviour. Threat intelligence is a process 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. Moreover, the threat intelligence system provides a significant amount of insightful data in near real-time. The experiment shows that collecting and analysing honeypot data has huge potentials for future threat prevention since a honeypot is not a real system but could be mimicked as a real one. 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. One of the most significant finding to emerge from this study is the continuous monitoring to find an attack pattern and attack on the real-system before an attack happens.

6.2 Limitations and Challenges

Although this thesis satisfies all the aims and objectives, it comes with some limitations.

Firstly, a limitation of this study is the sample size, when surveying IT employee knowledge. However, due to the security and privacy issue, it was not possible to reach more participants.

Secondly, the study did not evaluate other types of malware apart from PE. Analysing another type of malware could give different types of result. Also, we consider only four attack modelling systems to analyse an attack scenario, which gives interesting result, but if the scenario changes the behaviour of these model may change as well. However, this approach could be generalised in the future.

Thirdly, the proposed Object-based attack modelling technique could not be compared with other modelling techniques and scenarios. It would be interesting to see how this model reacts in a real attack scenario.

Finally, we set up the honeypot in the Amazon cloud service, which mimics a real Ubuntu operating system. Due to the limitation of time and resources, we could not deploy different types of honeypots such as web and database server, to extract and visualise data.

6.3 Future Work

The main objectives of this research were to provide better protection to organisation’s that are currently facing challenges with several types of cyber-attacks. This research satisfies all the requirements initially defined; however, there is a particular aspect that can be enhanced in the research and within the implementation context. This section explains the possible future direction of this cyber-attack modelling for threat intelligence.

We aim to perform further survey research on other organisations to get reliable results on the average awareness of IT employees on cybersecurity. It is recommended to use another method to assess employees' knowledge on cybersecurity through penetration testing, observations and simulation. To reduce the gap of knowledge between Security operation team and other IT expert. It is suggested that in order for future work to enhance communication between both teams there needs to be regular meetings and to provide a scheduled transfer knowledge session on specific knowledge including cybersecurity. Also, providing specialised training on cybersecurity or awareness sessions would be beneficial to ensure that all IT employees absorb the required knowledge on cybersecurity and on how to handle such incidents effectively.

The Object-based model can be extended to handle APT, which is one of the issues in the cyber security world [72]. In APT, a group of organised attackers utilise full planning using advanced infrastructure and capability to attack a corporate network. The Object-based model can be used to analyse each of the objects within the network to identify the vulnerability.

It is important to have an intelligent driven mechanism for network defence. 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 gives a significant number of indicators to prevent a future attack.

The model works only when there are a considerable number of network incident-related data available for analysis. The data is analysed using Elastic Stack for log data visualisation, which is a highly scalable and flexible technology for analysing and visualising data. Honeypot data analysis for threat intelligence also provides a cost-effective way to gather intelligence rather than using the production system. In the future, we aim to extend this model using machine learning to automated attack pattern and behaviour, which could be fed to security systems like IDS and IPS. Actual network data can be analysed using Artificial Intelligence based Machine Learning Techniques and associated algorithms in order to predict future security incidents and enable the design and development of credible preventive security mechanisms, as appropriate.

Security processing (c.f., encryption and decryption times etc.) as well as delays due to intrusion detection of security bridges and subsequent restoration operations and closures to recover and bring the network back into the secure state have adverse impact on performance. In this context, combined performance security measures of ‘optimal’ performance and security trade-offs of advanced queueing network models (QNMs) and Generalised Stochastic Petri Nets (GSPNs) will be devised [81]. Moreover, extended security detection control models will be developed, based on the ones proposed in [81] in order to determine ‘optimal’ encryption/decryption times and associated optimal encryption/decryption key lengths.

References

[1] Mario Golling and Björn Stelte. Requirements for a future ews-cyber defence in the internet of the future. In *Cyber conflict (ICCC), 2011 3rd international conference on*, pages 1–16. IEEE, 2011.

[2] Jason Andress and Steve Winterfeld. *Cyber warfare: techniques, tactics and tools for security practitioners*. Elsevier, 2013.

[3] Radoniaina Andriatsimandefitra and Valérie Viet Triem Tong. Capturing android malware behaviour using system flow graph. In *International Conference on Network and System Security*, pages 534–541. Springer, 2015.

[4] Daavid Hentunen. Behaviour based malware prevention, June 8 2017. US Patent App. 15/362,012.

[5] Michael Sikorski and Andrew Honig. *Practical malware analysis: the hands-on guide to dissecting malicious software*. no starch press, 2012.

[6] Qassim Nasir and Zahraa A Al-Mousa. Honeypots aiding network forensics: Challenges and notions. *JCM*, 8(11):700–707, 2013.

[7] Adel Ammar. Comparison of feature reduction techniques for the binominal classification of network traffic. *Journal of Data Analysis and Information Processing*, 3(02):11, 2015.

[8] Sumeet Kumar and Kathleen M Carley. Approaches to understanding the motivations behind cyber attacks. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 307–309. IEEE, 2016.

[9] Ben Brewster, Benn Kemp, Sara Galehbakhtiari, and Babak Akhgar. Cybercrime: attack motivations and implications for big data and national security. In *Application of Big Data for National Security*, pages 108–127. Elsevier, 2015.

[10] Bank. Cbest intelligence led testing an introduction to cyber threat modelling. 2016.

[11] SQRRL. A framework for cyber threat hunting, 2016.

[12] Jungsuk Song, Hiroki Takakura, Yasuo Okabe, Masashi Eto, Daisuke Inoue, and Koji 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*, pages 29–36. ACM, 2011.

[13] Georgios Portokalidis, Asia Slowinska, and Herbert Bos. Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. In *ACM SIGOPS Operating Systems Review*, volume 40, pages 15–27. ACM, 2006.

[14] Jop van der Lelie-jop and Rory Breuk-rory. A visual analytic approach for analyzing ssh honeypots. 2012.

[15] Pavol Sokol, Patrik Pekarčík, and Tomáš Bajtoš. Data collection and data analysis in honeypots and honeynets. *Proceedings of the Security and Protection of Information. University of Defence*, 2015.

[16] Chris Moore and Ameer Al-Nemrat. An analysis of honeypot programs and the attack data collected. In *International Conference on Global Security, Safety, and Sustainability*, pages 228–238. Springer, 2015.

[17] David Binaco. A framework for cyber threat hunting part 1: The pyramid of pain, 2015.

[18] Xiaoli Lin, Pavol Zavarsky, Ron Ruhl, and Dale Lindskog. Threat modeling for csrf attacks. *2013 IEEE 16th International Conference on Computational Science and Engineering*, 3:486–491, 2009.

[19] BSIMM. Attack models with bsimm frameworks. *Online*, https://www.bsimm.com/framework/intelligence/attack-models/, 2016.

[20] 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)*, pages 69–76, Aug 2016.

[21] Tolulope Awojana. Threat modelling and analysis of web application attacks. 2018.

[22] Cynthia Phillips and Laura Painton Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 Workshop on New Security Paradigms*, NSPW '98, pages 71–79, New York, NY, USA, 1998. ACM.

[23] Bruce Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

[24] Martin Mulazzani, Sebastian Schrittwieser, Manuel Leithner, Markus Huber, and Edgar R Weippl. Dark clouds on the horizon: Using cloud storage as attack vector and online slack space. In *USENIX Security Symposium*, pages 65–76. San Francisco, CA, USA, 2011.

[25] Pratyusa K Manadhata and Jeannette M Wing. An attack surface metric. *Software Engineering, IEEE Transactions on*, 37(3):371–386, 2011.

[26] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. The diamond model of intrusion analysis. Technical report, DTIC Document, 2013.

[27] United States. Joint Chiefs of Staff. *Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Battlespace*. Joint Chiefs of Staff, 2000.

[28] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1:80, 2011.

[29] Walter L Sharp. Joint publication 3-60: Joint targeting. *Washington DC: Joint Chiefs of Staff*, 2007.

[30] Taketoshi Sakuraba, Seiichi Domyo, Bin-Hui Chou, and Kouichi Saku. Exploring security countermeasures along the attack sequence. In *Information Security and Assurance, 2008. ISA 2008. International Conference on*, pages 427–432. IEEE, 2008.

[31] Dennis Kiwia, Ali Dehghantanha, Kim-Kwang Raymond Choo, and Jim Slaughter. A cyber kill chain based taxonomy of banking trojans for evolutionary computational intelligence. *Journal of computational science*, 27:394–409, 2018.

[32] Blake D Bryant and Hossein Saiedian. A novel kill-chain framework for remote security log analysis with siem software. *computers & security*, 67:198–210, 2017.

[33] Nirnay Ghosh and Soumya K Ghosh. A planner-based approach to generate and analyze minimal attack graph. *Applied Intelligence*, 36(2):369–390, 2012.

[34] Karthik Selvaraj and Nino Fred Gutierrez. The rise of pdf malware. *Symantec Security Response*, 2010.

[35] Vivek Shandilya, Chris B Simmons, and Sajjan Shiva. Use of attack graphs in security systems. *Journal of Computer Networks and Communications*, 2014, 2014.

[36] NOEL STEVEN and SUSHIL JAJODIA. Measuring security risk of networks using attack graphs. *International Journal of Next—Generation Computing*, 1(1), 2010.

[37] R Lippmann. Netspa: A network security planning architecture. *Massachusetts Institute of Technology*, 2002.

[38] Xinming Ou, Sudhakar Govindavajhala, and Andrew W Appel. Mulval: A logic-based network security analyzer. In *USENIX security*, 2005.

[39] Xinming Ou and Anoop Singhal. Attack graph techniques. In *Quantitative Security Risk Assessment of Enterprise Networks*, pages 5–8. Springer, 2011.

[40] Leo Obrst, Penny Chase, and Richard Markeloff. Developing an ontology of the cyber security domain. In *STIDS*, pages 49–56, 2012.

[41] Christian Seifert, Ian Welch, Peter Komisarczuk, et al. Honeyc-the low-interaction client honeypot. *Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand*, 2007.

[42] ROMAN Jasek, MARTIN Kolarik, and TOMAS Vymola. Apt detection system using honeypots. In *Proceedings of the 13th International Conference on Applied Informatics and Communications (AIC'13), WSEAS Press*, pages 25–29, 2013.

[43] Nathalie 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*, pages 109–114. IEEE, 2002.

[44] Steffen Liebergeld, Matthias Lange, and Collin Mulliner. Nomadic honeypots: A novel concept for smartphone honeypots. In *Proc. W'shop on Mobile Security Technologies (MoST'13), together with 34th IEEE Symp. on Security and Privacy*, 2013.

