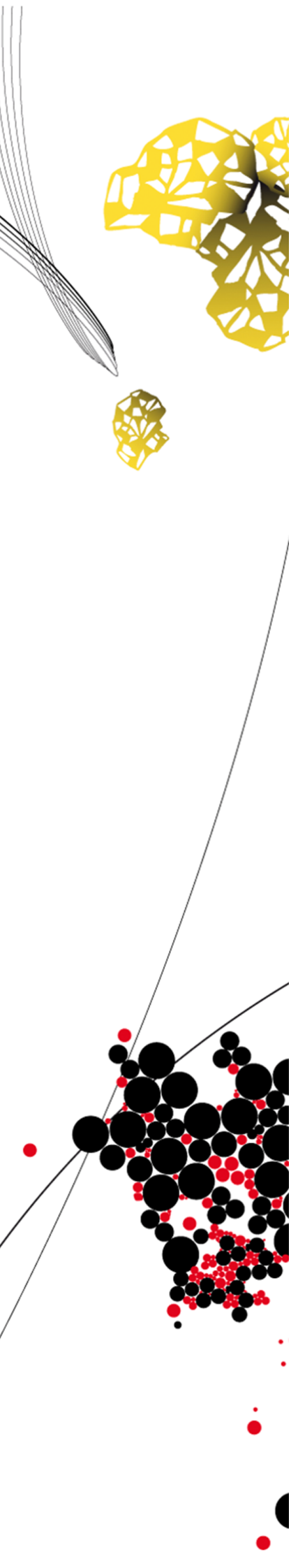

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Detecting Lateral Movement Attacks through SMB using BRO

Ikram Ullah

Master Thesis
November 2016

University Supervisors:
dr. Andreas Peter
Riccardo Bortolameotti
Company Supervisor:
Alex Van t Veer (Fox-IT)

Services Cyber Security Safety
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

This thesis is submitted for the degree of Master of Computer Science at the University of Twente and Technical University of Berlin. The research work was conducted under supervision of Dr. Andreas Peter, Riccardo Bortolameotti in the Department of Electrical Engineering Mathematics and Computer Science University of Twente and Alex Van T Veer from FOX-IT Delft between March 2016 and November 2016.

Acknowledgements

First of all I would like to thank my supervisors Dr. Andreas Peter, Riccardo Bortolameotti and Alex Van T Veer.

I acknowledge Dr. Andreas Peter's incredible guidance both in academic and administrative procedures.

Riccardo Bortolameotti has always endowed me incredible suggestions and support. He was always there when I needed his supervision. His guidance helped me throughout the research work and writing of thesis. I could not have imagined having a better supervisor and mentor for my thesis.

I am grateful to Alex Van T Veer for his support in evaluating my research work and adept feedbacks throughout the research work.

My sincere thanks also goes to manager of MSS deparment at FOX-IT, Christian Prickaerts, for offering me the thesis and internship opportunities in his department and guiding me working on this exciting topic.

Last but not the least, I would like to thank my friends and family for their encouragements and motivation.

Abstract

The purpose of this study is to develop an anomaly based intrusion detection technique to detect lateral movement attack and exerting it in BRO network analyser which is an open source network security platform. Lateral movement attack is one of the phase of Advance Persistent Threat attack during which the attacker progressively move from one system to another in the network, exploit credentials to perform pass the hash attack, escalate privileges, and finally reaching his final targets which are critical systems where key data and assets resides. Lateral movement attack are performed using legitimate computer features and tools. The usage of legitimate features makes it hard to detect it. Although there are many methods of performing lateral movement attack, we have evaluated our detection mechanism against three of the most common lateral movement methods: PSEXEC Windows Management Instrumentation and Pass the hash. One of the consequences of a successful lateral movement attack can be the unauthorized access to personal and financial information of a corporate or organization. This study is an initial attempt to detect lateral movement attack performed through Server Message Block protocol using BRO network analyser. Our proposed detection model is a multi-variant approach as it monitors and detects five different types of user behavioural anomalies in the network. Thus making it harder for any sophisticated lateral movement attack to be perform successfully. We model user behavioural anomalies through a supervised machine learning algorithm. The evaluation results demonstrate that this is a promising model to distinguish legitimate users from an intruder. Our detection model can be easily deployed in any environment and is inexpensive.

Contents

Chapter 1

Introduction

Advance persistent threat (APT) is a set of stealthy and relentless hacking processes, in which a group of highly skilled cybercriminal gains unauthorised access to a targeted network, steals sensitive information and remains undetected while having permanent access to the network. The aim of APT is not to cause damage to the network but to have constant access to the sensitive data. Unlike traditional threats, APTs are fully-customized sophisticated attack against a specific target, planned and designed to thoroughly bypass the security measures of the target. APTs are initiated with stealing legit credentials, phishing, and execution of malware on the victim machine. For victims of APT the implications are huge which includes lost of private data, trade secrets, government secrets, diminished value to brand and reputation, lost of revenue, and cost of technical support. In the recent years large number of corporates became victims of APT resulted in significant losses, including Sony pictures [84], Target [85] and JPMorgan chase [86]. There are six stages of an APT attack: intelligence gathering, point of entry, control command and communication, lateral movement, sensitive data discovery, and exfiltration of sensitive data.

Lateral movement which is also known as east-west movement is a later stage of an APT attack in which the attacker tries to establish and maintain access to the compromised network and increases his footprints by compromising other computers, servers and infrastructure components. During the lateral movement of the APT attack, the attacker wanders further in the inner network to seek more hosts, servers, services and data. The impact of APT attack increases as the attack proceeds to the lateral movement phase because at this stage the attacker remain undetected, gains more access privileges while continuously further exploiting the network. Once the attacker exploits one machine in the network, he can use different techniques and tools to remain persistent in the network and move laterally to further compromise the network. The attacker avoids detections by using legitimate administrative tools. For incident and response team (or system administrator) detecting lateral

movement attack at the early stages is very important before the whole network is compromised and the attacker successfully removes the traces. In 2014 1,000 retail businesses were hit by remote attacks[18]. Ultimately, most retail attacks started with stolen credentials, which enabled attackers to move laterally, harvesting credentials along the way until they reached their final destination. Single factor authentication (username and password) makes attacker job easy because once he steals the administrator's username and password then he uses legitimate tools: PSEXEC[99], WMI[101] and techniques (Pass the Hash) to remotely access the system and install persistent backdoors. With valid administrator credentials the attacker uses dumping tools: PROCDUMP[16] to dump the usernames and hashes and access other machines on the network. The majority of system administrators implement Credentials reuse. Credential reuse means using the same username and password for multiple accounts or machines. Credential reuse makes the system administrators job easier but with credential reuse the system administrators are not only expediting his own administrative tasks but also making attacker task easier. So, for the attacker to compromise one machine means compromise almost the whole network especially in an environment with all the machines having same local system administrator account. Detecting lateral spread is very difficult because the attacker subterfuge and behaves as authorised users. Below are listed the methods used by attackers to move laterally in the network:

(a) **Credential Harvesting.** The term credential harvesting is also called account harvesting which refers to the attacking technique or activities of grabbing legitimate user ID and passwords to gain access to target systems for illegal or malicious purposes[115]. In large environments attacker wants to compromise all the hosts in the network, to do so he needs to steal the credentials of all the hosts in the network. The attacker can steal or exploit credentials (passwords, hashes) using social engineering techniques, use defaults credentials, perform phishing attack or dumping Local Security Authority Subsystem Service of the victim machine as described in Section 4.0.3. Hashes are used to perform pass the hash attack which is also described in Section 4.0.3.

(b) **Persistency** Persistence is any access, action, or configuration change to a system that gives an adversary a persistent presence on that system. Adversaries will often need to maintain access to systems through interruptions such as system restarts, loss of credentials, or other failures that would require a remote access tool to restart or alternate backdoor for them to regain access[116]. For the attacker it is important to gain persistent access to the network. Persistency techniques are: installing backdoor, creating new accounts in case credentials are changed, and changes the network configuration to keep his existence hidden.

(c) **Internal reconnaissance.** After one host in the network is compromised, the attacker needs to perform internal reconnaissance to better understand the network architecture, the type of

firewall installed, and other network information. This knowledge is important because it helps the attacker to avoid detection. Attacker can use Netstat[117] tool for internal reconnaissance.

(d) **Escalate privileges.** Most of the time the attacker motive is to go much deeper into the network by escalating privileges based on the knowledge the attacker acquired during reconnaissance. Tools like PSEXEC[99], and WMI [101], allows the attacker to access, scheduling tasks and execute malicious code on other machines on the network using the knowledge and credentials he obtained in the previous three steps (Credential harvesting, remain persistence, internal reconnaissance).

1.1 Overview of existing solutions

The number and complexity of cyber-attacks are ceaselessly increasing while the existing security measures is far lagging behind in detecting sophisticated attacks such as lateral movement. Jasek et al.[93] proposed honeypots to detect lateral movement which is unnecessarily expensive. Honeypots are expensive to maintain because if virtual machines are not available, one physical computer must be maintained for each honeypot, which can be exorbitantly expensive[118]. Example: Honeynet. Balduzzi et al.[91] suggests the identification of lateral movement attack by detecting malicious URLs that spreads malware and backdoors. The attacker can bypass this approach by changing URL to spread malware. Bencsth et al.[92] proposed detection approach that detect one specific malware that is frequently used in APT attack including lateral movement, however, having a detection mechanism so tailored for a specific malware it makes the entire detection unable to detect new malware that behave differently. Bass[96] detection approach implements different types of anomaly sensors which include sensors to monitor user activities, sensors to monitor host activities, and sensors to monitors application activities. These sensors are trained for a subsequent amount of time. Any new activity in the network is considered anomalous. The shortcoming of this approach includes complexity, costly, and host behaviour swiftly changes.

Siadati et al.[95] introduced a detection approach based on the observation of attackers usage of stolen credentials by analysing the time of logins, frequencies of login, and machine used to login. The attacker can bypass this approach if the attacker login during the times of normal user and less frequently. Oberle et al.[94] suggested two factors-authentication to detect pass the hash attack. Two factor-authentication may help in reducing the risk of lateral movement by introducing a new security check. However, this requires changes to the infrastructure, such as introducing security tokens. The goal of this work is to deploy a security solution that detects lateral movement and adapts to the current infrastructure without additional changes. Most of the existing approaches that try to detect

lateral movement cannot effectively detect a general scenario of lateral movement without relevant changes to the underlying infrastructure.

1.2 Research Question

Although there are many methods through which attacker can perform lateral movement, the problem being addressed in this study is attacker performing lateral movement attack through Server Message Block protocol. Server Message Block (SMB) protocol is installed in all Windows Operating Systems. SMB protocol is used to share files and printers on a network. Numerous vulnerabilities exist in SMB protocol. The attacker can exploit these vulnerabilities and perform lateral movement attack against the windows machines and compromise the whole network. Exploitations of SMB vulnerabilities have immense impact because of its high percentage usage in critical infrastructure such as banking and finance, electricity grids, law enforcement agencies, aviation and medical centres. Also in the real world most of the communication protocols are interlinked. Vulnerability in one protocol can lead to exploitation of other protocols or machines. For example vulnerability in authentication protocols (like NLTM, Kerberos) can lead to exploitation of SMB protocol[119]. New Vulnerabilities are emerging everyday, it is hard and expansive to patch all the vulnerabilities. In some cases vulnerabilities are exposed when it has already made considerable amount of damages. Being vigilant with proper detection technology can limit the adverse impact of these attacks. In this work we want to develop a real-time monitoring approach for the detection of lateral movement attack. The intrusion detection approach we are investigating is anomaly based. Anomaly based intrusion detection systems are a better alternative for signature-based intrusion detection systems, which detects both known and unknown attacks. The main challenge in developing anomaly based intrusion detection system is to correctly differentiate between anomalous traffic and normal traffic, because if normal activities are flagged as intrusion it can generate false alarms which can undermine the network performance and also an additional unnecessary work for the technical team. Considering teh lack of comprehensive lateral movement detection techniques, with this work we want to answer the following main research question: How can we detect lateral movement attacks over SMB using anomaly based approach?

1.3 Our proposed approach

We propose a novel framework that combines different techniques based on data analytics. Our proposed model analyses multiple vectors of anomalies. While analysing malicious lateral movement

By SMB, we mean both SMB1 and SMB2 versions

traffic, we find out five crucial anomalies related to lateral movement attack and write five policies to detect these anomalies. A brief description of each anomaly is discussed below:

(1) Instead of following standard NTLM authentication procedure which requires correct password as a mean of authentication, the attacker bypass the normal authentication procedure by providing hashes, which generates empty session key. Such contravention of authentication procedure can be detected via empty session key. A policy is written that detects empty session key.

(2) The attacker might try to connect to multiple hosts at the same time. Such an aggressive malicious behaviour is detected by most of the commercial Intrusion detection systems and firewalls which includes policies that controls the number of connections or session initiated from a specific host. Attacker invades firewall and IDS, by generating random hostname for each connection. These random hostnames have much high randomness entropy. A policy is written that calculates the randomness entropy of each hostname, and checks if its above a certain scale.

(3) To escalate the privileges the attacker might perform brute force attack by trying many passwords hoping to eventually guess the passwords correctly. K-nearest neighbour algorithm is trained with the number of failed login attempts during the normal network behaviour and malicious network behaviour. Based on the training, the machine learning algorithm provides a threshold that can be considered malicious. The threshold is used in policy that monitors failed login attempts over a specified time interval.

(4) Our models monitors the flow of data along IPC$ share. K-nearest neighbour machine learning algorithm is trained with the flow of data towards IPC$ in normal network behaviour and malicious network behaviour. Based on the training, the machine learning algorithm provides a threshold that can be considered malicious. The threshold is used in policy that monitors access to IPC$ over a specified time interval.

(5) Thereupon the upload of malicious executables to IPC$, the attacker install these executables as services. K-nearest neighbour machine learning algorithm is trained with number of services created or started in normal network behaviour and malicious network behaviour. Based on the training, the machine learning algorithm provides a threshold that can be considered malicious. The threshold is used in policy that monitors start and creation of services over a specified time interval.

Our model provides a real time comprehensive data security analytics that precisely identifies invasive behavioural patterns. Along with detection of malicious lateral movement, the results of evaluation shows that the model also detects misconfiguration of the internal network, violation of access control policies by legit users. The real time reporting characteristic of our model gives the

network administrator an early glimpse of intrusion before it gets successful and persistent. The characteristics that makes our approach superior are easy to develop and deploy, inexpensive with better detection capabilities, and very limited human interaction needed. Moreover this model is equally advantageous for small and large network environments. The limitation of our approach is that some features requires a large datasets in order to be precise. This is because they are based on the K-nearest neighbour algorithm. However, in a company setting this is not a big problem, because this large dataset can be captured in one or few days. It is worth to mention that our approach works also with limited datasets, although it may raise a higher number of false positives and false negatives.

1.4 Report organization

The remainder of this report is organized as follows. Chapter 2, describes related work. Chapter 3 provides the reader with background knowledge about Server Message block protocol, BRO network analyser and K-nearest neighbour algorithm. Chapter 4 lateral movement attack techniques, which tools can be used to perform lateral movement attack and how these attacks are performed. Chapter 5 discusses our detection model and the evaluation of our model. Finally, in Chapter 6, conclusions and recommendations are given.

Chapter 2

Anomaly based Intrusion Detection Techniques

An attempt to compromise the security of a host or a network is called an intrusion. An intrusion can be disrupting the system or service performance, illegally accessing sensitive information, system modification by installing persistent backdoors, lateral movement etc. The objective of intrusion detection system is to detect intrusions that exploit the Confidentiality, Integrity and Availability of a host or network. Large resources are concentrated on the development of intrusion detection systems

to detect any internal or external intrusion. Considering the continuous expansion of Internet and the continuous spurt of new attacks, it is getting harder and harder to provide adequate detection mechanism against the vast number of networks based attacks. Many techniques have been suggested to detect the large number of attacks. There are many challenges in developing a full-fledged anomaly based intrusion detection system because the network traffic is so dynamic and complex that it is hard to differentiate between normal and abnormal traffic. Also there are risks of false intrusion alarms. Understanding and predicting all the abnormal traffic patterns remain a long time challenge. Accuracy is the essential requirement of an intrusion-detection system, its extensibility and adaptability are also critical in todays network computing environment [50]. Intrusion detection system monitors host or network traffic for malicious activities or policy violations.

2.1 Host based intrusion detection systems

Intrusion detection system installed in the host is known as Host based intrusion detection system (HIDS). HIDS are among the first intrusion detection systems built [58]. Simple statistical methods can be used to model normal user or program behavior. Deviation from the normal behavior is flagged as anomaly. Examples of detecting anomalies at host level are, detecting anomalous program behavior [44] and detecting intrusions using system calls [45].

2.2 Network based intrusion detection systems

Intrusion detection system installed for the security of network is known as network based intrusion detection system (NIDS). In case of NIDS, depending on the type of information thats is used for performing the detection; one can distinguish between traffic and application models [58]. NIDS that use traffic models monitors inbound and outbound traffic to all devices in the network. While the application models use application specific knowledge. Application models are used to detect attacks against specific applications or services.

2.3 Intrusion detection types

2.3.1 Anomaly based intrusion detection system

Anomaly based intrusion detection system detects any activity that deviates from normal traffic or user behaviour. Three broad categories of anomaly detection techniques exist [62]. Supervised anomaly detection, semi-supervised anomaly detection, and unsupervised anomaly detection. Supervised anomaly detection methods are trained with normal and malicious datasets of traffic.

Semi-supervised anomaly detection methods are trained with normal traffic datasets. Unsupervised anomaly detection do not requires any training. An anomaly detection approach usually consists of two phases: a training phase and a testing phase. During the training phase, the normal traffic profile is defined; while in the testing phase, the learned profile is applied to new data [31].

Anomaly-based NIDS can be classified according to:

(i) The underlying algorithm used.

(ii) Whether to analyze the features of each packet singularly; Packet oriented or of the whole connection; Connection oriented.

(iii) Type of data analyzed. In particular, whether to focus on the packet header or on the payload [59].

(a) **Packet oriented NIDS**

In packet oriented NIDS each packet is analyzed and anomalous attacks are detected by inspecting packet payload, packet header or the combination of both. Header information is mainly useful to recognize attacks aiming at vulnerabilities of the network stack implementation or probing the operating system to identify active network services. On the other hand, payload information is mostly useful to identify attacks against vulnerable applications [42].

(b) **Connection oriented NIDS**

Connection oriented NIDS considers features of the whole communication before establishing whether it is anomalous or not. Theoretically, a connection-oriented system could use as input the content (payload) of a whole communication (allowing at least in principle a more precise analysis), but this would require a long computational time, which would seriously limit the throughput of the system. In practice, connection- oriented systems typically take into account the number of sent/received bytes, the duration of the connection and layer 4 protocol used [59].

2.3.2 Signature based intrusion detection system

Signature based intrusion detection system detects malicious activities using patterns of known attacks. Signature based IDS are of two kinds, Host based IDS that monitors network traffic and Network based IDS that monitors a host processes. In signature based IDS intrusion patterns of malicious traffic or processes are formed, if these patterns are detected in the traffic or processes, alarm is flagged. Such as to detect virus in an email, a signature is written that detects attachments with a

particular name of the virus and to detect DOS attack, signature is written that detects the number of times a command is issued. Also signatures can be written to detect keywords in the traffic. Below are two examples of signature based intrusion detection system [60][61].

```
if (traffic contains x90+de[^\r \n]30) then attack detected
```

Snort implements Signature based intrusion detection. Below is an example of Snort signature

```
alert tcp any any - > any \

(flow: establish, to_server; \

content: foo; msg: detected foo;)
```

2.3.3 Rule based intrusion detection system

In rule based intrusion detection system, rules are formed from normal traffic patterns or rules can also be formed to represent malicious behavior. These rules are saved in the database. During the detection phases these rules are applied to the network traffic to identify if the traffic is normal or malicious. For instance rules can be, remote users can login with username and password not with username and hashes or no user is allowed to remotely install applications in other users computers.

2.3.4 Advantages and Disadvantages of Anomaly and Signature based IDS

Both such systems have advantages and disadvantages. A brief comparison between the anomaly based intrusion system and signature based intrusion detection system is described here. The advantages of signature based intrusion detection system are that such systems are quite reliable in detecting known attacks, easy to implement, and suffer by less number of false positives. The disadvantages of signature based intrusion detection system require updated databases of known attacks patterns, attacks that are unknown to the IDS system cannot be detected and, creating signatures is an expensive manual process, easily by-passable by attackers with small modifications. Advantages of Anomaly based IDS are that, in contrast to signature based IDS anomaly based IDS can detect new attacks without the need of having an updated database, can detect wide number of intrusions. Disadvantages of anomaly based IDS are, large amount of training data is required to accurately form normal profiles of users and services and to train the IDS for precise detection, if not properly configured any attack traffic that simulates normal traffic cannot be detected, compare to Signature based IDS anomaly based IDS are more vulnerable to false positive because they use statistical technique with no prior knowledge of the attack in contrast to Signature based IDS which has signatures of all know attacks. Attacks against the present information systems networks are

increasing, largely new attacks are conducted, so in such circumstances signature based IDS is not an appropriate approach to detect the large number of new attacks, in contrast anomaly based IDS can cope well with the emerging new attacks.

2.3.5 Anomaly based intrusion detection techniques

Anomaly based intrusion detection comprise of three techniques Data mining anomaly detection, machine learning anomaly detection and statistical anomaly detection [31].

(a) **Data mining and machine learning technique**

Data mining anomaly detection generally refers to the process of extracting useful rules from large stores of data. The recent rapid development in data mining contributes to developing wide variety of algorithms suitable for network intrusion detection problems. In Data mining based anomaly detection technique, the audit data is classified as normal or anomalous based on set of rules and patterns, or associating events together. The advantage of this technique is that it is simple to define rules but the disadvantage is that it is hard to represent different types of information. Machine Learning is an artificial intelligence technique that stores the user-input stream of commands in a vectorial form and is used as a reference of normal user behavior profile. These profiles are then grouped in a library of user commands having certain common characteristics [55]. In Machine learning based anomaly detection technique, the system learns the normal behavior of a program, improves its execution strategy by acquiring new information about the program, thus when the program deviates from the normal behavior, the system triggers an alarm.Although there are large number of challenges linked with machine learning algorithms, they do perform very well in most networks and the advantages of machine learning based anomaly detection are that it can avoid manual configuration and customization for different industrial control system networks.

Masud et al. [40] developed Data Mining based Exploit code detector (DExtor) technique. Their system differentiates between normal traffic and remote exploitation code. DExtor technique consists of two phases. One is disassembling the traffic and the second phase is features extraction from the traffic. During the disassembly process, illegal and important instruction sequences are distilled from the sequence byte. In feature extraction phase, the occurrence of useful instructions is identified and the frequency of each instruction is measured. Based on the set of instructions and their frequency distribution it is decided if the traffic is normal or an attack. From the disassembled data features like Useful instruction Count (UIC), Instruction Usage Frequencies (IUF) and Code Data Length (CDL) are extracted and classifiers such as Support vector

machine (SVM), Bayes net, decision tree (J48), and boosted J48 are applied. The implementation of DExtor in large networks is unrealistic because it is efficient only for 42KB/sec network traffic.

Harinee et al. [50] proposed intrusion detection system implements fuzzy class association rule mining method that is based on Genetic Network Programming (GNP). Association-rule mining is used to discover association rules among different attributes in network traffic dataset. The proposed technique can be applied to Network related databases that contains network architecture attributes. Fuzzy rules can be defined based on the type of service or user normal activities. The limitation of this work is that it is designed for small ad hoc networks and its capability is not shown statistically like which attacks were detected with this technique and what is its efficiency for large networks.

Zhang et al. [54] implemented data mining algorithm called Random forests algorithm to achieve unsupervised network based anomalies detection. A random forests algorithm build patterns of network services (i.e. http, ftp, telnet) from the traffic data. Any traffic that is not consistent with the normal traffic patterns is considered as anomalous. The limitation of this approach is that processing is done offline using Wireshark so such a system is not applicable for real time anomalies detection.

Gu et al. [30] develop network anomaly detection technique thats based on maximum entropy and relative entropy techniques. The normal traffic packets are divided into two dimensional packet classes according to packets protocol information and destination port numbers with total of 2348 packet classes. This approach comprises two phases. In phase one Maximum entropy baseline distribution of the packet classes in the training data is calculated through maximum entropy estimation. The training data is labelled and packets labelled as anomalous are removed. In the second phase the observed network traffic trace is given as input and relative entropy is used to compare the empirical distribution of observed packet classes to baseline distribution. Variation in the two distributions determines that the packet classes are different from the training dataset and indicates an anomaly. The limitation of this approach is that the labelling of normal traffic is done offline and manually, so for large networks and datasets this technique is time consuming and error prone.

Mantere et al. [46] describes the network features that can be used in machine learning based

anomaly detection to detect IP traffic anomalies in specific Industrial Control System (ICS) Network. Their technique is based on forming separate models for different functional level such as device levels, cell levels, and the plant levels. The network features that are considered for anomaly detection are;

(i) Throughput

(ii) Any large variation in the throughput of ICS network can be considered as anomaly

(iii) IP address-Port Pairs, new IP-Port pairs is clear indication of new (malicious) service

(iv) Average packet size, average packet size can be an identification of normal traffic

(v) Timing; packet timing and interval between packets from a network node can be used to identify the traffic type,

(vi) Flow Direction and duration; variations in ICS traffic direction or duration can be an identification of anomaly e Payload data, payload data can be used to detect malicious actions

(vii) Network Protocol, to static networks sign of new protocol or changes in protocol settings is identification of anomaly

(viii) Connectivity number; in ICS networks the number of connections for each system is static.

Tools like Wireshark[106], BRO[109] are used to extract these features from the traffic. The limitation of this approach is that it is literally designed for ICS network. The ICS network traffic is quite uniform and predictable compare to general ICT network traffic, which are unpredictable. Also this approach is based on usage of combination of many tools and suits offline analysis not real time.

Fiore et al. [49] network anomalies detection approach is based on machine learning using Restricted Boltzmann machine (RBM). RBM is an artificial neural network that can learn probability distribution from its input. Such model is first trained on normal traffic dataset and then used to classify previously unseen or suspicious events. The tool used in this model is called Discriminative Restrictive Boltzmann. The performance of this approach is not impressive because it suffers from many classifications errors because of the vague notion of normal traffic.

Mahoney et al [41] developed an IDS system known as Packet header anomaly detection (PHAD) to detect attacks that are caused by anomalous packets based on examining packet header fields. In PHAD alarms are ranked based on how unusual and an unexpected an event is so events that occur with the probability p receive a score of 1/p. So if a packet field is observed n times with r distinct values, then there must be r anomalies and the probability that the next observation to be anomalous is approximated by r/n. Because of the dynamic behavior of real time traffic, PHAD

uses non-stationary model so if an event last occurred t seconds ago, then the probability it occurs in the next one second is approximated by 1/t. In PHAD each packet header field containing an unusual and unexpected anomalous value is assigned a score inversely proportional to the probability

$$Score_{field} = tn/r$$

To score the packet, as the fields of the packet are not independent because the fields occur sequentially, for k consecutive anomalies the packet score is

$$Score_{packet} = \sum t_i n_i / r_i$$

Where i $\in$ anomalous fields, the fields not found in the testing or training data.