[45] Gary Kelly and Diane Gan. Analysis of attacks using a honeypot. In *International Cybercrime, Security and Digital Forensics Conference*, 2011.

[46] Vasileios Mavroeidis and Audun Jøsang. Data-driven threat hunting using sysmon. In *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*, pages 82–88. ACM, 2018.

[47] Onur Catakoglu, Marco Balduzzi, and Davide Balzarotti. Automatic extraction of indicators of compromise for web applications. In *Proceedings of the 25th International Conference on World Wide Web*, pages 333–343. International World Wide Web Conferences Steering Committee, 2016.

[48] Di Xiao, Xiaofeng Liao, and Shaojiang Deng. One-way hash function construction based on the chaotic map with changeable-parameter. *Chaos, Solitons & Fractals*, 24(1):65–71, 2005.

[49] Simon J Bell. Building a honeypot to research cyber-attack techniques interim report. *Univ. Sussex, Brighton, UK, Tech. Rep*, 2013.

[50] Marcin Nawrocki, Matthias Wählisch, Thomas C Schmidt, Christian Keil, and Jochen Schönfelder. A survey on honeypot software and data analysis. *arXiv preprint arXiv:1608.06249*, 2016.

[51] Emmanouil Vasilomanolakis, Shankar Karuppayah, Panayotis Kikiras, and Max Mühlhäuser. A honeypot-driven cyber incident monitor: lessons learned and steps ahead. In *Proceedings of the 8th International Conference on Security of Information and Networks*, pages 158–164. ACM, 2015.

[52] Stephen Brown, Rebecca Lam, Shishir Prasad, Sivasubramanian Ramasubramanian, and Josh Slauson. Honeypots in the cloud. *University of Wisconsin-Madison*, 2012.

[53] Robert McGrew. Experiences with honeypot systems: Development, deployment, and analysis. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 9, pages 220a–220a. IEEE, 2006.

[54] Lance Spitzner. *Honeypots: tracking hackers*, volume 1. Addison-Wesley Reading, 2003.

[55] T Grudziecki, P Jacewicz, Ł JUSZCZYK, P Kijewski, and P Pawliński. Proactive detection of security incidents. *Honeypots. ENISA*, 2012.

[56] Abhishek Mairh, Debabrat Barik, Kanchan Verma, and Debasish Jena. Honeypot in network security: a survey. In *Proceedings of the 2011 international conference on communication, computing & security*, pages 600–605. ACM, 2011.

[57] William Wright, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. The sandbox for analysis: concepts and methods. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 801–810. ACM, 2006.

[58] Eirik Albrechtsen and Jan Hovden. Improving information security awareness and behaviour through dialogue, participation and collective reflection. an intervention study. *Computers & Security*, 29(4):432–445, 2010.

[59] Surbhi Gupta, Abhishek Singhal, and Akanksha Kapoor. A literature survey on social engineering attacks: Phishing attack. In *2016 international conference on computing, communication and automation (ICCCA)*, pages 537–540. IEEE, 2016.

[60] Jemal Abawajy. User preference of cyber security awareness delivery methods. *Behaviour & Information Technology*, 33(3):237–248, 2014.

[61] Ameya Sanzgiri and Dipankar Dasgupta. Classification of insider threat detection techniques. In *Proceedings of the 11th annual cyber and information security research conference*, page 25. ACM, 2016.

[62] Michael R Cunningham, John W Jones, and Brian W Dreschler. Personnel risk management assessment for newly emerging forms of employee crimes. *International Journal of Selection and Assessment*, 26(1):5–16, 2018.

[63] Lance Spitzner. Honeypots: Catching the insider threat. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 170–179. IEEE, 2003.

[64] Ruey Shiang Shaw, Charlie C Chen, Albert L Harris, and Hui-Jou Huang. The impact of information richness on information security awareness training effectiveness. *Computers & Education*, 52(1):92–100, 2009.

[65] Mark Wilson and Joan Hash. Building an information technology security awareness and training program. *NIST Special publication*, 800(50):1–39, 2003.

[66] Kenneth J Knapp, Thomas E Marshall, R Kelly Rainer Jr, and F Nelson Ford. Information security effectiveness: Conceptualization and validation of a theory. *International Journal of Information Security and Privacy (IJISP)*, 1(2):37–60, 2007.

[67] Nalin Asanka Gamagedara Arachchilage, Steve Love, and Konstantin Beznosov. Phishing threat avoidance behaviour: An empirical investigation. *Computers in Human Behavior*, 60:185–197, 2016.

[68] Ronald C Dodge Jr, Curtis Carver, and Aaron J Ferguson. Phishing for user security awareness. *computers & security*, 26(1):73–80, 2007.

[69] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao. Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Computing Surveys (CSUR)*, 39(1):3, 2007.

[70] Brad Arkin, Scott Stender, and Gary McGraw. Software penetration testing. *IEEE Security & Privacy*, 3(1):84–87, 2005.

[71] Hamad Al-Mohannadi, Irfan Awan, Jassim Al Hamar, Yousef Al Hamar, Mohammad Shah, and Ahmad Musa. Understanding awareness of cyber security threat among it employees. In *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 188–192. IEEE, 2018.

[72] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *Communications and Multimedia Security*, pages 63–72. Springer, 2014.

[73] Michael Ovelgönne, Tudor Dumitras, B Aditya Prakash, VS Subrahmanian, and Benjamin Wang. Understanding the relationship between human behavior and susceptibility to cyber attacks: A data-driven approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(4):51, 2017.

[74] Martin Hilbert. Big data for development: A review of promises and challenges. development policy review. *martinhilbert. net. Retrieved*, pages 10–07, 2015.

[75] Clinton Gormley and Zachary Tong. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine.* " O'Reilly Media, Inc.", 2015.

[76] Tomas Sochor and Matej Zuzcak. *Study of Internet Threats and Attack Methods Using Honeypots and Honeynets*, pages 118–127. Springer International Publishing, Cham, 2014.

[77] Donald Robinson. *Amazon Web Services Made Simple: Learn how Amazon EC2, S3, SimpleDB and SQS Web Services enables you to reach business goals faster.* Emereo Pty Ltd, 2008.

[78] Jeff Barr, Attila Narin, and Jinesh Varia. Building fault-tolerant applications on aws. *Amazon Web Services*, pages 1–15, 2011.

[79] Engin Kirda, Christopher Kruegel, Greg Banks, Giovanni Vigna, and Richard Kemmerer. Behavior-based spyware detection. In *Usenix Security Symposium*, page 694, 2006.

[80] Jinesh Varia, Sajee Mathew, et al. Overview of amazon web services. *Amazon Web Services*, pages 1–22, 2014.

[81] D. D. Kouvatsos. Networks security and performance engineering, August, 2015.

Appendix A

Assessment Questionnaire

A.1 Questionnaire1

Security Assessment Services

In this section, you will find the description of the most common assessment scenarios. These can be customized in many ways to meet a customer's needs. Each type of assessment takes varying amounts of time and is impacted by the number of targets (applications, servers, networks, etc.). The exact type of assessment should be determined in the "kickoff" meeting.

- **Network Based (Attack & Penetration)**

Penetration testing includes components of application vulnerability assessment, host vulnerability assessment, and security best practices. This type of test can be performed with or without detailed prior knowledge of the environment. When it is performed without prior knowledge additional steps will be taken to enumerate hosts and applications and to assess the ease with which any outsider could exploit publicly available information or social engineering to gain unauthorized access.

An attack and penetration test will answer questions like:

- How vulnerable is the network, host, and application(s) to attacks from the internet or intranet?
- Can an intruder obtain unauthorized access to critical resources?
- Are social engineering techniques effective?
- Are operational controls effective?

This would involve the ISO acting as an attacker and looking at the system as an outsider. The ISO would look for:

- Remotely exploitable vulnerabilities
- Patch levels (OS and Apps)
- Unnecessary services
- Weakness of encryption
- Weakness of authentication
- Etc.

- **Host Based**

This is an assessment of the health and security of given workstation or server. Automated scanning tools (e.g. Nessus) are the primary vehicle for this type of assessment. Additional hands-on inspection may also be necessary to assess conformance to security best practice.

This assessment will answer questions like:

- Is patching up to date?
- Are unnecessary services running?
- Are anti-virus/anti-malware signatures up to date?

This would involve the ISO acting as a Sys Admin and auditing the system and applications looking for:

- Locally exploitable vulnerabilities

- Patch levels (OS and Apps)
- Access rights
- Security best practices
- Etc.

- **Application**

This is an assessment of the functionality and resilience of the compiled application to known threats. This assessment focuses on the compiled and installed elements of the entire system: how the application components are deployed, communicate or otherwise interact with both the user and server environments.

Application scanning tools as well as manual testing with and without application credentials are used to perform this assessment. Typically some host, network, and general information security practices are assessed as part an application vulnerability assessment.

This assessment will answer questions like:

- Does the application expose the underlying servers and software to attack
- Can a malicious user access, modify, or destroy data or services within the system

This would involve the ISO auditing an application (typically web based) and looking for vulnerabilities like:

- SQL Injection
- Cross Site Scripting
- Cross Site Request Forgery
- Improper data sanitization
- Buffer overflows (limited)
- Mis-configured/weak authentication
- Etc.

- **Compliance**

This would involve the Information Security Office auditing (or assisting in the coordination of an audit if the ISO is not trained to conduct the specific audit) systems for compliance with specific regulations:

- HIPAA
- FERPA
- GLBA
- PCI

- **Physical Security Assessment**

This assessment typically involves interviews with key staff, documentation review, and an on-site visit to assess appropriate physical and environmental controls for safeguarding computing resources.

This assessment will answer questions like:

- Are there appropriate physical access controls in place for securing servers and desktop machines
- Are appropriate environmental controls in place to sustain critical computing infrastructure
- Are systems left logged in while staff are away

- **Enterprise Security Assessment**

This is a comprehensive study of the hosts, networks, applications, environmental controls, as well as policies and procedures.

Questionnaire:

The following questionnaire is necessary to guarantee the accuracy of the time estimates as well as the thoroughness of the assessment. Please fill out as much of the information as possible.

Basic Information

Name:	
Title:	
Telephone:	
Cell phone:	
Email address:	
All machines: • IP Addresses • OS • All machine names (DNS, WINS, Virtual Hosts, etc.)	
Is your organization subject to any specific regulatory requirements? (Examples – Sarbanes-Oxley, GLBA, HIPAA)	

Audit Information

Would you like the Information Security Office to perform a network-based assessment? (A&P)	
How many Internet-facing hosts do you want the Information Security Office to assess?	
Would you like the Information Security Office to perform a host-based assessment?	
Which hosts?	
Would you like the Information Security Office to perform compliance, physical or enterprise assessment?	
If compliance, which regulations? (HIPAA, FERPA, etc.)	
Would you like the Information Security Office to perform an application security assessment?	
Which specific applications? (URL, Application name, Installer, etc.)	
Would you like this tested with or without credentials?	
Would you like this tested with or without administrative credentials?	

Network Security Information

Has your organization ever been compromised (internally or externally)?	
List all IP address blocks registered to your organization. (Example – 12.34.56.x/24)	
List all the domain names registered to your organization. (Examples – acme.com; acmesales.com)	
Does your organization use a local Firewall(s)? If so, please list quantity and manufacturer(s) of firewall(s).	
Does your organization use a local Intrusion Detection System(s) (IDS)?	
Does your organization use a local Intrusion Prevention System(s) (IPS)?	
If your organization uses local IDS, do you use "host-based" IDS (HIDS) or "network-based" IDS (NIDS) or a combination of both? List the quantity of IDS (both HIDS and NIDS) and IPS devices, as well as the manufacturer(s).	
Do you use DMZ networks?	
Does your organization have any dedicated connections to other organization's networks (vendors, business partners)? If so, please list all dedicated connections to other networks.	
Does your organization use any Remote Access services? Specifically, what type of remote access services does your organization use (VPN or Dial-Up RAS)?	
How many employees use remote access services?	
Does your organization use site-to-site Virtual Private Network (VPN) tunnels? If so, how many site-to-site VPN tunnels are in use?	
Does your organization have any systems that use modems?	

System Information

How many Microsoft Windows NT/2000/2003 servers does your organization use?	
How many Unix servers (AIX, HPUX, Linux, Solaris, etc.) does your organization use? Please list specific distributions.	
List any servers with operating systems other than what is listed above. Please include quantities and list specific operating system versions/distributions.	
How many Microsoft Windows 2000/XP Professional clients does your organization use?	
List any clients with operating systems other than what is listed above. Please include quantities and list specific operating system versions/distributions.	
What Enterprise Resource Planning (ERP) application(s) does your organization use? (Examples – SAP, Peoplesoft, Oracle, JD Edwards) Please include a brief description of each.	
What E-commerce application(s) does your organization use? Please include a brief description of each.	
What database technologies does your organization use? (Examples – Oracle, Microsoft SQL, IBM DB2, MySQL) Please include a brief description of the purpose for each.	

Service Information

What services do you expose to the internet? (Examples: Web, Database, FTP, SSH, etc.)	
What services do you expose to the campus?	
What type of authentication do you use for your web services? (Examples: PubCookie, Windows Integrated, htaccess, etc.)	
What languages do you use for your web services? (Examples: PHP, Perl, Ruby, ASP, etc.)	
What antivirus application(s) do you use?	
Is your antivirus application implemented using a “managed” client/server architecture, or in a stand-alone configuration?	

A.2 Questionnaire2

Questionnaire on behaviour and knowledge IT employees.

The questions in this section are designed to collect information on your behavior and knowledge to respond to cyber security attack and how you have been supported during cyber-attack event.

A1. In terms of your security duty, do you have a defined checklist for your daily duty?

Mark up to three

□ Exploitable vulnerabilities
□ Patch levels (OS and Apps)
□ Unnecessary services
□ Weakness of encryption
□ Weakness of authentication
□ Access rights
□ Security best practices

A2. What the common attacks do you handle normally?

Mark up to three

□ Malware
□ Phishing
□ Password Attack
□ Denial-of-Service (DoS) Attacks
□ "Man in the Middle" (MITM)
□ Drive-By Downloads.
□ Rogue Software

A3. Do you recognize any of the terms during a cyber-attack?

Mark up to three

□ Hash values
□ IP Address
□ Domain Name
□ Network Artefact
□ Host Artefact
□ Attack tools. (e.g. Nmap, OWASP Zed)
□ Other Special techniques

A4. Which of the following Indicator of compromise (IOC) is the most/least difficult to trace in your environment?

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

	Hash	IP	DN	NA	HA	Tools	TTP
SOC							
NOC							
Incident Response							
Antivirus							
Border Security							
Business							
Research							
Home User							

Acronyms for table -
(SOC - Security Operation Center) – (NOC - Network Operation Center) – (IR - Incident Response) – (AV – Antivirus) – (BS - Border Security) – (Bus – Business) – (Res – Researcher) – (HA - Home User).

A5. What are the common __alerts__ do you handle daily?

IF NONE: MARK THIS BOX: □

1. **Security:** ____

2. **Performance:** ____

3. **Availability:** ____

A6. In case of __repetitive attacks__, what action do you take?

IF NONE: MARK THIS BOX: □

1. **Isolate and Contain:** ____

2. **Shutdown the system:** ____

3. **Contact security officer:** ____

A7. Do you have any procedure to follow in case of attacks?

□ Yes
□ No

A8. **Do you use any (firewall, IDS, IPS, router etc.) log data to understand activities in the network?**

□ Yes
□ No

A9. **Do you have an operation center to monitor all attacks?**

□ Yes
□ No

A10. **Is your workstation/Server implemented using a "managed" client/server architecture, or in a stand-alone to push the policy configuration and update?**

IF NONE: MARK THIS BOX: □

1. **Managed:** ______________________________

2. **Standalone:** ______________________________

A11. **Do you have DMZ for external and firewall for internal cross-site?**

□ Yes
□ No

IF YES:

A12. **Does it help in isolating or preventing the attack?**

□ Yes
□ No

Thank you for your time!