PHAD detected attacks (i.e. DOS, SMTP buffer overflow, R2L, DNS buffer overflow, SYN flood, ping of death, backdoor install, NT bug exploit, UTR, probe) via anomalies like, weird TTL field value, bad TCP checksum, fragmented IP, FIN without ACK, IP source address, outgoing packet size. Many of the attacks like ffbconfig, httptunnel, ntfsdos, sqlattack, sshtrojan, tcpreset were not detect mostly because in these attacks vulnerabilities at the application layer are exploited which is not monitor by PHAD.

Ye et al. [51] presents an anomaly detection system that is based on Chi-square statistic. In this method profile of normal behavior of traffic in a network is formed and the deviation from the normal behavior is considered as anomaly. This approach uses modified version of Hotelling T2 statistics; which is a multivariate statistics that can anticipate multiple subjects, multiple actions and multiple behaviors to compare the patterns of normal and anomalous traffic. This approach establishes a normal behavior profile thats compared with the observed behaviour. Modified version of Hotelling T2 statistics

$$X^2 = \sum_{i=1}^{n} (X_i - E_i)^2 / E_i$$

Where Xi denotes the observed value of the ith variable, n is the size of data sample, Ei is the expected value of the ith variable and n is the number of variables. To estimate the expectation $\bar{X_1}$, $\bar{X_2}$, ... $\bar{X_n}$ where is sample mean vector of expected value. The previous equation becomes

$$X^2 = \sum_{i=1}^{n} (X_i - \bar{X_i})^2 / \bar{X_i}$$

X^2 is the sum of squared differences between the observed and expected values of the variable. If the value of X^2 is small it means the observed traffic is similar to the expected traffic. The anomaly detection rate of this approach is tested on a small dataset. For large set of dataset this technique lacks performance and scalability because of the fact that its is hard to fully define normal traffic for a better estimation.

Bolzoni et al. [52] presents POSEIDON anomaly-based network intrusion detection system. It is a combination of two different techniques, Self-Organization Map (SOM) and modified PAYL [42]. SOM has the capability of high quality classification of payload data so it is used to form profiles, while n-gram algorithm used in PAYL has the knack of better detection of anomaly with respect to the normal behaviour. In this approach packets are classified in an unsupervised manner using neural networks (SOM). As POSEIDON is packet oriented so during the training phase for given dataset, a set of Models Mnjk are computed for each packet where j is the destination address, k is destination port and n is the classification derived from the neural network which is computed during SOM phase. Model Mnjk stores the average byte frequency and the standard deviation of each byte frequency. During the detection phase the value of each incoming packet is compared to the model value. Large deviation from the norm value causes an alarm. Compare to PAYL [42] and PHAD; POSEIDON has higher detection rate and lower number of false positives. Thus if an attacker insert extra bytes in the payload, which results in different classification and an alarm is flagged. But in this approach the implementation of SOM is not optimal because if the attacker inserts small malicious payload that has the same byte frequency distribution as of normal traffic and the attack cant be detected. The real challenge in detecting attacks is when the traffic doesnt look significantly malicious which is the case in application layer attacks. 75% of the successful attacks happened on application layer and 80% of enterprises become victim of application layer attacks [57].

Mahoney et al. [56] suggests a learning algorithm that generates models of normal behaviour from normal network traffic. Traffic behaviour that deviates from the learned normal models indicates possible network based anomaly. This approach is the combination of two existing techniques PHAD [41] and ALAD [56]. PHAD models normal behaviour of data link, network and transport layer traffic and ALAD, models application layer behaviour. Different models are formed based on the features of PHAD and ALAD. In PHAD the features are the fields of the packet header, in ALAD the features are application protocol keywords, opening and closing TCP flags, source address, destination address and port number. The limitation of this approach is that most attacks couldnt be detected because these attacks features are neither defined in PHAD or ALAD.

(b) **Statistical anomaly detection technique**

The goal of statistical Anomaly detection is to identify some traffic parameters, which can be used to describe the network traffic and that vary significantly from the normal behaviour to the anomalous. For example packet length, inter-arrival time, flow size and the number of packets per flow [29]. In statistical anomaly detection method, a profile of the user is formed on the basis of his behaviour (audit logs, incoming traffic etc.) and this profile is stored. Anomalies are flagged when the user current behaviour varies from the stored profile. The advantage of this technique is that it is a good indicator of detecting malicious activities. An obvious disadvantage is that an attacker can deceive the detection system to consider malicious traffic as normal traffic.

Abouzakhar et al. [53] developed statistical detection model, which uses Chi-Square statistics to detect Network based intrusions. This approach comprise of three steps; First step is Network traffic categorizer & data pre-processing in which the TCP flags are extracted of each input packet, frequency distribution is generated, split into four categories as number of RST, SYN, ACK, ICMP packets per second and average number of packets per second is calculated for each category. Secondly, Chi-square Goodness-of-fit test is used to detect anomaly by comparing the Chi-squared values of the expected (normal) traffic and the observed traffic.

$$X^2 = \sum_{i=0}^{k} (O_i - E_i)^2 / E_i$$

Where O_i is the observed frequency for a category, E_i is the expected frequency for a category and k is the number of observations in the sample. Thirdly, decision phase an alarm is raised if the Chi-square value of the observed traffic is higher than the Chi-Square value of the normal traffic. Such an approach works efficiently for small and static network in which the number of protocols used are known and fixed. For large and dynamic networks with varying number and types of protocols this approach lacks efficiency and performance.

Wang et al [42] developed payload-based anomaly intrusion detection system called PAYL. It automatically and efficiently models application payload of Network traffic. It is site or service specific intrusion detection system. PAYL consists of two phases. First is learning phase where it computes profile of application payload to a host and port. The system first learns a model or profile of the expected payload delivered to a service during normal operation of a system. Each payload is analyzed to produce a byte frequency distribution of those payloads, which serves as a model for normal payloads. The second phase is anomaly detection phase; Mahalanobis Distance is used to calculate the similarities between the new data and the pre-computed profile

or model. In this phase two distributions are compared. Simplified Mahalanobis Distance is used to compare the two statistical distributions, the model and the received payload. So any payload that is found to be too distant from the normal expected payload is considered as anomalous and alert is generated. While within each port the length of the payload differs for example TCP packet length ranges from 0 to 1460; so different length of payloads have different type of payload. Thus in PAYL they compute a payload model for each different length range for each port and service and for each direction (inbound, outbound) of payload flow. To compute this payload model they used N-GRAM analysis and in particular byte value distribution (n=1). In a payload, N-GRAM is a sequence of adjacent bytes. A Model Mij (i is the payload length for a port j) stores the average byte frequency and the standard deviation of each bytes frequency. So a payload is characterized by combination of means and variance of each bytes frequency. For example if there are 10 ports and each ports payload has 10 different lengths then there can be 100 models. During the detection, each incoming payloads byte value distribution is computed and then compared with if the distribution is different the detector flags the packet as anomalous and generates an alarm. The distance between the byte distribution of the received payload and the profile from the model computed for the corresponding length range is measured. If the distance is higher it is likely that the payload is anomalous. The simplified Mahalanobis distance is given below

$$d(x,\bar{y}) = \sum_{i=0}^{n-1} (|x_i - \bar{y}_i| / (\bar{\sigma}_i + \sigma))$$

Where $\bar{\sigma}$ is the standard deviation, σ is the smoothing factor which represents the statistical confidence of the sampled training data, n is 256 possible byte values, x is the new feature value and $\bar{y}$ is the average feature value computed from the training data. Although with PAYL attacks using TCP can be detected, while attacks that use UDP, ICMP and ARP cannot be detected. Also attacks that doesnt include payload cannot be detected so remote code execution attacks cannot be detected with this technique.

Krugel et al. [58] approach is based on utilizing the application level knowledge to protect the network services from intrusion. Such an application level model allows the detection of malicious contents hidden in single network packets. The idea behind service specific anomaly detection is to include application payload information along with packet header information. So network traffic is partitioned and independently analyzes packets sent by different applications. Concentrating on one type of service traffic allows the collection of statistical data with less variance and thus allowing precisely establishing notion of normal traffic for each service. This approach is trained for a specific training period during which it reads packets from the network, this data is split into service specific traffic and normal profile of each service is formed. During the detection, observed traffic is compared with normal profiles.

Chapter 3

Preliminary Knowledge

3.1 Server Message Block Protocol

Server Message Block protocol (SMB) is a client-server and request-response protocol in a computer network. SMB protocol can be used on the top of protocols like TCP/IP, IPX/SPX or other network protocols. It is installed in almost all Microsoft Windows machines. Through SMB protocol clients can access files that are present on the server. Based on the file access control, the client can create, read and update the files on the server. Including file system support, SMB protocol also specializes in Inter process communication (IPC). IPC share is useful because it facilitates data exchange between computers over SMB protocol. SMB protocol is evolving and continuously updating protocol. It evolved from CIFS to SMBv1 to SMBv2 to SMBv3. SMB protocol is remote sharing/file protocol for accessing files and printers across the LAN and WAN. Many Operating systems vendors like Apple, EMC, Microsoft, and Linux have implemented SMB protocol. With the passage of time Windows has made enormous improvements to the protocols such as adding Kerberos authentication, Signing using HMAC MD5 and many other improvements. SMBv2 is the first upgraded version of SMB. Compare to SMB, SMBv2 has increased file-sharing scalability, security, number of round trips of request is reduced, asynchronous operations, and the larger reads and writes (more data in each packet). Security related improvements are the signing uses HMAC SHA-256, and also the size of command set is reduced from 75 to 19 [2]. In older versions of Windows (e.g 95,98, ME & NT), SMB shares ran on NETBIOS over TCP/IP (NBT) on ports 137/tcp and UDP, 138/UDP, and 139/tcp. However, in later version of Windows (e.g 2000 and XP), it is possible to run SMB directly over TCP/IP on port 445/tcp [87]. For further information see [4].

SMB protocol provides two level of security that is user level and share level. Share is a file or printer that can be accessed by client. In user level authentication the client provides username and pass-

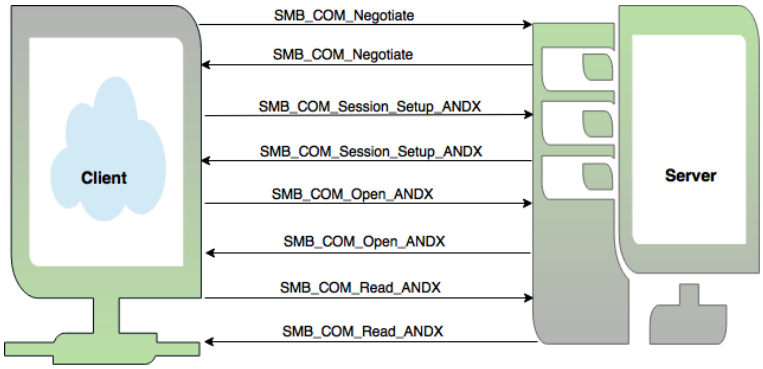

Figure 3.1.1: Microsoft SMB Protocol Packet Exchange Scenario

word to access a share. In share level authentication each share is protected individually. So the client has to provide password for each share. The password is encrypted in both the cases. Some of SMB supported authentication protocols are WindowsNT Challenge/Response NTLM, NTLMv2, KERBEROS, W2K and NT Domain Authentication. NTLMv2 authentication is based on challenge response, which contains nonce from a client and nonce from server.

Figure 3.1.1 shows how SMB client and server initiates communication. Client sends SMB_C0M_Negotiate to server to request negotiation of SMB protocol dialect. In this message the client includes his dialects (max buffer size, canonical file names, etc). In response to the client request the Server identify SMB protocol dialects for the session and also includes 8-byte random string used in the next to authenticate client. Client then sends SMB_COM_Session_Setup _ANDX message to identify his capabilities. The client also sends username, domainname, or password hash; the server supports both plaintext password as well as password hash. In response to this message, if the server accepts challenge/response, Server will issue a valid UID to the client for the session else the server will deny the client access request. The issued UID is submitted with all subsequent SMBs on that connection to the server. If the client is granted access, the client sends SMB_COM_Tree_Connect_ANDX message to request access to the share (e.g. IPC$, Admin$, C$) and fully specifying the path of the share. Based on the client credentials if the client is allowed to access the requested share, the server returns 16-bit tree ID (TID) else the server responds with error message and deny access to the request share. With SMB_COM_Open_ANDX message the client request the server to open a file on the accessed share. If access to the requested file is granted, in response the server returns file ID of the requested file. In SMB_COM_Read_ANDX message the client includes the file ID issued to the client in the previous message to request the server to read data from the previously opened file and return its data to the client. In response to this request the server return the requests file data.

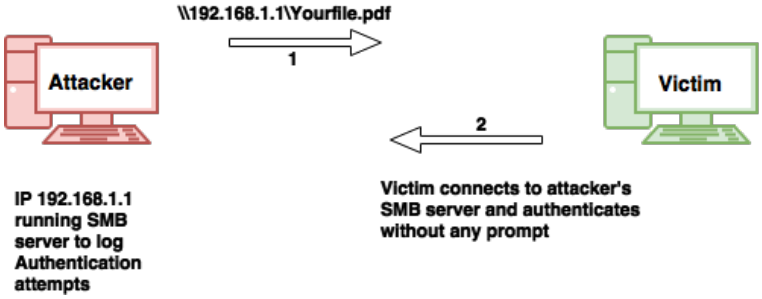

Figure 3.1.1.1: Phishing style attack abusing SMBs automatic authentication

This research work is specifically about three major attacks through SMB that are running a process remotely using PsExec[99], running a service through Windows management instrumentation and Pass the hash. How this goal can be achieved is discussed in the later section of this paper. SMB protocol is an attractive protocol for attackers because of the reasons describe below:

(a) SMB is installed in all Windows Operating Systems and it is a trusted file sharing protocol.