Appendix B

Honeypot Data

The following is an example of the data collected from kippo honeyot.

```
2016-12-09 13:08:28+0000 [-] root failed auth password
2016-12-09 13:08:28+0000 [-] unauthorized login:
2016-12-09 13:08:28+0000 [SSHService ssh-userauth on HoneyPotTransport,198713,116.31.116.16] root trying auth password
2016-12-09 13:08:28+0000 [SSHService ssh-userauth on HoneyPotTransport,198713,116.31.116.16] login attempt [root/yanxiong001] failed
2016-12-09 13:08:29+0000 [-] root failed auth password
2016-12-09 13:08:29+0000 [-] unauthorized login:
2016-12-09 13:08:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198713,116.31.116.16] root trying auth password
2016-12-09 13:08:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198713,116.31.116.16] login attempt [root/yaya123456] failed
2016-12-09 13:08:30+0000 [-] root failed auth password
2016-12-09 13:08:30+0000 [-] unauthorized login:
2016-12-09 13:08:30+0000 [HoneyPotTransport,198713,116.31.116.16] Got remote error, code 11
2016-12-09 13:08:30+0000 [HoneyPotTransport,198713,116.31.116.16] connection lost
2016-12-09 13:08:37+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:11732 (172.31.22.139:22) [session: 198714]
2016-12-09 13:08:37+0000 [HoneyPotTransport,198714,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:08:37+0000 [HoneyPotTransport,198714,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:08:37+0000 [HoneyPotTransport,198714,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:08:37+0000 [HoneyPotTransport,198714,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:08:38+0000 [HoneyPotTransport,198714,116.31.116.16] NEW KEYS
2016-12-09 13:08:38+0000 [HoneyPotTransport,198714,116.31.116.16] starting service
```

```
ssh-userauth
2016-12-09 13:08:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
root trying auth none
2016-12-09 13:08:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
root trying auth password
2016-12-09 13:08:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
login attempt [root/ygzfidc] failed
2016-12-09 13:08:39+0000 [-] root failed auth password
2016-12-09 13:08:39+0000 [-] unauthorized login:
2016-12-09 13:08:40+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
root trying auth password
2016-12-09 13:08:40+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
login attempt [root/yibaotuan#7758] failed
2016-12-09 13:08:41+0000 [-] root failed auth password
2016-12-09 13:08:41+0000 [-] unauthorized login:
2016-12-09 13:08:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
root trying auth password
2016-12-09 13:08:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198714,116.31.116.16]
login attempt [root/yinggang] failed
2016-12-09 13:08:42+0000 [-] root failed auth password
2016-12-09 13:08:42+0000 [-] unauthorized login:
2016-12-09 13:08:42+0000 [HoneyPotTransport,198714,116.31.116.16] Got remote error,
code 11
2016-12-09 13:08:42+0000 [HoneyPotTransport,198714,116.31.116.16] connection lost
2016-12-09 13:08:50+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection:
116.31.116.16:32953 (172.31.22.139:22) [session: 198715]
2016-12-09 13:08:50+0000 [HoneyPotTransport,198715,116.31.116.16] Remote SSH version:
SSH-2.0-PUTTY
2016-12-09 13:08:50+0000 [HoneyPotTransport,198715,116.31.116.16] kex alg, key alg:
diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:08:50+0000 [HoneyPotTransport,198715,116.31.116.16] outgoing: aes128-
ctr hmac-sha1 none
2016-12-09 13:08:50+0000 [HoneyPotTransport,198715,116.31.116.16] incoming: aes128-
ctr hmac-sha1 none
2016-12-09 13:08:50+0000 [HoneyPotTransport,198715,116.31.116.16] NEW KEYS
2016-12-09 13:08:51+0000 [HoneyPotTransport,198715,116.31.116.16] starting service
ssh-userauth
2016-12-09 13:08:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16]
root trying auth none
2016-12-09 13:08:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16]
root trying auth password
2016-12-09 13:08:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16]
login attempt [root/yinsha.com] failed
```

```
2016-12-09 13:08:52+0000 [-] root failed auth password
2016-12-09 13:08:52+0000 [-] unauthorized login:
2016-12-09 13:08:52+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16] root trying auth password
2016-12-09 13:08:52+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16] login attempt [root/yisence#@#1qaz] failed
2016-12-09 13:08:53+0000 [-] root failed auth password
2016-12-09 13:08:53+0000 [-] unauthorized login:
2016-12-09 13:08:53+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16] root trying auth password
2016-12-09 13:08:53+0000 [SSHService ssh-userauth on HoneyPotTransport,198715,116.31.116.16] login attempt [root/yone.com.cn] failed
2016-12-09 13:08:54+0000 [-] root failed auth password
2016-12-09 13:08:54+0000 [-] unauthorized login:
2016-12-09 13:08:55+0000 [HoneyPotTransport,198715,116.31.116.16] Got remote error, code 11
2016-12-09 13:08:55+0000 [HoneyPotTransport,198715,116.31.116.16] connection lost
2016-12-09 13:09:03+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:53502 (172.31.22.139:22) [session: 198716]
2016-12-09 13:09:03+0000 [HoneyPotTransport,198716,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:09:03+0000 [HoneyPotTransport,198716,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:09:03+0000 [HoneyPotTransport,198716,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:09:03+0000 [HoneyPotTransport,198716,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:09:03+0000 [HoneyPotTransport,198716,116.31.116.16] NEW KEYS
2016-12-09 13:09:04+0000 [HoneyPotTransport,198716,116.31.116.16] starting service ssh-userauth
2016-12-09 13:09:04+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] root trying auth none
2016-12-09 13:09:04+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] root trying auth password
2016-12-09 13:09:04+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] login attempt [root/yoyo123456] failed
2016-12-09 13:09:05+0000 [-] root failed auth password
2016-12-09 13:09:05+0000 [-] unauthorized login:
2016-12-09 13:09:05+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] root trying auth password
2016-12-09 13:09:05+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] login attempt [root/yoyo*idc2*] failed
2016-12-09 13:09:06+0000 [-] root failed auth password
```

```
2016-12-09 13:09:06+0000 [-] unauthorized login:
2016-12-09 13:09:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] root trying auth password
2016-12-09 13:09:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198716,116.31.116.16] login attempt [root/ytisp.net] failed
2016-12-09 13:09:07+0000 [-] root failed auth password
2016-12-09 13:09:07+0000 [-] unauthorized login:
2016-12-09 13:09:08+0000 [HoneyPotTransport,198716,116.31.116.16] Got remote error, code 11
2016-12-09 13:09:08+0000 [HoneyPotTransport,198716,116.31.116.16] connection lost
2016-12-09 13:09:15+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:19330 (172.31.22.139:22) [session: 198717]
2016-12-09 13:09:15+0000 [HoneyPotTransport,198717,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:09:15+0000 [HoneyPotTransport,198717,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:09:15+0000 [HoneyPotTransport,198717,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:09:15+0000 [HoneyPotTransport,198717,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:09:16+0000 [HoneyPotTransport,198717,116.31.116.16] NEW KEYS
2016-12-09 13:09:16+0000 [HoneyPotTransport,198717,116.31.116.16] starting service ssh-userauth
2016-12-09 13:09:16+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] root trying auth none
2016-12-09 13:09:16+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] root trying auth password
2016-12-09 13:09:16+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] login attempt [root/ywxl123456] failed
2016-12-09 13:09:17+0000 [-] root failed auth password
2016-12-09 13:09:17+0000 [-] unauthorized login:
2016-12-09 13:09:17+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] root trying auth password
2016-12-09 13:09:17+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] login attempt [root/yy123456] failed
2016-12-09 13:09:18+0000 [-] root failed auth password
2016-12-09 13:09:18+0000 [-] unauthorized login:
2016-12-09 13:09:19+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] root trying auth password
2016-12-09 13:09:19+0000 [SSHService ssh-userauth on HoneyPotTransport,198717,116.31.116.16] login attempt [root/yzdx!@#$%^] failed
2016-12-09 13:09:20+0000 [-] root failed auth password
2016-12-09 13:09:20+0000 [-] unauthorized login:
```

```
2016-12-09 13:09:20+0000 [HoneyPotTransport,198717,116.31.116.16] Got remote error, code 11 2016-12-09 13:09:20+0000 [HoneyPotTransport,198717,116.31.116.16] connection lost
2016-12-09 13:09:28+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:41627 (172.31.22.139:22) [session: 198718]
2016-12-09 13:09:28+0000 [HoneyPotTransport,198718,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:09:28+0000 [HoneyPotTransport,198718,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:09:28+0000 [HoneyPotTransport,198718,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:09:28+0000 [HoneyPotTransport,198718,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:09:29+0000 [HoneyPotTransport,198718,116.31.116.16] NEW KEYS
2016-12-09 13:09:29+0000 [HoneyPotTransport,198718,116.31.116.16] starting service ssh-userauth
2016-12-09 13:09:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] root trying auth none
2016-12-09 13:09:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] root trying auth password
2016-12-09 13:09:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] login attempt [root/yzdx110] failed
2016-12-09 13:09:30+0000 [-] root failed auth password
2016-12-09 13:09:30+0000 [-] unauthorized login:
2016-12-09 13:09:30+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] root trying auth password
2016-12-09 13:09:30+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] login attempt [root/yzdx123456] failed
2016-12-09 13:09:31+0000 [-] root failed auth password
2016-12-09 13:09:31+0000 [-] unauthorized login:
2016-12-09 13:09:32+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] root trying auth password
2016-12-09 13:09:32+0000 [SSHService ssh-userauth on HoneyPotTransport,198718,116.31.116.16] login attempt [root/yzdx23321] failed
2016-12-09 13:09:33+0000 [-] root failed auth password
2016-12-09 13:09:33+0000 [-] unauthorized login:
2016-12-09 13:09:33+0000 [HoneyPotTransport,198718,116.31.116.16] Got remote error, code 11
2016-12-09 13:09:33+0000 [HoneyPotTransport,198718,116.31.116.16] connection lost
2016-12-09 13:09:41+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:12700 (172.31.22.139:22) [session: 198719]
2016-12-09 13:09:41+0000 [HoneyPotTransport,198719,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
```

```
2016-12-09 13:09:41+0000 [HoneyPotTransport,198719,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:09:41+0000 [HoneyPotTransport,198719,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:09:41+0000 [HoneyPotTransport,198719,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:09:41+0000 [HoneyPotTransport,198719,116.31.116.16] NEW KEYS
2016-12-09 13:09:42+0000 [HoneyPotTransport,198719,116.31.116.16] starting service ssh-userauth
2016-12-09 13:09:42+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] root trying auth none
2016-12-09 13:09:42+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] root trying auth password
2016-12-09 13:09:42+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] login attempt [root/yzdx654321] failed
2016-12-09 13:09:43+0000 [-] root failed auth password
2016-12-09 13:09:43+0000 [-] unauthorized login:
2016-12-09 13:09:43+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] root trying auth password
2016-12-09 13:09:43+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] login attempt [root/yzdxidc] failed
2016-12-09 13:09:44+0000 [-] root failed auth password
2016-12-09 13:09:44+0000 [-] unauthorized login:
2016-12-09 13:09:45+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] root trying auth password
2016-12-09 13:09:45+0000 [SSHService ssh-userauth on HoneyPotTransport,198719,116.31.116.16] login attempt [root/yzidc!#&(38] failed
2016-12-09 13:09:46+0000 [-] root failed auth password
2016-12-09 13:09:46+0000 [-] unauthorized login:
2016-12-09 13:09:46+0000 [HoneyPotTransport,198719,116.31.116.16] Got remote error, code