(b) Attackers can exploit SMB vulnerabilities with legitimate tools, which helps attacker to avoid detection. Tools like PsExec[99], and WMI[101].

(c) Attacker can exploit SMB vulnerabilities to perform lateral movement attack.

3.1.1 SMB protocol Vulnerabilities

Below are described some of the well-known SMB protocol vulnerabilities that can be exploited by an attacker to perform lateral movement attack:

(i) SMB-RELAY, which is a specific type of man in the middle attack perform to steals users SMB credentials.

(ii) Phishing style attack abusing SMB protocol automatic authentication as shown in Figure 3.1.1.1, is another way for the attacker to get SMB protocol credentials of a victim is to trick a victim to click on a link, which causes the browser to authenticate with a remote SMB server but the attacker controls this server. The link might look like $192.168.1.1\backslash Yourfile.pdf$. While 192.168.1.1 can be the IP address of the untrusted SMB server controlled by the attacker.

Thus the attacker gets access to SMB credentials. The potential down fall of stolen credentials can be access to network shares, executing of code with applications such as PsExec, RDP access, Windows live account access and much more [23].

(iii) The vulnerability is an index error in the SMB2 protocol implementation in srv2.sys, which allows remote attackers to either cause a denial of service attack or execute remote code on a vulnerable system through an ampersand (&) character in a Process ID High header field in a NEGOTIATE PROTOCOL REQUEST packet. This triggers an attempted dereference of an out-of-bounds memory location [19].

(iv) CVE-2009-2532: Microsoft Windows Vista Gold, SP1, and SP2, Windows Server 2008 Gold and SP2, and Windows 7 RC do not properly process the command value in an SMB Multi-Protocol Negotiate Request packet, which allows remote attackers to execute arbitrary code via a crafted SMBv2 packet to the Server service, aka ”SMBv2 Command Value Vulnerability” [20].

(v) Microsoft Windows is prone to a remote code-execution vulnerability that affects RPC (Remote Procedure Call) handling in the Server service. An attacker could exploit this issue to execute arbitrary code with SYSTEM-level privileges. Successful exploitations result in the complete compromise of vulnerable computers. This issue may be prone to widespread automated exploits. Attackers require authenticated access on Windows Vista and Server 2008 platforms to exploit this issue [21].

(vi) CVE-2012-4774: Microsoft Windows XP SP2 and SP3, Windows Server 2003 SP2, Windows Vista SP2, Windows Server 2008 SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 allow remote attackers to execute arbitrary code via a crafted (1) file name or (2) subfolder name that triggers use of unallocated memory as the destination of a copy operation, aka ”Windows Filename Parsing Vulnerability ”[22].

(vii) In situations where the Windows shares are not properly configured, these poorly configured shares can be exposed to the rest of the internet. Attackers can exploit abominably protected shares by exploiting week or null passwords and thus gain access to the administrative shares.

3.2 BRO Network Analyser

BRO is a stand-alone open source network traffic analyzing system [27]. This stand-alone system has the capability to monitor traffic directly and passively using packet filters. It monitors incoming, outgoing as well as internal traffic to trace suspicious traffic or traffic that violates policies. To avoid any intrusion; BRO monitors the traffic and raises alarm as soon as it see unusual (malicious) traffic. BRO performs the detection of malicious activities in real time. BRO transforms traffic into high-level events, based on the policies and configurations alarm is raises for malicious traffic. BRO can be used to analyze real-time flow or pre-recorded flow and packets (pcaps), to extract files from network traffic streams, as an intrusion detection system by enforcing policy, and to generate statistics about

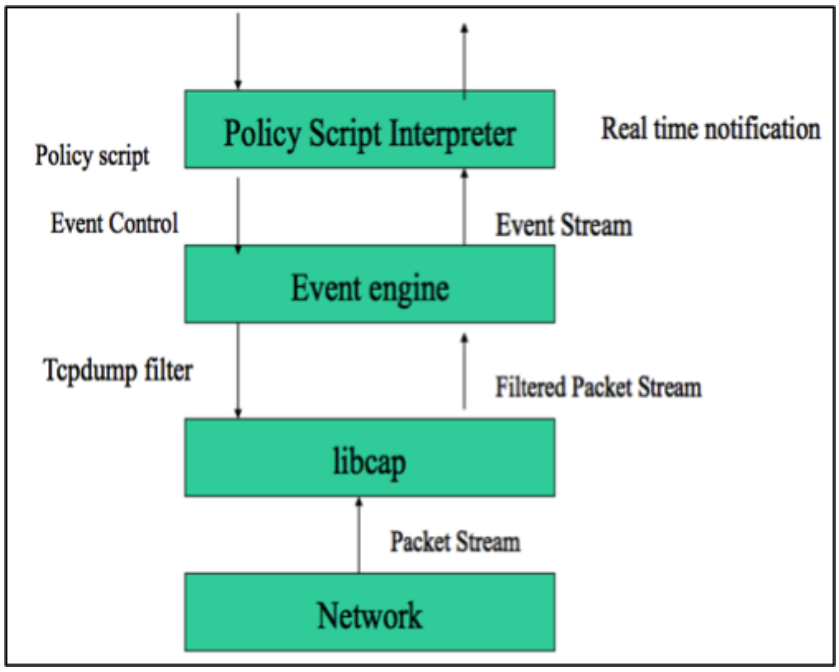

Figure 3.2.1: BRO Architecture

network traffic patterns and usage [5]. BRO supports both signature-based and anomaly-based detection. In this work we research a solution for lateral movement that can be used by BRO. An advantage of anomaly based intrusion detection system is to detect new attacks. We have chosen BRO because it is well-known open-source tool, that is widely used by the entire security community. Bro architecture as shown in Figure 3.2.1 comprises of: the libpcap libraries, an event engine and a policy script interpreter. Below is the brief description of BRO architecture.

3.2.1 LIBPCAP:

Libpcap is a portable packet-capturing library. It extracts packets from the network packet stream. The packets can be FTP, Telnet, Rlogin, IP fragments, SMB and TCP packets (SYN, FIN, RST). Libpcap helps in reducing the load because with libpcap packet of specific protocol can be extracted. Libpcap adds significant advantages to BRO; it isolates BRO from details of the network link technology (Ethernet, FDDI) [27]. And if the host operating system provides a sufficiently powerful kernel packet filter, such as BPF, then libpcap downloads the filter used to reduce the traffic into Kernel. In BRO it is easy to specify which packet to capture by specifying the bits of the TCP header for example

$$tcp[13]\&7! = 0$$

where 13 is static offset in TCP header, it points to 13th octet which contains TCP flags.
This command instructs BRO to capture all TCP packets in which SYN, FIN and RST control bits are

TCP Events	Description
Connection_attempt	When no response is received from the other endpoint to the SYN packet requesting a connection.
Connection_established	When a correct response is received from the other endpoint to the SYN packet requesting a connection.
Connection_rejected	When the other endpoint responds with RST packet to the SYN packet requesting a connection.
Connection_finished	Connection termination with FIN packet.

Figure 3.2.2.1: TCP events

set. Although in packet filters through Snapshot length it is possible to specify the size of a packet to be captured while in the BRO the size of Snapshot length is full packet.

3.2.2 Event Engine

Libpcap hand over the filtered packets to Event Engine. Event engine first checks the packets integrity. If a packet passes the integrity test, BRO then reassembles the IP datagrams, processes TCP/UDP, creates a state for each connection and generates events. BRO includes several protocol analysers that communicate through events. In order to get a BRO script to do something we need an event. Events in network scripting language are things that are called protocol activity [88]. BRO includes protocol analysers for large number of protocols like SMB, TCP, UDP, NTLM, RPC etc. Connection handler determines whether to record the whole packet, or only the header. Event handlers updates state information, generates new events and generate new events. Event handlers are used to write policies. Event handlers are almost syntactically and semantically identical to BRO functions except that they don' t return value. If a packet fails an integrity test, BRO drops the packet and generates an event. Bro processes TCP and UDP packets differently. For a TCP packet, the connections handlers uphold the presence of TCP header and calculates the checksum of the whole packet. If successful, then the header of the packet is checked for control flags (SYN/FIN/RST) and the connection state is adjusted accordingly. If payload is present, a handler is invoked to process the payload. Connection state determines the type of event. Figure 3.2.2.1 shows types of TCP events

UDP Events	Description
udp_request	When a host sends a UDP packet to another host
udp_reply	When the host responds to the request.

Figure 3.2.2.2: UDP events

In case of UDP processing, the events are shown in Figure 3.2.2.2

3.2.3 Policy Script Interpreter

Once Event Engine generates the events, policy script interpreter process the events and forms real time notifications, recording data, or changing the internal state. The policy script interpreter executes scripts written in Bro language. These scripts specify event handlers. Adding new functionality to Bro generally consists of adding a new protocol analyzer to the event engine and then writing new event handlers for the events generated by the analyzer [27].

3.2.4 BRO platform

Paxson et al. [27] describes how BRO network security monitor is built. In this paper they presented an overview of BRO design, BRO language, application specific processing and future possible improvements. Application specific processing includes FTP , Finger, Portmapper, and Telnet. This paper also describes possible attacks against Bro and their mitigations, attacks such as Overloads attacks, crash attacks, and Subterfuge attacks. During the initial development phase there was no support for SMB, so in this paper it is not mentioned Bro supporting SMB protocol. Remote exploitation is an appealing technique for attackers to take control of vulnerable machines, protocols and perform lateral movement. According to the best of our knowledge so far there is no standard evaluation technique or model to implement BRO to detect lateral movement attack through SMB.

3.3 K-nearest neighbour algorithm

K-nearest neighbour algorithm (KNN) is a supervised machine learning algorithm. KNN classify new instances based on stored labelled training samples in the feature space. The distance between

the stored data and the new instance is calculated by means of a similarity measure. This similarity measure is typically calculated by a distance measure such as Euclidean distance, Minkowski distance, or Mahalanobis distance. In other words, the similarity to the data that is already stored in the system is calculated for any new instance that is input into the system[89]. One of the most popular choices to measure this distance is known as Euclidian distance [97]. It is the easiest distance calculation method. It calculates the approximate distances between various points on the input vectors, and then assigns the unlabelled point to the class of its K-nearest neighbours. In our case we have used Euclidian distance. Then, this similarity value is used to perform predictive modelling. Predictive modelling is either classification, assigning a label or a class to the new instance, or regression, assigning a value to the new instance. For our model we have performed classification. In k-NN classification, the output is a class membership [90]. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small) [90]. In the process of creating k-NN classifier, (k) is an important parameter and various (k) values can cause various performances. If k is very huge, the neighbors, which used for prediction, will consume large classification time and affect the prediction accuracy. If k = 1, then the object is simply assigned to the class of that single nearest neighbour. In our case we have selected k=3. After the distance of the new point to all stored data points has been calculated, the distance values are sorted and the k-nearest neighbours are determined. The labels of these neighbours are gathered and a majority vote or weighted vote is used for classification. In other words, the higher the score for a certain data point that was already stored, the more likely that the new instance will receive the same classification as that of the neighbour [89].

3.3.1 Model Validation

Model validation is motivated by two fundamental problems[112]: model selection and performance estimation. Model selection means what is the optimal model for a given classification problem and performance estimation means how to estimate the performance of the selected model. There are different techniques to validate the classifier (model)

(1) **Cross validation method**

Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against [111].

Cross validation includes[113]:

- Splitting the original dataset into k equal parts (folds)
- Takes out one fold aside and performs training over the rest k-1 folds and measures the performance
- Repeat s the process k times by taking different fold each time

(2) **Holdout Method**

The dataset is split into two equal sets: training set and test set. Training set is used to train the classifier while the test set is used to validate the classifier by estimating the error rate of the trained classifier. This method has two drawbacks: it can not be used in problems where we have a sparse dataset we may not be able to afford the luxury of setting aside a portion of the dataset for testing and since it is a single train-and-test experiment, the holdout estimate of error rate can be misleading [112]. This drawback can be avoided with cross validation method.

(3) **Percentage Split**

In this method the dataset is randomly split into two set: a certain percentage of the dataset used to train and the rest used for testing. For example 80% of the dataset is used for training the classifier and 20% of the dataset is used to validate the classifier [114].

Chapter 4

Lateral Movement Attack techniques

As we discussed in the previous chapter that SMB protocol is vulnerable to large number of exploits. Once these SMB vulnerabilities are exploited, below are the tools and methods that attacker can use to perform lateral movement attack. In this project we are discussing three main methods and tools, PsExec[99], WMI[101] and Pass the hash.

4.0.1 PsExec

PsExec[99] is part of PsTools suite owned by Microsoft. PsExec is a lightweight telnet-replacement that lets you execute processes on other systems, without having to manually install client software. PsExec's most powerful uses include launching interactive command-prompts on remote systems and remote enabling tools like IpConfig that otherwise do not have the ability to show information about remote systems [1]. Most of the network administration tools are developed to provide flexibility in managing interconnected devices, and services. As long as these tools are used by legitimate people and for legitimate purposes so the purpose of these tools is achieved. But if misused for malicious activities it can be destructive. It is the same case with PsExec. PsExec can be used for malicious activities. For the PsExec to work, two requirements must be fulfilled, first requirement is both local and remote computers must enable file and print sharing and second requirement is remote computer must have defined Admin$ share. PsExec command given below can be used to execute malicious code on remote machine.

```
Psexec.exe \\IP_address_of_remote_machine −u username −p password maclicious.exe
```

When PsExec is executed it defaults to the $\%SYSTEM\%$ directory on the remote system so there is no need to specify a full path.