2016-12-09 13:09:46+0000 [HoneyPotTransport,198719,116.31.116.16] connection lost
2016-12-09 13:09:53+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:32723 (172.31.22.139:22) [session: 198720]
2016-12-09 13:09:53+0000 [HoneyPotTransport,198720,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:09:53+0000 [HoneyPotTransport,198720,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:09:53+0000 [HoneyPotTransport,198720,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:09:53+0000 [HoneyPotTransport,198720,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:09:54+0000 [HoneyPotTransport,198720,116.31.116.16] NEW KEYS
```

```
2016-12-09 13:09:54+0000 [HoneyPotTransport,198720,116.31.116.16] starting service ssh-userauth
2016-12-09 13:09:54+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] root trying auth none
2016-12-09 13:09:54+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] root trying auth password
2016-12-09 13:09:54+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] login attempt [root/yzwl!@#$%ˆ] failed
2016-12-09 13:09:55+0000 [-] root failed auth password
2016-12-09 13:09:55+0000 [-] unauthorized login:
2016-12-09 13:09:56+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] root trying auth password
2016-12-09 13:09:56+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] login attempt [root/yzwl123456] failed
2016-12-09 13:09:57+0000 [-] root failed auth password
2016-12-09 13:09:57+0000 [-] unauthorized login:
2016-12-09 13:09:57+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] root trying auth password
2016-12-09 13:09:57+0000 [SSHService ssh-userauth on HoneyPotTransport,198720,116.31.116.16] login attempt [root/zcidc.com] failed
2016-12-09 13:09:58+0000 [-] root failed auth password
2016-12-09 13:09:58+0000 [-] unauthorized login:
2016-12-09 13:09:58+0000 [HoneyPotTransport,198720,116.31.116.16] Got remote error, code 11
2016-12-09 13:09:58+0000 [HoneyPotTransport,198720,116.31.116.16] connection lost
2016-12-09 13:10:06+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:53886 (172.31.22.139:22) [session: 198721]
2016-12-09 13:10:06+0000 [HoneyPotTransport,198721,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:10:06+0000 [HoneyPotTransport,198721,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:10:06+0000 [HoneyPotTransport,198721,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:10:06+0000 [HoneyPotTransport,198721,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:10:06+0000 [HoneyPotTransport,198721,116.31.116.16] NEW KEYS
2016-12-09 13:10:07+0000 [HoneyPotTransport,198721,116.31.116.16] starting service ssh-userauth
2016-12-09 13:10:07+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16] root trying auth none
2016-12-09 13:10:07+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16] root trying auth password
2016-12-09 13:10:07+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16]
```

```
login attempt [root/zero] failed
2016-12-09 13:10:08+0000 [-] root failed auth password
2016-12-09 13:10:08+0000 [-] unauthorized login:
2016-12-09 13:10:08+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16]
root trying auth password
2016-12-09 13:10:08+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16]
login attempt [root/zero2010] failed
2016-12-09 13:10:09+0000 [-] root failed auth password
2016-12-09 13:10:09+0000 [-] unauthorized login:
2016-12-09 13:10:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16]
root trying auth password
2016-12-09 13:10:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198721,116.31.116.16]
login attempt [root/zero2011] failed
2016-12-09 13:10:10+0000 [-] root failed auth password
2016-12-09 13:10:10+0000 [-] unauthorized login:
2016-12-09 13:10:11+0000 [HoneyPotTransport,198721,116.31.116.16] Got remote error,
code 11
2016-12-09 13:10:11+0000 [HoneyPotTransport,198721,116.31.116.16] connection lost
2016-12-09 13:10:18+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection:
116.31.116.16:20146 (172.31.22.139:22) [session: 198722]
2016-12-09 13:10:18+0000 [HoneyPotTransport,198722,116.31.116.16] Remote SSH version:
SSH-2.0-PUTTY
2016-12-09 13:10:18+0000 [HoneyPotTransport,198722,116.31.116.16] kex alg, key alg:
diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:10:18+0000 [HoneyPotTransport,198722,116.31.116.16] outgoing: aes128-
ctr hmac-sha1 none
2016-12-09 13:10:18+0000 [HoneyPotTransport,198722,116.31.116.16] incoming: aes128-
ctr hmac-sha1 none
2016-12-09 13:10:19+0000 [HoneyPotTransport,198722,116.31.116.16] NEW KEYS
2016-12-09 13:10:19+0000 [HoneyPotTransport,198722,116.31.116.16] starting service
ssh-userauth
2016-12-09 13:10:19+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16]
root trying auth none
2016-12-09 13:10:19+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16]
root trying auth password
2016-12-09 13:10:19+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16]
login attempt [root/zero.com] failed
2016-12-09 13:10:20+0000 [-] root failed auth password
2016-12-09 13:10:20+0000 [-] unauthorized login:
2016-12-09 13:10:21+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16]
root trying auth password
2016-12-09 13:10:21+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16]
login attempt [root/zgl6650000] failed
```

```
2016-12-09 13:10:22+0000 [-] root failed auth password
2016-12-09 13:10:22+0000 [-] unauthorized login:
2016-12-09 13:10:23+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16] root trying auth password
2016-12-09 13:10:23+0000 [SSHService ssh-userauth on HoneyPotTransport,198722,116.31.116.16] login attempt [root/zhang123456] failed
2016-12-09 13:10:24+0000 [-] root failed auth password
2016-12-09 13:10:24+0000 [-] unauthorized login:
2016-12-09 13:10:24+0000 [HoneyPotTransport,198722,116.31.116.16] Got remote error, code 11
2016-12-09 13:10:24+0000 [HoneyPotTransport,198722,116.31.116.16] connection lost
2016-12-09 13:10:31+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:38969 (172.31.22.139:22) [session: 198723]
2016-12-09 13:10:31+0000 [HoneyPotTransport,198723,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:10:31+0000 [HoneyPotTransport,198723,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:10:31+0000 [HoneyPotTransport,198723,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:10:31+0000 [HoneyPotTransport,198723,116.31.116.16] incoming: aes128-ctr hmac-sha1 none

    2016-12-09 13:10:31+0000 [HoneyPotTransport,198723,116.31.116.16] NEW KEYS
2016-12-09 13:10:32+0000 [HoneyPotTransport,198723,116.31.116.16] starting service ssh-userauth
2016-12-09 13:10:32+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] root trying auth none
2016-12-09 13:10:32+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] root trying auth password
2016-12-09 13:10:32+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] login attempt [root/zhangqiang] failed
2016-12-09 13:10:33+0000 [-] root failed auth password
2016-12-09 13:10:33+0000 [-] unauthorized login:
2016-12-09 13:10:34+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] root trying auth password
2016-12-09 13:10:34+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] login attempt [root/zhenglong] failed
2016-12-09 13:10:35+0000 [-] root failed auth password
2016-12-09 13:10:35+0000 [-] unauthorized login:
2016-12-09 13:10:35+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] root trying auth password
2016-12-09 13:10:35+0000 [SSHService ssh-userauth on HoneyPotTransport,198723,116.31.116.16] login attempt [root/zhenglongjifang] failed
```

```
2016-12-09 13:10:36+0000 [-] root failed auth password
2016-12-09 13:10:36+0000 [-] unauthorized login:
2016-12-09 13:10:36+0000 [HoneyPotTransport,198723,116.31.116.16] Got remote error, code 11
2016-12-09 13:10:36+0000 [HoneyPotTransport,198723,116.31.116.16] connection lost
2016-12-09 13:10:44+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:60052 (172.31.22.139:22) [session: 198724]
2016-12-09 13:10:44+0000 [HoneyPotTransport,198724,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:10:44+0000 [HoneyPotTransport,198724,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:10:44+0000 [HoneyPotTransport,198724,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:10:44+0000 [HoneyPotTransport,198724,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:10:44+0000 [HoneyPotTransport,198724,116.31.116.16] NEW KEYS
2016-12-09 13:10:45+0000 [HoneyPotTransport,198724,116.31.116.16] starting service ssh-userauth
2016-12-09 13:10:45+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] root trying auth none
2016-12-09 13:10:45+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] root trying auth password
2016-12-09 13:10:45+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] login attempt [root/zhiban_2006] failed
2016-12-09 13:10:46+0000 [-] root failed auth password
2016-12-09 13:10:46+0000 [-] unauthorized login:
2016-12-09 13:10:46+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] root trying auth password
2016-12-09 13:10:46+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] login attempt [root/zhiban_2007] failed
2016-12-09 13:10:47+0000 [-] root failed auth password
2016-12-09 13:10:47+0000 [-] unauthorized login:
2016-12-09 13:10:47+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] root trying auth password
2016-12-09 13:10:47+0000 [SSHService ssh-userauth on HoneyPotTransport,198724,116.31.116.16] login attempt [root/zhiban_2008] failed
2016-12-09 13:10:48+0000 [-] root failed auth password
2016-12-09 13:10:48+0000 [-] unauthorized login:
2016-12-09 13:10:49+0000 [HoneyPotTransport,198724,116.31.116.16] Got remote error, code 11
2016-12-09 13:10:49+0000 [HoneyPotTransport,198724,116.31.116.16] connection lost
2016-12-09 13:10:56+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:25726 (172.31.22.139:22) [session: 198725]
```

```
2016-12-09 13:10:56+0000 [HoneyPotTransport,198725,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:10:56+0000 [HoneyPotTransport,198725,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:10:56+0000 [HoneyPotTransport,198725,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:10:56+0000 [HoneyPotTransport,198725,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:10:57+0000 [HoneyPotTransport,198725,116.31.116.16] NEW KEYS
2016-12-09 13:10:57+0000 [HoneyPotTransport,198725,116.31.116.16] starting service ssh-userauth
2016-12-09 13:10:57+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] root trying auth none
2016-12-09 13:10:57+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] root trying auth password
2016-12-09 13:10:57+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] login attempt [root/zhiban_2009] failed
2016-12-09 13:10:58+0000 [-] root failed auth password
2016-12-09 13:10:58+0000 [-] unauthorized login:
2016-12-09 13:10:58+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] root trying auth password
2016-12-09 13:10:58+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] login attempt [root/zhiban_2010] failed
2016-12-09 13:10:59+0000 [-] root failed auth password
2016-12-09 13:10:59+0000 [-] unauthorized login:
2016-12-09 13:10:59+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] root trying auth password
2016-12-09 13:10:59+0000 [SSHService ssh-userauth on HoneyPotTransport,198725,116.31.116.16] login attempt [root/zhuanqian] failed
2016-12-09 13:11:00+0000 [-] root failed auth password
2016-12-09 13:11:00+0000 [-] unauthorized login:
2016-12-09 13:11:01+0000 [HoneyPotTransport,198725,116.31.116.16] Got remote error, code 11
2016-12-09 13:11:01+0000 [HoneyPotTransport,198725,116.31.116.16] connection lost
2016-12-09 13:11:08+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:45195 (172.31.22.139:22) [session: 198726]
2016-12-09 13:11:08+0000 [HoneyPotTransport,198726,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:11:08+0000 [HoneyPotTransport,198726,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:11:08+0000 [HoneyPotTransport,198726,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:11:08+0000 [HoneyPotTransport,198726,116.31.116.16] incoming: aes128-
```

```
ctr hmac-sha1 none
2016-12-09 13:11:09+0000 [HoneyPotTransport,198726,116.31.116.16] NEW KEYS
2016-12-09 13:11:09+0000 [HoneyPotTransport,198726,116.31.116.16] starting service
ssh-userauth
2016-12-09 13:11:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
root trying auth none
2016-12-09 13:11:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
root trying auth password
2016-12-09 13:11:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
login attempt [root/zj8812345] failed
2016-12-09 13:11:10+0000 [-] root failed auth password
2016-12-09 13:11:10+0000 [-] unauthorized login:
2016-12-09 13:11:11+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
root trying auth password
2016-12-09 13:11:11+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
login attempt [root/zjaomao888] failed
2016-12-09 13:11:12+0000 [-] root failed auth password
2016-12-09 13:11:12+0000 [-] unauthorized login:
2016-12-09 13:11:12+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
root trying auth password
2016-12-09 13:11:12+0000 [SSHService ssh-userauth on HoneyPotTransport,198726,116.31.116.16]
login attempt [root/zjidc] failed
2016-12-09 13:11:13+0000 [-] root failed auth password
2016-12-09 13:11:13+0000 [-] unauthorized login:
2016-12-09 13:11:13+0000 [HoneyPotTransport,198726,116.31.116.16] Got remote error,
code 11
2016-12-09 13:11:13+0000 [HoneyPotTransport,198726,116.31.116.16] connection lost
2016-12-09 13:11:21+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection:
116.31.116.16:11129 (172.31.22.139:22) [session: 198727]
2016-12-09 13:11:21+0000 [HoneyPotTransport,198727,116.31.116.16] Remote SSH ver-
sion: SSH-2.0-PUTTY
2016-12-09 13:11:21+0000 [HoneyPotTransport,198727,116.31.116.16] kex alg, key alg:
diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:11:21+0000 [HoneyPotTransport,198727,116.31.116.16] outgoing: aes128-
ctr hmac-sha1 none
2016-12-09 13:11:21+0000 [HoneyPotTransport,198727,116.31.116.16] incoming: aes128-
ctr hmac-sha1 none
2016-12-09 13:11:21+0000 [HoneyPotTransport,198727,116.31.116.16] NEW KEYS
2016-12-09 13:11:22+0000 [HoneyPotTransport,198727,116.31.116.16] starting service
ssh-userauth
2016-12-09 13:11:22+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16]
root trying auth none
2016-12-09 13:11:22+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16]
```

```
root trying auth password
2016-12-09 13:11:22+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16] login attempt [root/zouying] failed
2016-12-09 13:11:23+0000 [-] root failed auth password
2016-12-09 13:11:23+0000 [-] unauthorized login:
2016-12-09 13:11:23+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16] root trying auth password
2016-12-09 13:11:23+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16] login attempt [root/zrway.com] failed
2016-12-09 13:11:24+0000 [-] root failed auth password
2016-12-09 13:11:24+0000 [-] unauthorized login:
2016-12-09 13:11:24+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16] root trying auth password
2016-12-09 13:11:24+0000 [SSHService ssh-userauth on HoneyPotTransport,198727,116.31.116.16] login attempt [root/zsidc!sx] failed
2016-12-09 13:11:25+0000 [-] root failed auth password
2016-12-09 13:11:25+0000 [-] unauthorized login:
2016-12-09 13:11:26+0000 [HoneyPotTransport,198727,116.31.116.16] Got remote error, code 11
2016-12-09 13:11:26+0000 [HoneyPotTransport,198727,116.31.116.16] connection lost
2016-12-09 13:11:33+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:31084 (172.31.22.139:22) [session: 198728]
2016-12-09 13:11:33+0000 [HoneyPotTransport,198728,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:11:33+0000 [HoneyPotTransport,198728,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:11:33+0000 [HoneyPotTransport,198728,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:11:33+0000 [HoneyPotTransport,198728,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:11:34+0000 [HoneyPotTransport,198728,116.31.116.16] NEW KEYS
2016-12-09 13:11:34+0000 [HoneyPotTransport,198728,116.31.116.16] starting service ssh-userauth
2016-12-09 13:11:34+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] root trying auth none
2016-12-09 13:11:34+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] root trying auth password
2016-12-09 13:11:34+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] login attempt [root/zxcasdqwe] failed
2016-12-09 13:11:35+0000 [-] root failed auth password
2016-12-09 13:11:35+0000 [-] unauthorized login:
2016-12-09 13:11:35+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] root trying auth password
```

```
2016-12-09 13:11:35+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] login attempt [root/zxidc_654321] failed
2016-12-09 13:11:36+0000 [-] root failed auth password
2016-12-09 13:11:36+0000 [-] unauthorized login:
2016-12-09 13:11:37+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] root trying auth password
2016-12-09 13:11:37+0000 [SSHService ssh-userauth on HoneyPotTransport,198728,116.31.116.16] login attempt [root/zxwok2011] failed
2016-12-09 13:11:38+0000 [-] root failed auth password
2016-12-09 13:11:38+0000 [-] unauthorized login:
2016-12-09 13:11:38+0000 [HoneyPotTransport,198728,116.31.116.16] Got remote error, code 11
2016-12-09 13:11:38+0000 [HoneyPotTransport,198728,116.31.116.16] connection lost
2016-12-09 13:11:45+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:50659 (172.31.22.139:22) [session: 198729]
2016-12-09 13:11:45+0000 [HoneyPotTransport,198729,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:11:45+0000 [HoneyPotTransport,198729,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:11:45+0000 [HoneyPotTransport,198729,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:11:45+0000 [HoneyPotTransport,198729,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:11:46+0000 [HoneyPotTransport,198729,116.31.116.16] NEW KEYS
2016-12-09 13:11:46+0000 [HoneyPotTransport,198729,116.31.116.16] starting service ssh-userauth
2016-12-09 13:11:46+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16] root trying auth none
2016-12-09 13:11:46+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16] root trying auth password
2016-12-09 13:11:46+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16] login attempt [root/zzidc63335503] failed
2016-12-09 13:11:47+0000 [-] root failed auth password
2016-12-09 13:11:47+0000 [-] unauthorized login:
2016-12-09 13:11:47+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16] root trying auth password
2016-12-09 13:11:47+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16] login attempt [root/zzidc.com.cn] failed
2016-12-09 13:11:48+0000 [-] root failed auth password
2016-12-09 13:11:48+0000 [-] unauthorized login:
2016-12-09 13:11:49+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16] root trying auth password
2016-12-09 13:11:49+0000 [SSHService ssh-userauth on HoneyPotTransport,198729,116.31.116.16]
```

```
login attempt [root/zzxcc123] failed
2016-12-09 13:11:50+0000 [-] root failed auth password
2016-12-09 13:11:50+0000 [-] unauthorized login:
2016-12-09 13:11:50+0000 [HoneyPotTransport,198729,116.31.116.16] Got remote error, code 11
2016-12-09 13:11:50+0000 [HoneyPotTransport,198729,116.31.116.16] connection lost
2016-12-09 13:11:58+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:16725 (172.31.22.139:22) [session: 198730]
2016-12-09 13:11:58+0000 [HoneyPotTransport,198730,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:11:58+0000 [HoneyPotTransport,198730,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:11:58+0000 [HoneyPotTransport,198730,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:11:58+0000 [HoneyPotTransport,198730,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:11:58+0000 [HoneyPotTransport,198730,116.31.116.16] NEW KEYS
2016-12-09 13:11:59+0000 [HoneyPotTransport,198730,116.31.116.16] starting service ssh-userauth
2016-12-09 13:11:59+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] root trying auth none
2016-12-09 13:11:59+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] root trying auth password
2016-12-09 13:11:59+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] login attempt [root/zzxxcc123] failed
2016-12-09 13:12:00+0000 [-] root failed auth password
2016-12-09 13:12:00+0000 [-] unauthorized login:
2016-12-09 13:12:00+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] root trying auth password
2016-12-09 13:12:00+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] login attempt [root/0321] failed
2016-12-09 13:12:01+0000 [-] root failed auth password
2016-12-09 13:12:01+0000 [-] unauthorized login:
2016-12-09 13:12:01+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] root trying auth password
2016-12-09 13:12:01+0000 [SSHService ssh-userauth on HoneyPotTransport,198730,116.31.116.16] login attempt [root/1000] failed
2016-12-09 13:12:02+0000 [-] root failed auth password
2016-12-09 13:12:02+0000 [-] unauthorized login:
2016-12-09 13:12:03+0000 [HoneyPotTransport,198730,116.31.116.16] Got remote error, code 11
2016-12-09 13:12:03+0000 [HoneyPotTransport,198730,116.31.116.16] connection lost
2016-12-09 13:12:10+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection:
```

```
116.31.116.16:36792 (172.31.22.139:22) [session: 198731]
2016-12-09 13:12:10+0000 [HoneyPotTransport,198731,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:12:10+0000 [HoneyPotTransport,198731,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:12:10+0000 [HoneyPotTransport,198731,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:12:10+0000 [HoneyPotTransport,198731,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:12:10+0000 [HoneyPotTransport,198731,116.31.116.16] NEW KEYS
2016-12-09 13:12:11+0000 [HoneyPotTransport,198731,116.31.116.16] starting service ssh-userauth
2016-12-09 13:12:11+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] root trying auth none
2016-12-09 13:12:11+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] root trying auth password
2016-12-09 13:12:11+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] login attempt [root/1008] failed
2016-12-09 13:12:12+0000 [-] root failed auth password
2016-12-09 13:12:12+0000 [-] unauthorized login:
2016-12-09 13:12:12+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] root trying auth password
2016-12-09 13:12:12+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] login attempt [root/10086] failed
2016-12-09 13:12:13+0000 [-] root failed auth password
2016-12-09 13:12:13+0000 [-] unauthorized login:
2016-12-09 13:12:13+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] root trying auth password
2016-12-09 13:12:13+0000 [SSHService ssh-userauth on HoneyPotTransport,198731,116.31.116.16] login attempt [root/1020] failed
2016-12-09 13:12:14+0000 [-] root failed auth password
2016-12-09 13:12:14+0000 [-] unauthorized login:
2016-12-09 13:12:15+0000 [HoneyPotTransport,198731,116.31.116.16] Got remote error, code 11
2016-12-09 13:12:15+0000 [HoneyPotTransport,198731,116.31.116.16] connection lost
2016-12-09 13:12:22+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:57215 (172.31.22.139:22) [session: 198732]
2016-12-09 13:12:22+0000 [HoneyPotTransport,198732,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:12:22+0000 [HoneyPotTransport,198732,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:12:22+0000 [HoneyPotTransport,198732,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
```

```
2016-12-09 13:12:22+0000 [HoneyPotTransport,198732,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:12:23+0000 [HoneyPotTransport,198732,116.31.116.16] NEW KEYS
2016-12-09 13:12:24+0000 [HoneyPotTransport,198732,116.31.116.16] starting service ssh-userauth
2016-12-09 13:12:24+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] root trying auth none
2016-12-09 13:12:24+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] root trying auth password
2016-12-09 13:12:24+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] login attempt [root/11223311] failed
2016-12-09 13:12:25+0000 [-] root failed auth password
2016-12-09 13:12:25+0000 [-] unauthorized login:
2016-12-09 13:12:25+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] root trying auth password
2016-12-09 13:12:25+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] login attempt [root/121121] failed
2016-12-09 13:12:26+0000 [-] root failed auth password
2016-12-09 13:12:26+0000 [-] unauthorized login:
2016-12-09 13:12:26+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] root trying auth password
2016-12-09 13:12:26+0000 [SSHService ssh-userauth on HoneyPotTransport,198732,116.31.116.16] login attempt [root/12300] failed
2016-12-09 13:12:27+0000 [-] root failed auth password
2016-12-09 13:12:27+0000 [-] unauthorized login:
2016-12-09 13:12:28+0000 [HoneyPotTransport,198732,116.31.116.16] Got remote error, code 11