How the attack works

PsExec[99] is not only valuable for administrator but for the attacker as well. As long as the attacker has the credentials of the victim machine, using PsExec[99] the attacker can upload malicious code on the victim machine. Let us suppose a scenario as shown in Figure 4.0.1.1, in which the attacker has a malicious file (.exe), which the attacker wants to execute on a victim machine. Using the victim credentials the attacker authenticates to the victim machine and try to gain access to the Admin$ share (C:\Windows). Access to Admin$ share is important specially when deploying software (in our case malicious code) to the victim machine. Credentials supplied to PsExec[99] have the permission to access Admin$ share. Once the attacker get access to the Admin$, he can push the executable (.exe) malicious code into the Admin$, IPC$ shares. Now the attacker makes a separate call to the Remote Procedure Call (RPC) on the victim machine, which is running over the SMB protocol. Through RPC the attacker can talk with the Service Control Manager (SCM). SCM is maintaining and managing all the services running in the background. SCM loads the executable (.exe) and treat it as a service. Once the SCM puts the malicious .exe file in the memory the service shutdowns.
The strength of this attack can be understood from the fact that once the attacker exploits one machine now the attacker can exploit other machines in the network with proper tools: METSPLOIT[102].

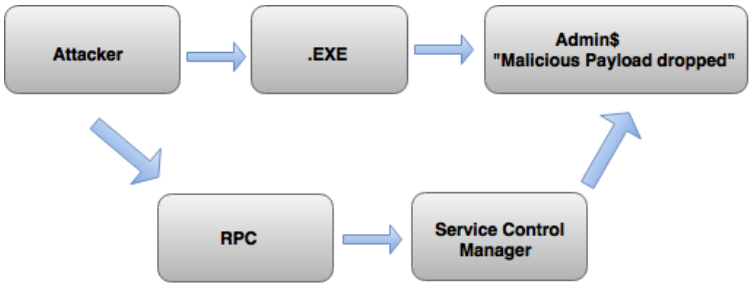

Figure 4.0.1.1: PsExec Attack scenario

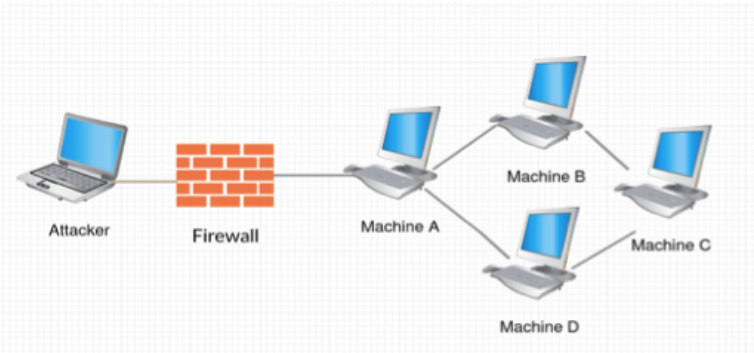

Figure 4.0.1.2: Local Authentication Token

Thus leading to lateral movement attack. There are many ways to perform lateral movement after the initial compromise. Below are the two methods that are briefly explained.

(1) **Local Authentication Tokens: Session abuse**

In this case the attacker exploits the authentication tokens. Suppose we have a network as shown in Figure 4.0.1.2, which consists of 4 machines i.e. machine A, machine B, machine C and machine D. All these machines are authenticated to each other. User on machine A is authorized to access machine B, C, D. The goal of the attacker is to access Machine B, C, D. But the attacker cant directly access machine B, C, or D. If the attacker authenticates to machine A then from machine A he can access machine B, machine C and machine D. So if the attacker uses PsExec[99] to authenticate to machine A, now PsExec[99] is a local user that uses the local authentication token of the current running process. Now as the machine A is compromised and PsExec[99] is running as local user on it, machine A can now PsExec[99] from itself to machine B, C, D. Thus allowing attacker to access machine B, C, D. Now the attacker can also execute commands on these machines.

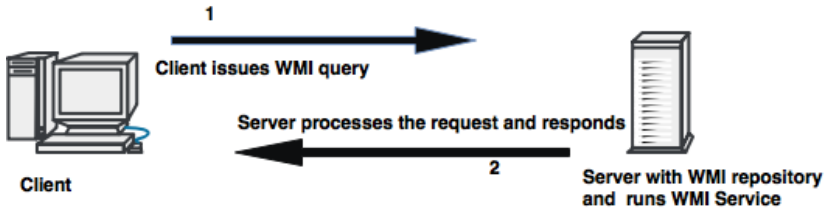

Figure 4.0.2.1: WMI client and server. Server in the Figure is a remote computer for which the attacker has the credentials and wants to know the operating system or hardware configurations and wants to start a service or process on it using DCOM or WinRM.

(2) **Credential Reuse**

In large environments the usage of same credential for many accounts and machines is very common. Credential reuse might make the System Admin task easier to remember only one password for many machines, but it is problematic as well because if the attacker got access to one password he can access all the machines. Credential reuse exploitation happens when many machines are having same password or hashes. For example if the attacker obtains the NTLM hashes of one SMB target, he can reuse the same hashes against other SMB targets in the network. The attacker can use METASPLOIT tool Credential Domino Meta Module for credential reuse. It provides an automated credential reuse. If the attacker comprises one machine in the network, the tool does the rest. With the compromised one machine and the tool in place, for the attacker it is very easy to do lateral movement across the network. Credential reuse is mostly used against the services like SMB, SNMP, SSH, TELNET, MSSQL, and MySQL. Credential reuse is serious issue because if a system thats vulnerable to credential reuse exploitation allow the attacker to bypass all the security controls and patches. Tool like Credential Domino can be used to abuse credential reuse. It shows machines that are accessible from each other using credential reuse.

4.0.2 Windows Management Instrumentation

Windows management instrumentation (WMI)[101] is a tool that is implemented as service to locally and remotely manages data, operations and configuring settings on windows operating systems.
WMI allows the administrator to see how the Operating system operates, what are its configurations and properties and to automatically collect a systems hardware and software data. Similarly, the attacker as shown in Figure 4.0.2.1 can also use WMI to acquire configuration and other information about the remote machines. WMI has its own query language called WQL. It also supports other scripting languages like Windows Script Host, VBScript, and PowerShell. WMI can be interacted

locally and remotely. The reason WMI is so powerful remotely is the fact that, there are many WMI events, which allow administrator and attacker to install backdoor, code execution as well as do lateral movement. There are two protocols (DCOM and WinRM) that give the system administrator the ability of remote object queries, event registration, WMI class execution and call creation. For more in- depth information about WMI see[13].

WMI Eventing

WMI can generate customized events that can be triggered when a process is created or a specific event happens. This is very useful for administrator because he can configure WMI to trigger an event when a process is created or an event happens. On the other hand, attackers can also leverage this WMI feature and use it for malicious purposes. For example he can install a persistent backdoor (run the malicious code) every time the system restarts. Figure 4.0.2.2 shows the simplified overview of WMI. In order to trigger off an event there are three requirements

(i) **Event Filter**
Through Event Filter the admin or attacker can specify the events of their interest. Event like systems restarts, user logoff, new process is created.

(ii) **Event Consumer**
If the event that is specified in Event Filter triggers, in Event Consumer the Admin or attacker specify action of their interest to happen. The action can be i.e. run the executable script.

(iii) **Binding Filter and consumer**
Once the Event Filter and Event Consumer are created, in Binding Filter and Consumer an association is created between Event Filter and Event Consumer.

Based on the above three specifications, a WMI event triggers.

How the attack works

Below are two methods which shows how WMI is used in lateral movement attack.

(1) **Creating Malicious Event**

DCOM and WinRM are handy tools for system administrator, but these tools can also be use for malicious purposes especially when its traffic is not inspected or filtered for malicious scripts or code and also no anti-virus can detect it. To fully exploit the WMI, the attacker only needs valid credentials or hashes. With the WMI the system administrator can specify which action to happen when a certain event happens. For instance when a process is created, make an alert.

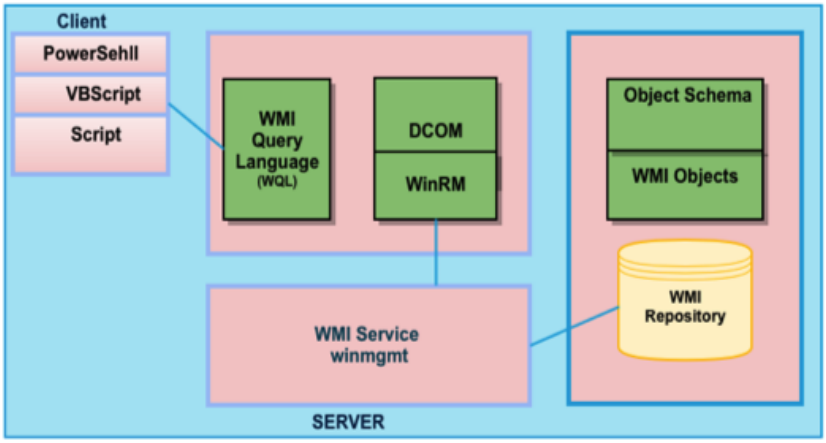

Figure 4.0.2.2: simplified High overview of WMI. Simplified and redesigned inspired from [13]

As the attacker has the valid credentials (password or hashes) he can specify malicious events to happen for example every time the victim restarts the computer run the executable that is present in the specified directory. In the example below, the attacker defines the WMI Eventing requirements as; Event filter as System restart, Event Consumer as run the executable with SYSTEM privileges that has been dropped before and Binding (FilterToConsumerBinding) binds the filter and consumer. This attack is a persistent attack; every time the system restarts the malicious code gets executed on the victim machine. Event Consumers like ActiveScriptEventConsumer and CommandLineEventConsumer allows attacker to execute any payload against the victim machine.

(2) **Win32_Process Create Method**

In this attack, the attacker is using WMI Win32_Process class as a mean of attack. The top-level of the WMI structure is the namespaces. Namespaces are referred to different components of the computer like (DNS, Windows, SQL). Classes are inside the namespace i.e. class for battery. These classes also have methods. There are methods to see the management information, methods to spawn a process remotely or locally and to change the configuration of the local or remote machine. CREATE method is what attracts attacker attention. With this method the attacker can start a local or remote process against the target and allow the attacker to run malicious executable against the target system. CREATE method is a better alternative of PsExec[99]. Below is an example of executing process on remote machine using CREATE method [13]

```
PS C:\> Invoke-WmiMethod -Class Win32_Process -name Create -ArgumentList "notepad.exe"
-ComputerName 192.168.178.134 -Credential "Win-885aa7st\Administrator"
```

4.0.3 Pass the Hashes

This is a type of attack in which the attacker steals victim hashes. Without even cracking the hashes, the attacker uses the hashes to impersonate the victim and login into the victim account or system. This attack be carried out against any machine but it is more famous against Windows machines. Attackers are interested in hashes because many machines can be accessed with the same hash, remote machines accept hashes so the attacker does not need to bother about cracking it, and Windows does not support salting so the hashes remain the same as long as the password is changed. The lack of a salted NTLM hashed password value ultimately means you do not need to crack a password in Windows, all you need to do is simply dump the hashes from memory and pass that to access a remote system. Detecting pass the hash attack will limit the impact and help in detecting lateral movement. Any system using LM or NTLM authentication in combination with any communication protocol (SMB, FTP, RPC, HTTP etc.) is at risk from pass the hash attack. The exploit is very difficult to defend against, because there are countless exploits in Windows and applications running on Windows that can be used by an attacker to elevate their privileges and then carry out the hash harvesting that facilitates the attack. Also there are many techniques that allows attacker to steal the hashes; using administrator privileges he dumps the hashes from Local Security Authority Subsystem (LSASS) this is the place where hashes are saved, this type of attack is called file-based attack or the attacker injects malicious code into LSASS process and dumps the hashes this type of attack is called process-based attack.

(1) **Dumping hashes from Local Security Authority Subsystem Service of the user machine (File based attack)**

Local Security Authority Subsystem and Service (lsass.exe) is a process in Microsoft Windows Operating systems; that is responsible for enforcing the security policy on the system. It verifies users logging onto a Windows computer or server, handles password changes and creates access tokens [15]. In other words lsass.exe allows or prevents a user from accessing files or locations in system based on the access control or security policy. lssas.exe supports many authentication protocols like NTLM, and Kerberos. Depending on the type of authentication protocols, the passwords are cached in different format. In case of NTLM, lsass.exe caches the user password in the form of hashes. There are many ways to dump lsass.exe mostly using legitimate tools installed in windows systems such as PRODUMP. First the attacker must have admin privileges of the victim machine. Once the attacker has successfully compromise the remote machine

then he can use PRODUMP tool to dump the memory of lsass.exe process, which contains the users credentials. So now the attacker has to search this memory and find the username and hashes. Command to dump lsass.exe process.

```
C:\Prodump> prodump.exe ma -accepteula lsass.exe thedump.dmp
```

Once the lsass.exe process is dumped, the attacker downloads thedump.dmp file to his machine from the victim machine. The attacker then uses Mimikatz tool to extract the usernames and plain passwords from the thedump.dmp file. Mimikatz commands [16]

```
Mimikatz #sekurlsa::minidump lsass.dmp
Switich to Minidump
Mimikatz # sekurlsa::logonPasswords full
```

As the plaintext passwords are extracted, the attack would become pass-the-password instead of pass-the-hash.

(2) **Process-based attack**

LSASS constantly communicates with a database file on the file system called NTDS.dit. This is the database that the Active Directory syncs it information to, about every five minutes usually. Which means that all the usernames and password hashes are stored in NTDS.dit. Only LSASS process can access this database and is locked for user or attacker. What the attacker does is, to utilize Volume Shadow Copy Service (VSS). VSS is windows based backup service. The attacker asks VSS to backup the C volume, assuming that the NTDS.dit is in C volume. Thus once the attacker got the backup of the C volume, he search into there for the NTDS.dit and pulls it out. Using tools like NTDS Extract; the attacker can find the username and hashes [17]. VSS is by default OFF in Windows server 2012. The attacker can ON it. But normally this raises red flag. Figure 4.0.3.1 shows the process based attack.