2016-12-09 13:12:28+0000 [HoneyPotTransport,198732,116.31.116.16] connection lost
2016-12-09 13:12:35+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:22513 (172.31.22.139:22) [session: 198733]
2016-12-09 13:12:35+0000 [HoneyPotTransport,198733,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:12:35+0000 [HoneyPotTransport,198733,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:12:35+0000 [HoneyPotTransport,198733,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:12:35+0000 [HoneyPotTransport,198733,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:12:35+0000 [HoneyPotTransport,198733,116.31.116.16] NEW KEYS
2016-12-09 13:12:36+0000 [HoneyPotTransport,198733,116.31.116.16] starting service ssh-userauth
2016-12-09 13:12:36+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] root trying auth none
```

```
2016-12-09 13:12:36+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] root trying auth password
2016-12-09 13:12:36+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] login attempt [root/123000] failed
2016-12-09 13:12:37+0000 [-] root failed auth password
2016-12-09 13:12:37+0000 [-] unauthorized login:
2016-12-09 13:12:37+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] root trying auth password
2016-12-09 13:12:37+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] login attempt [root/12345!] failed
2016-12-09 13:12:38+0000 [-] root failed auth password
2016-12-09 13:12:38+0000 [-] unauthorized login:
2016-12-09 13:12:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] root trying auth password
2016-12-09 13:12:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198733,116.31.116.16] login attempt [root/123456xxx] failed
2016-12-09 13:12:39+0000 [-] root failed auth password
2016-12-09 13:12:39+0000 [-] unauthorized login:
2016-12-09 13:12:40+0000 [HoneyPotTransport,198733,116.31.116.16] Got remote error, code 11
2016-12-09 13:12:40+0000 [HoneyPotTransport,198733,116.31.116.16] connection lost
2016-12-09 13:12:47+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:43060 (172.31.22.139:22) [session: 198734]
2016-12-09 13:12:48+0000 [HoneyPotTransport,198734,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:12:48+0000 [HoneyPotTransport,198734,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:12:48+0000 [HoneyPotTransport,198734,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:12:48+0000 [HoneyPotTransport,198734,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:12:48+0000 [HoneyPotTransport,198734,116.31.116.16] NEW KEYS
2016-12-09 13:12:48+0000 [HoneyPotTransport,198734,116.31.116.16] starting service ssh-userauth
2016-12-09 13:12:48+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16] root trying auth none
2016-12-09 13:12:49+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16] root trying auth password
2016-12-09 13:12:49+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16] login attempt [root/123456z] failed
2016-12-09 13:12:50+0000 [-] root failed auth password
2016-12-09 13:12:50+0000 [-] unauthorized login:
2016-12-09 13:12:50+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16]
```

```
root trying auth password
2016-12-09 13:12:50+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16]
login attempt [root/123xxx] failed
2016-12-09 13:12:51+0000 [-] root failed auth password
2016-12-09 13:12:51+0000 [-] unauthorized login:
2016-12-09 13:12:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16]
root trying auth password
2016-12-09 13:12:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198734,116.31.116.16]
login attempt [root/123zxc] failed
2016-12-09 13:12:52+0000 [-] root failed auth password
2016-12-09 13:12:52+0000 [-] unauthorized login:
2016-12-09 13:12:52+0000 [HoneyPotTransport,198734,116.31.116.16] Got remote error,
code 11
2016-12-09 13:12:52+0000 [HoneyPotTransport,198734,116.31.116.16] connection lost
2016-12-09 13:116.31.116.1613:00+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New
connection: 116.31.116.16:10190 (172.31.22.139:22) [session: 198735]
2016-12-09 13:13:00+0000 [HoneyPotTransport,198735,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:13:00+0000 [HoneyPotTransport,198735,116.31.116.16] kex alg, key alg:
diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:13:00+0000 [HoneyPotTransport,198735,116.31.116.16] outgoing: aes128-
ctr hmac-sha1 none
2016-12-09 13:13:00+0000 [HoneyPotTransport,198735,116.31.116.16] incoming: aes128-
ctr hmac-sha1 none
2016-12-09 13:13:01+0000 [HoneyPotTransport,198735,116.31.116.16] NEW KEYS
2016-12-09 13:13:01+0000 [HoneyPotTransport,198735,116.31.116.16] starting service
ssh-userauth
2016-12-09 13:13:01+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16]
root trying auth none
2016-12-09 13:13:01+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16]
root trying auth password
2016-12-09 13:13:01+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16]
login attempt [root/1314521] failed
2016-12-09 13:13:02+0000 [-] root failed auth password
2016-12-09 13:13:02+0000 [-] unauthorized login:
2016-12-09 13:13:02+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16]
root trying auth password
2016-12-09 13:13:02+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16]
login attempt [root/171823] failed
2016-12-09 13:13:03+0000 [-] root failed auth password
2016-12-09 13:13:03+0000 [-] unauthorized login:
2016-12-09 13:13:04+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16]
root trying auth password
```

```
2016-12-09 13:13:04+0000 [SSHService ssh-userauth on HoneyPotTransport,198735,116.31.116.16] login attempt [root/172839] failed
2016-12-09 13:13:05+0000 [-] root failed auth password
2016-12-09 13:13:05+0000 [-] unauthorized login:
2016-12-09 13:13:05+0000 [HoneyPotTransport,198735,116.31.116.16] Got remote error, code 11
2016-12-09 13:13:05+0000 [HoneyPotTransport,198735,116.31.116.16] connection lost
2016-12-09 13:13:12+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:31299 (172.31.22.139:22) [session: 198736]
2016-12-09 13:13:13+0000 [HoneyPotTransport,198736,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:13:13+0000 [HoneyPotTransport,198736,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:13:13+0000 [HoneyPotTransport,198736,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:13:13+0000 [HoneyPotTransport,198736,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:13:13+0000 [HoneyPotTransport,198736,116.31.116.16] NEW KEYS
2016-12-09 13:13:13+0000 [HoneyPotTransport,198736,116.31.116.16] starting service ssh-userauth
2016-12-09 13:13:13+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] root trying auth none
2016-12-09 13:13:14+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] root trying auth password
2016-12-09 13:13:14+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] login attempt [root/19920925] failed
2016-12-09 13:13:15+0000 [-] root failed auth password
2016-12-09 13:13:15+0000 [-] unauthorized login:
2016-12-09 13:13:15+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] root trying auth password
2016-12-09 13:13:15+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] login attempt [root/1999] failed
2016-12-09 13:13:16+0000 [-] root failed auth password
2016-12-09 13:13:16+0000 [-] unauthorized login:
2016-12-09 13:13:17+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] root trying auth password
2016-12-09 13:13:17+0000 [SSHService ssh-userauth on HoneyPotTransport,198736,116.31.116.16] login attempt [root/1qasw2] failed
2016-12-09 13:13:18+0000 [-] root failed auth password
2016-12-09 13:13:18+0000 [-] unauthorized login:
2016-12-09 13:13:18+0000 [HoneyPotTransport,198736,116.31.116.16] Got remote error, code 11
2016-12-09 13:13:18+0000 [HoneyPotTransport,198736,116.31.116.16] connection lost
```

```
2016-12-09 13:13:24+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:50318 (172.31.22.139:22) [session: 198737]
2016-12-09 13:13:25+0000 [HoneyPotTransport,198737,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:13:25+0000 [HoneyPotTransport,198737,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:13:25+0000 [HoneyPotTransport,198737,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:13:25+0000 [HoneyPotTransport,198737,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:13:25+0000 [HoneyPotTransport,198737,116.31.116.16] NEW KEYS
2016-12-09 13:13:25+0000 [HoneyPotTransport,198737,116.31.116.16] starting service ssh-userauth
2016-12-09 13:13:26+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] root trying auth none
2016-12-09 13:13:26+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] root trying auth password
2016-12-09 13:13:26+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] login attempt [root/1qazxcvb] failed
2016-12-09 13:13:27+0000 [-] root failed auth password
2016-12-09 13:13:27+0000 [-] unauthorized login:
2016-12-09 13:13:27+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] root trying auth password
2016-12-09 13:13:27+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] login attempt [root/1qazxsw23edc] failed
2016-12-09 13:13:28+0000 [-] root failed auth password
2016-12-09 13:13:28+0000 [-] unauthorized login:
2016-12-09 13:13:28+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] root trying auth password
2016-12-09 13:13:28+0000 [SSHService ssh-userauth on HoneyPotTransport,198737,116.31.116.16] login attempt [root/21vianet123] failed
2016-12-09 13:13:29+0000 [-] root failed auth password
2016-12-09 13:13:29+0000 [-] unauthorized login:
2016-12-09 13:13:29+0000 [HoneyPotTransport,198737,116.31.116.16] Got remote error, code 11
2016-12-09 13:13:29+0000 [HoneyPotTransport,198737,116.31.116.16] connection lost
2016-12-09 13:13:37+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:16222 (172.31.22.139:22) [session: 198738]
2016-12-09 13:13:37+0000 [HoneyPotTransport,198738,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:13:37+0000 [HoneyPotTransport,198738,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:13:37+0000 [HoneyPotTransport,198738,116.31.116.16] outgoing: aes128-
```

```
ctr hmac-sha1 none
2016-12-09 13:13:37+0000 [HoneyPotTransport,198738,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:13:38+0000 [HoneyPotTransport,198738,116.31.116.16] NEW KEYS
2016-12-09 13:13:38+0000 [HoneyPotTransport,198738,116.31.116.16] starting service ssh-userauth
2016-12-09 13:13:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] root trying auth none
2016-12-09 13:13:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] root trying auth password
2016-12-09 13:13:38+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] login attempt [root/222222] failed
2016-12-09 13:13:39+0000 [-] root failed auth password
2016-12-09 13:13:39+0000 [-] unauthorized login:
2016-12-09 13:13:40+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] root trying auth password
2016-12-09 13:13:40+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] login attempt [root/2222222222] failed
2016-12-09 13:13:41+0000 [-] root failed auth password
2016-12-09 13:13:41+0000 [-] unauthorized login:
2016-12-09 13:13:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] root trying auth password
2016-12-09 13:13:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198738,116.31.116.16] login attempt [root/3.14159] failed
2016-12-09 13:13:42+0000 [-] root failed auth password
2016-12-09 13:13:42+0000 [-] unauthorized login:
2016-12-09 13:13:42+0000 [HoneyPotTransport,198738,116.31.116.16] Got remote error, code 11
2016-12-09 13:13:42+0000 [HoneyPotTransport,198738,116.31.116.16] connection lost
2016-12-09 13:13:49+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:35107 (172.31.22.139:22) [session: 198739]
2016-12-09 13:13:50+0000 [HoneyPotTransport,198739,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:13:50+0000 [HoneyPotTransport,198739,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:13:50+0000 [HoneyPotTransport,198739,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:13:50+0000 [HoneyPotTransport,198739,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:13:50+0000 [HoneyPotTransport,198739,116.31.116.16] NEW KEYS
2016-12-09 13:13:50+0000 [HoneyPotTransport,198739,116.31.116.16] starting service ssh-userauth
2016-12-09 13:13:50+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16]
```

```
root trying auth none
2016-12-09 13:13:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16] root trying auth password