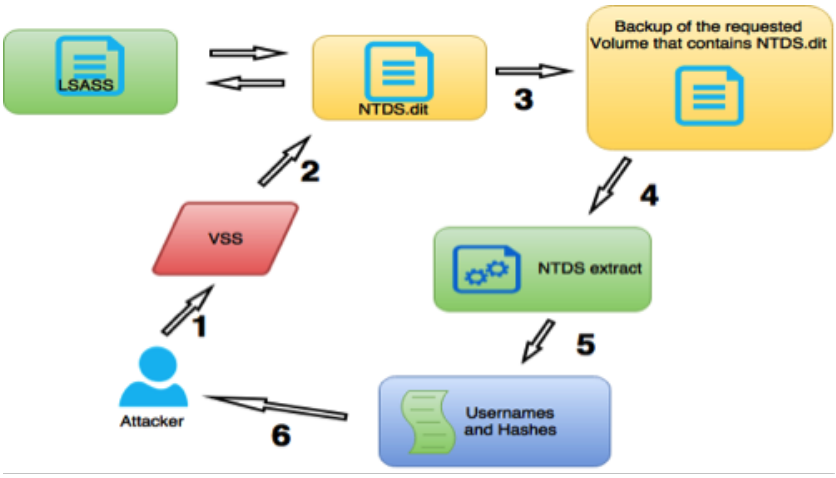

Figure 4.0.3.1: Pass-the-Hash (Process based attack)

Chapter 5

Our Model

This chapter discusses the complete description of our lateral movement detection approach, along with the description of datasets we used for the analysis and testing and finally the statistical evaluations of the approach.

5.1 Description of Datasets

The datasets that are analysed and tested in our analysis and evaluations are from multiple sources:

5.1.1 A Home lab setup

Figure 5.1.3.1 shows the architecture of the Home lab. To collect the datasets, Wireshark is running on Windows 8 OS and Windows 7 OS. Two types of datasets are generated at Home, normal datasets and datasets containing lateral movement attacks. In normal datasets real administrator

behaviour is simulated, while malicious datasets contain attacker behaviour. Tools like PsExec[99], PsExec.PY[100], Windows management instrumentations[101], Metasploit[102], Windows Management Instrumentation Command-line[103], and Veil-Evasion[104] are used to generate normal and malicious datasets. The total size of the datasets generated at Home lab is 459Mb. The total time duration of these datasets is approximately 13 hours.

5.1.2 An advanced forensics challenge [98]

On 23rd August 2014 Machnetico company discovered strange traffic on their network for two days. The forensics challenge is, being an incident response specialist to investigate the breach on Machnetico network. Advance forensics challenge datasets duration is two days containing both normal network traffic and one malicious lateral movement attack traffic. The total size of advanced forensics challenge datasets is 17.7Gb.

5.1.3 Datasets from a real corporate network

Real corporate network datasets are from medium sized corporate network containing both normal and malicious datasets. The malicious datasets contain one attack (lateral movement) where first one user account is compromised by guessing weak passwords. After getting hold of the compromised credentials, the attacker started using Meterpreter[105] malware and finally the attacker gained control over the whole windows domain and finally all workstations and servers were compromised. The total time duration of datasets from real corporate network is approximately 9 hours.

We divided the datasets discussed in Section 5.1.1, 5.1.2 and 5.1.3 into smaller datasets. Each smaller dataset is of time duration T=10 minutes. We divide it in smaller datasets because that is how our detection mechanism works. Approximately 150 small normal and malicious datasets are used for analysis and testing purposes. All the datasets contain both SMB1 and SMB2 traffic from Windows 7 and Windows 8 operating systems and Windows Servers.

5.2 Research Strategy

The main challenge in detecting lateral movement attack is that the attacker is using legit tools: PsExec[99], WMI[101]. As this attack is targeted so it is conducted very precisely and throughout the attack, the attacker tries to stay under the radar being undetected. Also the attacker is using credentials of authorized users which makes it further hard to detect. Initially we analysed the datasets we generated at Home lab. In which protocols like SMB, NT LAN Manager (NTLM), NT LAN Manager

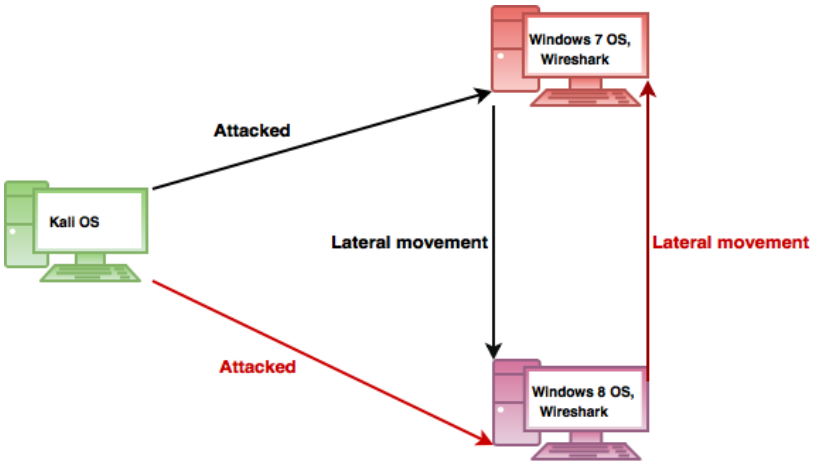

Figure 5.1.3.1: Architecture of the home-lab

Security Support Provider (NTLMSSP), Remote Procedure Call (RPC) are exploited to generate malicious datasets and these protocols are used normally to generate normal datasets. Tools like WIRESHARK[106], Network miner[107], Scapy library[108] and BRO-network analyser [109] are used to analyse the traffic. During our analysis we analysed the behaviour of these protocols in normal traffic datasets and as well as malicious traffic. By looking at the traffic, we considered nine possible features we thought could help us detecting lateral movement, which are described in Section 5.3. After this initial brainstorming phase, we have deeply investigate those features and removed some of them from our list, because we believed were not effective enough in detecting lateral movement. In the next Section we motivate our reasons for exclusion of those features. However, we did not scientifically proved that they are effective for the detection. So finally we decided on five detection techniques, which we have implemented and evaluated to detect anomalous behaviour. The results of the evaluation, discussed in Section5.6, tell us that these features can precisely identify possible lateral movement attacks.

5.3 Initial detection techniques

As mentioned earlier, initially we researched nine techniques to detect lateral movement which are described below

(1) **Known IP pairing**

In this method the amount of data transferred between the hosts is monitored. For example suppose there are three hosts in the network with IP addresses 192.178.168.1, 192.178.168.2, 192.178.168.3. And 192.178.168.1 is connected to 192.178.168.2 and 192.178.168.2 to 192.178.168.3.

In normal situations the average data transferred between these hosts is 1MB in 1 hour. And attacker has no information about these statistics. During the attack if the data transferred is higher than 1MB in 1 hour or suppose the attacker tries to connect from 192.178.168.1 to 192.178.168.3 which in normal cases they are not connected these activities can be used to differentiate between normal system behaviour and attack behaviour.

Problems of this technique

This technique may work for static network, such as SCADA but not in dynamic environments where the number of hosts connected can change, different protocols can be used, and data transferred between hosts changes. Maintaining and updating the IP pairing states is a hard and error prone task.

(2) **Extract RPC commands from Named PIPE**

PSExec has a Windows Service image inside of its executable. It takes this service and deploys it to the Admin$, IPC$ share on the remote machine. It then uses the DCE/RPC interface over SMB to access the Windows Service Control Manager API. It starts the PSExec service on the remote machine. The PSExec service then creates a named pipe that can be used to send commands to the system [67]. Some of the most frequently commands used are (Xcopy, copy, upload, dump, getsystem etc). This detection technique can be used to detect malicious remote access by differentiating number and type of commands send in normal situations and in malicious situations.

Problems of this technique

With further research we find out that the effectiveness of this technique is in question because some of the commands used by attacker are also used by administrator for legitimate purposes, also there are different types of RPC systems and each system supports specific type of RPC protocol. While the attacker can also write his own RPC calls and simply name them as other legitimate RPC calls and can simply bypass this technique.

(3) **Remote access through Phishing**

The most common goal of phishing attack is tricking victims into providing their passwords or credentials without even being aware of it[115]. During the lateral movement attack the attacker can get the victim credentials through phishing. The attacker generates reverse TCP malicious

payload, sends the payload to the victim and tricks the victim to execute it. Soon after the victim executes the payloads it gets executed initiating TCP connection to the attacker and the attacker gets shell on the victim machine. In lateral movement attack, the attacker can use phishing attack to bypass the firewall that blocks incoming connections because executing the malicious payload the victims requests connection to malicious server. Phishing attack allow the attacker to exploit SMB protocol automatic authentication described in Section 3.1.1.

Problems of this technique

There are many existing practical detection techniques to detect phishing attacks [72][73][74][75].

(4) **Average size and Average number**

In this technique the normal and malicious behaviour is differentiated by calculating average number of machines accessed in one session by administrator and the average session duration.

Problems of this technique

This technique is also vulnerable to large number of false alarms and is impractical because in a very large dynamic environments the number of computers might increase or decrease, and for certain remote activities the administrator might need more time and for certain less time.

(5) **Random hostname**

One strategy of the attacker is to bypass anti virus and IDS policies. Most anti virus and IDS exhibit policies which controls the number of connections or session initiated from a specific host. A host is blocked if it initiates more connection then the normal scenario. So in order to evade this, the attacker randomizes the host name for each connection or session. Random hostname in the network traffic might connote pass the hash attack or lateral movement attack. Tools generate random computer names when passing the hash, so aggregating on Workstation name can be quickly used for detection, strange or new hostnames in the network is the identification of Pass the hashes [70]. Random hostname can be extracted from SMB HEADER (Security blob) . After investigating malicious lateral movement traffic, it includes random hostname. In legit traffic the host name is not random, but instead a real hostname of the user logged in. This technique can be very useful in detecting lateral movement attack.

(6) **Empty session key**

Session is an authenticated context established between an SMB client and server. Authenticated context is runtime state that is associated with the successful authentication of a security principal between the client and the server, such as the security principal itself, the cryptographic key that was generated during authentication, and the rights and privileges of this security principal [80]. Session key is relatively short-lived symmetric key (a cryptographic key negotiated by the client and the server based on a shared secret). A session key's lifespan is bounded by the session to which it is associated. Session key is used to identify a user and to encrypt the information sent across the network [81]. NTLM protocol uses session key to encrypt information which are sent across the network. Session key algorithm is used to generate session key. Session key algorithm uses response message algorithm to generate session key. This algorithms takes three parameters. 8-bytes challenge server challenge, a 16-bytes key, and three 8-bytes signatures.

```
CalculateChallengeResponse (challenge, Key, LMResp)
```

16 bytes Key is generated by placing the user password in the first 8-bytes and the remaining 8-bytes is fill with 0xbd.

```
LMKey = substr (password, 0, 7)
LMKey = fill (8,15,0xbd)
```

CalculateChallengeResponse function generates three 8-byte values of which the first 16-bytes is valid session key [77].

Pass the hash attack allows the attacker to use the hashes as an alternative of password as shown in Figure 5.3.1 and access remote machines because NTLM hash is equivalent to clear password to authenticate remotely. While analysing pass the hash attack traffic we identified that Pass the hash generates empty session key however in normal NTLM authenctication (password is provided) the session key is not empty. We believe that empty session key is one identification of Pass the hash attack. We presume there are three reasons behind the emptiness of the session key.

(a) Our first assumption is that during Pass the hash the attacker bypass standard NTLM authentication procedure in which password is required by the NTLM session key algorithm to generate session key, instead the attacker provides hashes so the session key algorithm won't be able to generate the session key. Because in connection oriented authentication,

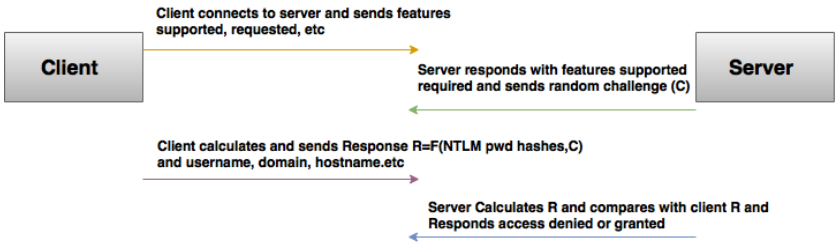

Figure 5.3.1: How pass the hash is performed. Redesigned and inspired from [76]

the client and server calculate the session key based on the user password.

(b) Another reason behind empty session key can be that if the user provides username and password for remote authentication, the hashes will be generated and thus session key. While if the user provides username and hashes for remote authentication no session key be generated because valid hashes indicates that session key already exists for this connection.

(c) A third reason can be, during the NTLM negotiation, the client and server negotiate certain flags. Among them one flag is U. If set, requests session key negotiation. Otherwise it is ignored. So during the Pass the hash, this flag is not set, thus no session key is generated [82]. Also not setting this flag is in favour of attacker in case the attacker wants to be anonymous, because session key is used to identify the user logged in. Anonymity is crucial when the attacker is insider and if certain IDS or firewall doesn't allow multiple sessions from the same user. The Null Session Key is employed when Anonymous authentication is performed [79]. Even Microsoft SMB documentation claims that any session having empty session key should be blocked; "if Session.SessionKey is NULL, the server MUST fail the request with STATUS_NOT_SUPPORTED and MUST stop processing the request" [83]. As emptiness of the session key behaviour is found in malicious datasets, it is indeed an anomaly and exploitation of NTLM authentication.

(7) **WRITE command and IPC share**

During the detection of malicious lateral movement attack it is crucial to monitor access to shared folders or SMB shares. There are certain share which are of high significance in performing the lateral movement attack. Shares such as IPC$, Admin$, and C$. IPC$ is a default share in

Windows Operating systems, used to send commands to servers. Windows Servers uses IPC$ to receive remote procedure calls (RPC). RPC calls can be used to Create, Start and Stop remote services and modify the Security Account Manager (SAM) database. SAM database stores passwords of the users and are used to authenticate both remote and local users. RPC calls are sent over named pipes. Named pipes are one method of inter-process communication in which different processes communicate with each other i.e one process on the client side and other process on the server side. Named pipes are used to exchange data about a specific task. IPC$ share exploit is a very common attack among Chinese attackers to exploit remote machines. Efficient attack tools (e.g. Fluxay) were developed to reconnaissance and compromise machines vulnerable to IPC$ exploits [69]. Exploiting IPC$ share allows the attacker to gain access to the server with full administrative privileges. Most of the cases IPC$ can be accessed using default or weak administrative passwords, or the attacker tries to guess and performs brute force attack. Many worms come up recently which make use of the weak administrator password and infects the Windows machines via IPC$. Example given bellows shows the attacker uses dictionary words to brute-force the target password via IPC$ share [69]. Script to perform dictionary attack is shown in Figure 5.3.2. The attacker can use the following script to perform the brute force attack against multiple computers at the same time. Once the attacker gets the correct administrator password he then logon as administrator and transfer malicious executable to the target machine and gets executed. WRITE command is of importance because using this command the attacker is either uploading files or making changes in the victim machine configuration. In this technique the number of write commands towards IPC$ is monitored.

(8) **Error Messages**

Once one system is compromised, lateral movement allow attacker to access and compromise all the systems in the networks. To understand intrinsic security dependencies, it is important to know the relationships between accounts and access privileges across all systems on a network [68] . Unlike administrator, the attacker has limited or no information about the relationships between users' accounts and access privileges of systems in the network. This inadequacy of the attacker generates good identification of malicious behaviour in the network through generating error messages.Error messages such as Access Denied, Logon Failure. After analysing the error messages in both normal and malicious datasets we find out that there is huge difference between the error messages generated in normal network behaviour and in malicious behaviour. Error messages such as Access Denied and Logon Failure are of significant importance while investigating malicious lateral movement attack. During the lateral movement the attacker might

```
#IPC$crack
#Created by Mnemonix 1st of May 1998

$victim = $ARGV[0];
$user = $ARGV[1];
open (OUTPUT, ">c:\net.txt");
open (PASSWORD, "c:\passwd.txt");
$passwd = <PASSWORD>;
while($passwd ne "")
{
        chop ($passwd);
        $line = system ("net use \\\\$victim\\ipc\$ $passwd /user:$user ");
        if ($line eq "0")
            {
                print OUTPUT ("$user\'s password on $victim is $passwd");
                $passwd = "";
            }
else
        {
                $passwd = <PASSWORD>;
                if ($passwd eq "")
                        {
                                print OUTPUT ("Not cracked.");
                        }

        }
}
```

Figure 5.3.2: Dictionary attack script

guess passwords, use wrong hashes or try to access directories of which the attacker is not authorized so there can be large number of error messages. Based on the analysis of the malicious traffic it is obvious that during the initial phase of the attack, the attacker tries to acquire credentials of all the users, so the traffic contains a large number of Access Denied and Logon Failure messages. Once the attacker gets the correct passwords of all users, the remaining traffic contains access to IPC$ share, Admin$ share, and initiation of services. In this technique the number of error messages: Access Denied and Logon Failure are monitored.

(9) **Services Created and Started**

One important technique to identify anomalous lateral movement activities is to reckon the number and type of services running on the victim computer, because after the successful access to all credentials and exploitation of IPC$ share, the attacker immediately install malicious services on the victim machine. The number of Create service and Start service in normal and malicious traffic are comparably different.

Entropy range of normal hostnames	Entropy range of malicious hostnames
1.53–3.46	3.51–4.0

Table 5.1: Entropy range of hostnames

5.4 Implemented detection techniques

Considering the time limitation of this research work and practical impact of some of the techniques we narrow down to the following five detection techniques

5. **Random hostname**

   We claim that any traffic containing random hostnames clearly signifies the possibility of unwanted activities in the network. The major reason we claim this is that we haven't seen this behaviour in normal traffic, only malicious traffic contained this behaviour. In this technique the entropy of each hostname is calculated. If a hostname has entropy above a threshold (3.5), an alarm is triggered indicating anomalous behaviour. The calculation of entropy is based on the information density expressed as a number of bits per character. BRO function `find_entropy` is used to calculate the entropy. The entropy of the hostname depends on character set used (lower-case character, higher-case character, numbers,length, symbol used, and random distribution of characters). We select the value (3.5) because after a certain number of tests, we find out that normal hostnames entropy is below 3.5 and the entropy of malicious random hostname we detected in malicious traffic is always above 3.5 as shown in table 5.1. Mostly in normal hostnames, higher-case characters are followed by lower-case characters and seldomly by numbers and very rarely includes special symbols (i.e. &). In such scenario the entropy is below 3.5. Lets consider a normal hostname for example `JohnSmith123` which has entropy of 3.42. While in malicious random hostnames the lower-case, higher-case characters and numbers are randomly distributed as well as it's length is 16 characters long. Such malicious random hostname has always entropy above 3.5. A malicious random hostname we detected `3rDnOHztW4SiHGj` has 3.75 entropy and another malicious random hostname we detected `Y2vpEf0hFEW7K5sC` has 3.87 entropy.

6. **Empty session key**

   As mentioned before empty session key is an indication of anomalous network behaviour. Our model flags an anomaly as it finds empty session key in NTLM authentication.

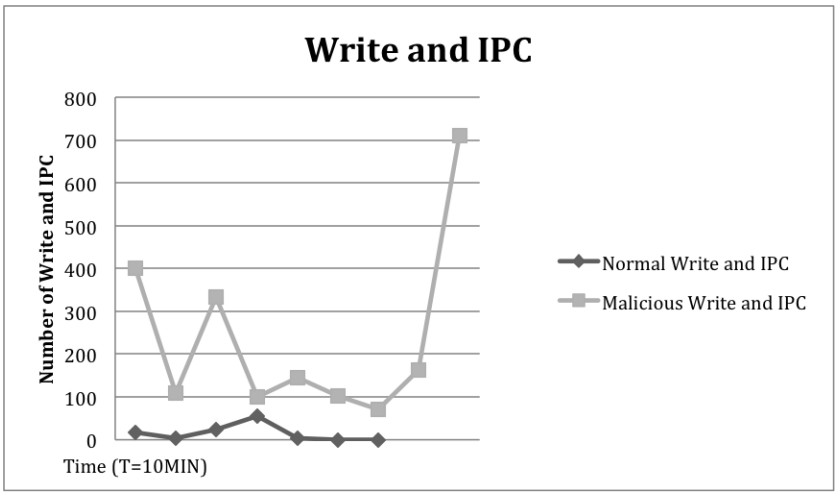

Figure 5.4.1: Comparison between number of Normal and Malicious Write commands towards IPC$ in Home lab datasets.

7. **Prevalence of WRITE command towards IPC share**

   The reason we consider this technique of high significance because there is a clear difference between number of WRITE commands towards IPC$ in normal and malicious traffic. With the Figure 5.4.1 we support this claim. From Home lab datasets, we extracted number of Write commands towards IPC$ from normal and malicious traffic datasets. As we can see in Figure 5.4.1 in normal datasets the number of WRITE commands towards IPC$ is very low, while in malicious datasets they are huge for the reasons mentioned earlier. Each of the dataset is of time stamp T=10 minutes. Figure 5.4.1 shows the comparison between number of Write commands towards IPC$ in normal and malicious datasets generated in Home lab.

8. **Prevalence of Error Messages**

   In order to access IPC$ administrator privileges are required. Rarely the attackers have administrator privileges of all the machines in the network while in most of the cases the attacker doesn't have all theses credentials, so he has the option to perform brute force attack or guess passwords. Before the correct password is hit, large number of wrong passwords are tried. Every time a wrong password is enter, NTLM authentication protocol generates access denied or logon failure message. We are manoeuvring this behaviour as an indication of network exploitation. In normal traffic the number of error messages negligibly small, while malicious traffic it is rather too much. In some malicious traffic the number of error messages are above thousands in just few hours. Each of the dataset is of time stamp T=10 minutes. Figure 5.4.2 shows the comparison between number of normal and malicious error messages in Home lab

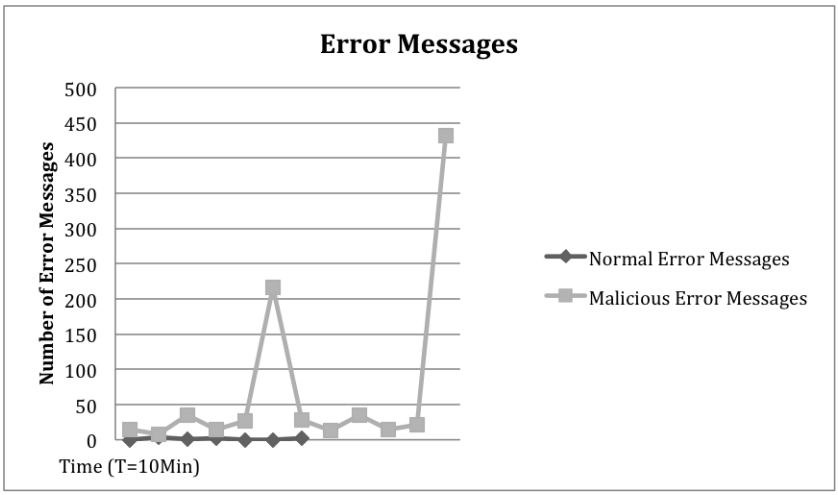

Figure 5.4.2: Comparison between number of normal and malicious Error Messages in Home lab datasets.

datasets.

9. **Prevalence of Services Created and Started**

   The attacker copy malicious executables to the victim machine and installs these executables which run as services in order to have an ongoing access to the victim machine. The number of remote services installed and running in anomalous situations are much higher than normal behaviour of the network. Each of the dataset is of time stamp T=10 minutes. Figure 5.4.3 shows the comparison between number of Services Started, Created in normal and malicious datasets generated in Home lab.

5.5 Our detection model

The detection techniques we implemented are quite effective in differentiating normal traffic from malicious traffic. Random hostname and empty session key if present in any traffic immediately indicates malicious behaviour because they are only present in malicious traffic and identifies the exploitation of normal protocols behaviour. This is demonstrated in Figure 5.5.1 and in Algorithm 1 and 2. Therefore for these two techniques no machine learning algorithm is used to learn threshold. While the remaining three detection techniques imply malicious behaviour if their number are above a fixed threshold. In the last three detection techniques the threshold is learned using supervised machine learning algorithms. The threshold is learned individually for each of the detection technique.

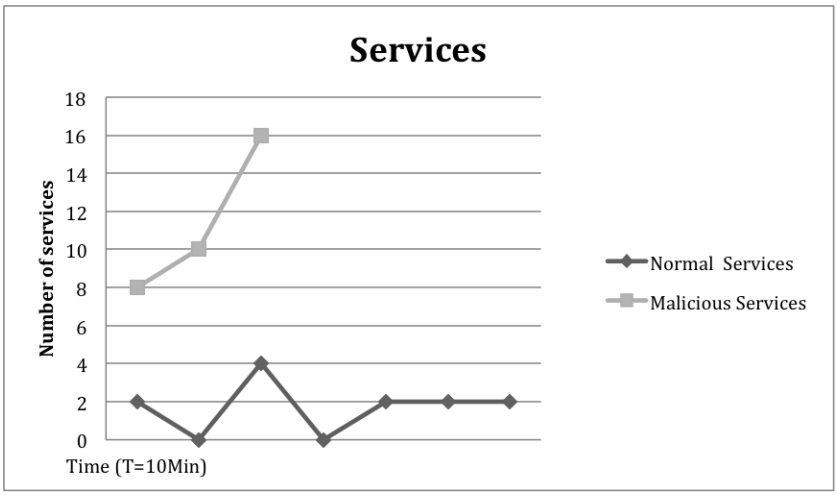

Figure 5.4.3: Comparison between the number of Normal and Malicious Services Created,Started in Home lab datasets

Once the thresholds are learned, then these thresholds are embedded in the last three policies: error messages, Write command towards IPC$ and services started, and created. Figure 5.5.2 shows the training and phase of machine learning algorithm to learn threshold and embedding of threshold in policies. Figure 5.5.3 shows the testing phase of the model. Algorithms 3, 4 and 5 demonstrate the testing phase of the last three policies: error messages, Write command towards IPC$ and services started, and created.