2016-12-09 13:13:51+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16] login attempt [root/3edcxsw21qaz] failed
2016-12-09 13:13:52+0000 [-] root failed auth password
2016-12-09 13:13:52+0000 [-] unauthorized login:
2016-12-09 13:13:52+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16] root trying auth password
2016-12-09 13:13:52+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16] login attempt [root/415263] failed
2016-12-09 13:13:53+0000 [-] root failed auth password
2016-12-09 13:13:53+0000 [-] unauthorized login:
2016-12-09 13:13:53+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16] root trying auth password
2016-12-09 13:13:53+0000 [SSHService ssh-userauth on HoneyPotTransport,198739,116.31.116.16] login attempt [root/454545] failed
2016-12-09 13:13:54+0000 [-] root failed auth password
2016-12-09 13:13:54+0000 [-] unauthorized login:
2016-12-09 13:13:54+0000 [HoneyPotTransport,198739,116.31.116.16] Got remote error, code 11
2016-12-09 13:13:54+0000 [HoneyPotTransport,198739,116.31.116.16] connection lost
2016-12-09 13:14:04+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:55096 (172.31.22.139:22) [session: 198740]
2016-12-09 13:14:04+0000 [HoneyPotTransport,198740,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:14:04+0000 [HoneyPotTransport,198740,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:14:04+0000 [HoneyPotTransport,198740,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:14:04+0000 [HoneyPotTransport,198740,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:14:05+0000 [HoneyPotTransport,198740,116.31.116.16] NEW KEYS
2016-12-09 13:14:06+0000 [HoneyPotTransport,198740,116.31.116.16] starting service ssh-userauth
2016-12-09 13:14:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] root trying auth none
2016-12-09 13:14:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] root trying auth password
2016-12-09 13:14:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] login attempt [root/456852] failed
2016-12-09 13:14:07+0000 [-] root failed auth password
2016-12-09 13:14:07+0000 [-] unauthorized login:
```

```
2016-12-09 13:14:08+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] root trying auth password
2016-12-09 13:14:08+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] login attempt [root/55665566] failed
2016-12-09 13:14:09+0000 [-] root failed auth password
2016-12-09 13:14:09+0000 [-] unauthorized login:
2016-12-09 13:14:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] root trying auth password
2016-12-09 13:14:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198740,116.31.116.16] login attempt [root/5tgb6yhn] failed
2016-12-09 13:14:10+0000 [-] root failed auth password
2016-12-09 13:14:10+0000 [-] unauthorized login:
2016-12-09 13:14:10+0000 [HoneyPotTransport,198740,116.31.116.16] Got remote error, code 11
2016-12-09 13:14:10+0000 [HoneyPotTransport,198740,116.31.116.16] connection lost
2016-12-09 13:14:14+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:19132 (172.31.22.139:22) [session: 198741]
2016-12-09 13:14:15+0000 [HoneyPotTransport,198741,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:14:15+0000 [HoneyPotTransport,198741,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:14:15+0000 [HoneyPotTransport,198741,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:14:15+0000 [HoneyPotTransport,198741,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:14:15+0000 [HoneyPotTransport,198741,116.31.116.16] NEW KEYS
2016-12-09 13:14:15+0000 [HoneyPotTransport,198741,116.31.116.16] starting service ssh-userauth
2016-12-09 13:14:16+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16] root trying auth none
2016-12-09 13:14:16+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16] root trying auth password
2016-12-09 13:14:16+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16] login attempt [root/6625329a! ] failed
2016-12-09 13:14:17+0000 [-] root failed auth password
2016-12-09 13:14:17+0000 [-] unauthorized login:
2016-12-09 13:14:17+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16] root trying auth password
2016-12-09 13:14:17+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16] login attempt [root/6yhnbgt5] failed
2016-12-09 13:14:18+0000 [-] root failed auth password
2016-12-09 13:14:18+0000 [-] unauthorized login:
2016-12-09 13:14:18+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16]
```

```
root trying auth password
2016-12-09 13:14:18+0000 [SSHService ssh-userauth on HoneyPotTransport,198741,116.31.116.16]
login attempt [root/708090] failed
2016-12-09 13:14:19+0000 [-] root failed auth password
2016-12-09 13:14:19+0000 [-] unauthorized login:
2016-12-09 13:14:20+0000 [HoneyPotTransport,198741,116.31.116.16] Got remote error,
code 11
2016-12-09 13:14:20+0000 [HoneyPotTransport,198741,116.31.116.16] connection lost
2016-12-09 13:14:28+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection:
116.31.116.16:39773 (172.31.22.139:22) [session: 198742]
2016-12-09 13:14:28+0000 [HoneyPotTransport,198742,116.31.116.16] Remote SSH version:
SSH-2.0-PUTTY
2016-12-09 13:14:28+0000 [HoneyPotTransport,198742,116.31.116.16] kex alg, key alg:
diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:14:28+0000 [HoneyPotTransport,198742,116.31.116.16] outgoing: aes128-
ctr hmac-sha1 none
2016-12-09 13:14:28+0000 [HoneyPotTransport,198742,116.31.116.16] incoming: aes128-
ctr hmac-sha1 none
2016-12-09 13:14:28+0000 [HoneyPotTransport,198742,116.31.116.16] NEW KEYS
2016-12-09 13:14:28+0000 [HoneyPotTransport,198742,116.31.116.16] starting service
ssh-userauth
2016-12-09 13:14:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
root trying auth none
2016-12-09 13:14:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
root trying auth password
2016-12-09 13:14:29+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
login attempt [root/7418529630] failed
2016-12-09 13:14:30+0000 [-] root failed auth password
2016-12-09 13:14:30+0000 [-] unauthorized login:
2016-12-09 13:14:30+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
root trying auth password
2016-12-09 13:14:30+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
login attempt [root/741963] failed
2016-12-09 13:14:31+0000 [-] root failed auth password
2016-12-09 13:14:31+0000 [-] unauthorized login:
2016-12-09 13:14:31+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
root trying auth password
2016-12-09 13:14:31+0000 [SSHService ssh-userauth on HoneyPotTransport,198742,116.31.116.16]
login attempt [root/7758521] failed
2016-12-09 13:14:32+0000 [-] root failed auth password
2016-12-09 13:14:32+0000 [-] unauthorized login:
2016-12-09 13:14:32+0000 [HoneyPotTransport,198742,116.31.116.16] Got remote error,
code 11
```

```
2016-12-09 13:14:32+0000 [HoneyPotTransport,198742,116.31.116.16] connection lost
2016-12-09 13:14:40+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:58400 (172.31.22.139:22) [session: 198743]
2016-12-09 13:14:40+0000 [HoneyPotTransport,198743,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:14:40+0000 [HoneyPotTransport,198743,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:14:40+0000 [HoneyPotTransport,198743,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:14:40+0000 [HoneyPotTransport,198743,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:14:40+0000 [HoneyPotTransport,198743,116.31.116.16] NEW KEYS
2016-12-09 13:14:41+0000 [HoneyPotTransport,198743,116.31.116.16] starting service ssh-userauth
2016-12-09 13:14:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] root trying auth none
2016-12-09 13:14:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] root trying auth password
2016-12-09 13:14:41+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] login attempt [root/7895123] failed
2016-12-09 13:14:42+0000 [-] root failed auth password
2016-12-09 13:14:42+0000 [-] unauthorized login:
2016-12-09 13:14:42+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] root trying auth password
2016-12-09 13:14:42+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] login attempt [root/7u8i9o] failed
2016-12-09 13:14:43+0000 [-] root failed auth password
2016-12-09 13:14:43+0000 [-] unauthorized login:
2016-12-09 13:14:44+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] root trying auth password
2016-12-09 13:14:44+0000 [SSHService ssh-userauth on HoneyPotTransport,198743,116.31.116.16] login attempt [root/9023] failed
2016-12-09 13:14:45+0000 [-] root failed auth password
2016-12-09 13:14:45+0000 [-] unauthorized login:
2016-12-09 13:14:45+0000 [HoneyPotTransport,198743,116.31.116.16] Got remote error, code 11
2016-12-09 13:14:45+0000 [HoneyPotTransport,198743,116.31.116.16] connection lost
2016-12-09 13:14:53+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:24852 (172.31.22.139:22) [session: 198744]
2016-12-09 13:14:53+0000 [HoneyPotTransport,198744,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:14:53+0000 [HoneyPotTransport,198744,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
```

```
2016-12-09 13:14:53+0000 [HoneyPotTransport,198744,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:14:53+0000 [HoneyPotTransport,198744,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:14:53+0000 [HoneyPotTransport,198744,116.31.116.16] NEW KEYS
2016-12-09 13:14:54+0000 [HoneyPotTransport,198744,116.31.116.16] starting service ssh-userauth
2016-12-09 13:14:54+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] root trying auth none
2016-12-09 13:14:54+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] root trying auth password
2016-12-09 13:14:54+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] login attempt [root/999999999] failed
2016-12-09 13:14:55+0000 [-] root failed auth password
2016-12-09 13:14:55+0000 [-] unauthorized login:
2016-12-09 13:14:55+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] root trying auth password
2016-12-09 13:14:55+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] login attempt [root/a] failed
2016-12-09 13:14:56+0000 [-] root failed auth password
2016-12-09 13:14:56+0000 [-] unauthorized login:
2016-12-09 13:14:56+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] root trying auth password
2016-12-09 13:14:56+0000 [SSHService ssh-userauth on HoneyPotTransport,198744,116.31.116.16] login attempt [root/aaa777] failed
2016-12-09 13:14:57+0000 [-] root failed auth password
2016-12-09 13:14:57+0000 [-] unauthorized login:
2016-12-09 13:14:57+0000 [HoneyPotTransport,198744,116.31.116.16] Got remote error, code 11
2016-12-09 13:14:57+0000 [HoneyPotTransport,198744,116.31.116.16] connection lost
2016-12-09 13:15:05+0000 [kippo.core.honeypot.HoneyPotSSHFactory] New connection: 116.31.116.16:44417 (172.31.22.139:22) [session: 198745]
2016-12-09 13:15:05+0000 [HoneyPotTransport,198745,116.31.116.16] Remote SSH version: SSH-2.0-PUTTY
2016-12-09 13:15:05+0000 [HoneyPotTransport,198745,116.31.116.16] kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2016-12-09 13:15:05+0000 [HoneyPotTransport,198745,116.31.116.16] outgoing: aes128-ctr hmac-sha1 none
2016-12-09 13:15:05+0000 [HoneyPotTransport,198745,116.31.116.16] incoming: aes128-ctr hmac-sha1 none
2016-12-09 13:15:06+0000 [HoneyPotTransport,198745,116.31.116.16] NEW KEYS
2016-12-09 13:15:06+0000 [HoneyPotTransport,198745,116.31.116.16] starting service ssh-userauth
```

```
2016-12-09 13:15:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
root trying auth none
2016-12-09 13:15:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
root trying auth password
2016-12-09 13:15:06+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
login attempt [root/aaaaa] failed
2016-12-09 13:15:07+0000 [-] root failed auth password
2016-12-09 13:15:07+0000 [-] unauthorized login:
2016-12-09 13:15:08+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
root trying auth password
2016-12-09 13:15:08+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
login attempt [root/abcd123] failed
2016-12-09 13:15:09+0000 [-] root failed auth password
2016-12-09 13:15:09+0000 [-] unauthorized login:
2016-12-09 13:15:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
root trying auth password
2016-12-09 13:15:09+0000 [SSHService ssh-userauth on HoneyPotTransport,198745,116.31.116.16]
login attempt [root/ad123456] failed
2016-12-09 13:15:10+0000 [-] root failed auth password
2016-12-09 13:15:10+0000 [-] unauthorized login:
2016-12-09 13:15:10+0000 [HoneyPotTransport,198745,116.31.116.16] Got remote error,
code 11
```