Algorithm 1: Detecting Empty session key in traffic dataset

```
1 event ntlm_authenticate ();
  Input  : Traffic dataset
  Output: Empty session key detected
2 if request$sessionkey = empty then
3 |   return Empty session key detected;
4 else
5 end
```

5.5.1 Description of our model

In this section we have used the datasets from Home lab for training and testing of the model. We have written five policies, one policy for each of the five detection techniques. Empty session key policy, monitors NTLM authentication. During NTLM authentication between the SMB client and Server

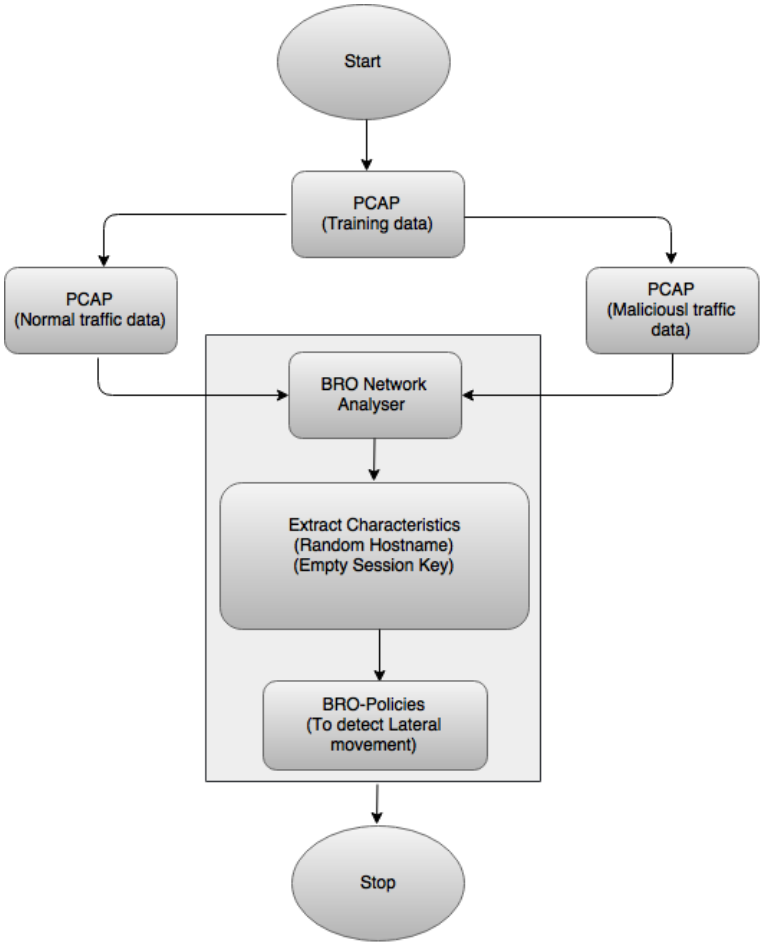

Figure 5.5.1: Extraction of Empty session key and random hostname.

Algorithm 2: Detecting Random Hostname

```
1 event ntlm_authenticate ();
  Input  : Traffic dataset, set hostname entropy threshold
  Output: Random hostname detected
2 if entropy_hostname > 3.50 then
3 |   return random hostname detected;
4 else
5 end
```

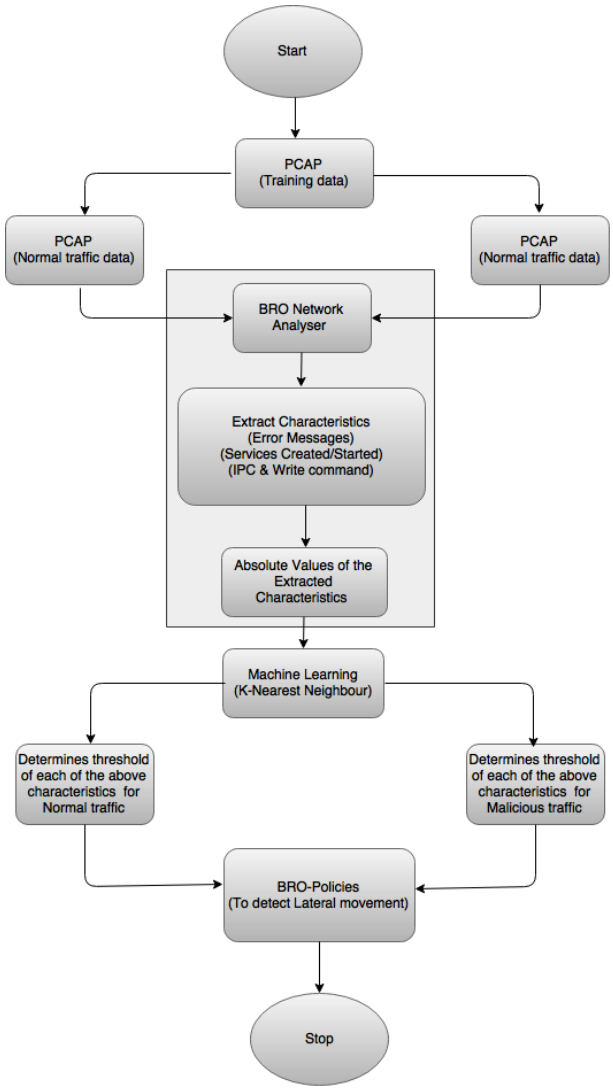

Figure 5.5.2: Training K-nearest machine learning algorithm.

Algorithm 3: Prevalence of Error messages

event smb1_message ();

event smb2_message ();

Total error messages SMB1 SMB2 = $Total\_errors$;

Input : Traffic dataset, set Error messages threshold, set time T

Output: Error messages above normal threshold detected

if $Total\_errors > set\ threshold$ **then**

    return Error messages are above the normal threshold;

else

end

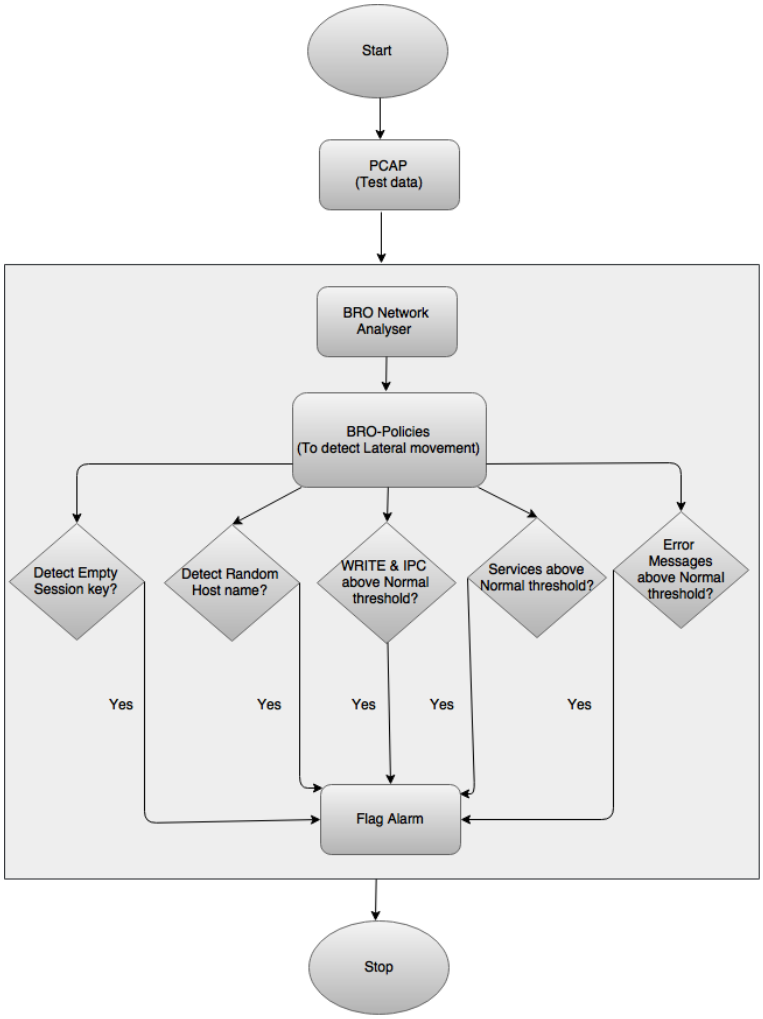

Figure 5.5.3: Testing phase of BRO policies.

Algorithm 4: Prevalence of WRITE commands towards IPC

```
event smb1_message ();
event smb2_message ();
event smb2_write_request ();
Total WRITE commands towards IPC in SMB1 SMB2 = Total_WRITE_commands;
Input  : Traffic dataset, set WRITE IPC threshold, set time T
Output: WRITE commands towards IPC is above threshold
if Total_WRITE_commands > set threshold then
    return WRITE commands towards IPC is above the normal threshold;
else
end
```

Algorithm 5: Prevalence of Services Started or Created

```
1 event dce_rpc_request ();
2 Total Services Started or Created in SMB1 SMB2 = Total_Services;
  Input  : Traffic dataset, set services threshold, set time T
  Output: Services Started or Created is above normal threshold
3 if Total_Services > set threshold then
4 |   return Services Started or Created is above the normal threshold;
5 else
6 end
```

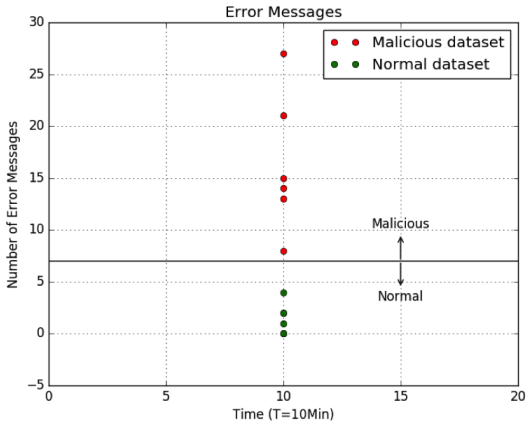

Figure 5.5.1.1: Training of K-nearest neighbour algorithm with datasets from Home lab. X-axis indicates Time (T=10Min) while Y-axis indicates number of Error messages in different datasets.

if the session key is not established, this policy triggers anomalous behaviour incident. If a session key is established no incident is triggered. Random hostname policy, also monitors NTLM authentication. It calculates the entropy of each hostname seen during the authentication procedure. A 3.5 entropy threshold is fixed in this policy. So it compares each calculated entropy with threshold of 3.5. A hostname having entropy above 3.5 is reported as anomalous. No incident is triggered for hostnames having entropy below 3.5.

Error messages policy, counts the number of error messages in SMB1 and SMB2 protocol in each dataset. The number of error messages are extracted from a set of normal and malicious datasets generated in Home lab. These normal and malicious datasets ought to be of the same network. To have sense of real time monitoring, we are considering the number of error messages during each

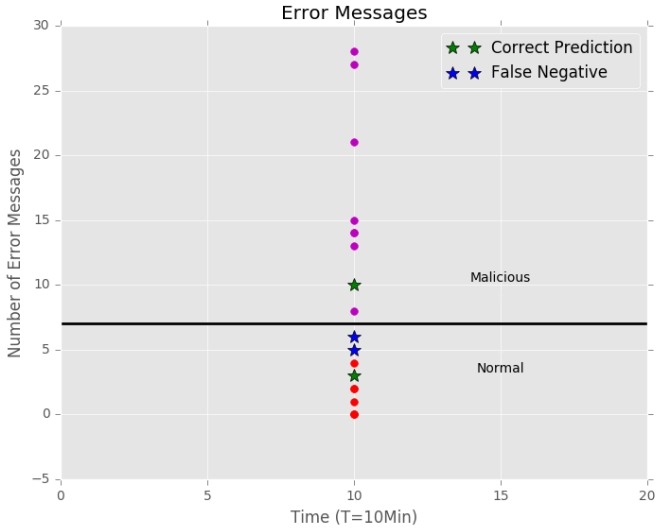

Figure 5.5.1.2: Testing of machine learning algorithm with datasets from Home lab that were not used in the training set.

time T=10 minutes both in normal and malicious datasets. Time T=10 minutes is a good estimate because mainly the attack needed 10 minutes to complete and also for time T=10 minutes it is becomes more precise to predict normal and malicious behaviour. For time less than 10 minutes the prediction of behaviour becomes ambiguous. Also too much short time generates unnecessary large number of false alarms, and the attack won't be detected. While long time duration is ineffective because in this way the number of error messages, number of Write commands towards IPC$ share, services started, created be distributed equally and might simulate real behaviour. This time is tested on datasets from different networks and is equally suitable for all networks. So the number of error messages for each time T=10 minutes in each dataset (normal and malicious) are used to train K-nearest neighbour algorithm. Figure 5.5.1.1 shows training of K-nearest algorithm with both normal and malicious datasets and a threshold is learned. These are the datasets we generated in our Home lab. This threshold is fixed in Error messages policy. If in any dataset the number of error messages in time T=10 minutes are above the learned threshold the dataset is flagged as anomalous. Figure 5.5.1.2 shows the testing phase of machine learning algorithm with datasets from Home lab. The algorithm is tested with the datasets of time T=10 minutes. These testing instances of datasets were not used in the training set.

Write command towards IPC$ policy, counts the number of Write commands IPC$ in SMB1 and SMB2 protocol in each dataset. The number of Write commands towards IPC$ are extracted from a set of normal and malicious datasets. These normal and malicious datasets ought to be of the same network. For the reasons mentioned earlier here the time is also taken as T= 10 minutes. So the

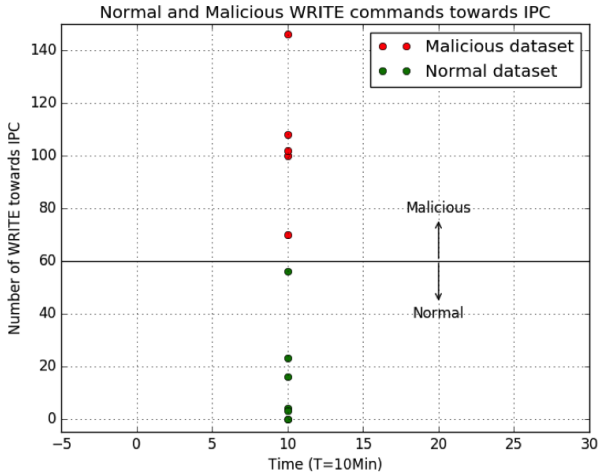

Figure 5.5.1.3: Training of K-nearest neighbour algorithm with datasets from Home lab. X-axis indicates Time (T=10Min) , Y-axis indicates number of Number of WRITE commands towards IPC$ in different datasets.

number of Write command towards IPC$ for each time T=10 minutes in each dataset (normal and malicious) are used to train K-nearest neighbour algorithm.

Figure 5.5.1.3 shows training of K-nearest algorithm with both normal and malicious datasets from the Home lab. These datasets we generated in our Home lab and a threshold is learned. This threshold is fixed in Write command towards IPC$ policy. If in any dataset the number of Write command towards IPC$ in time T=10 minutes are above the learned threshold the dataset is flagged as anomalous. Figure 5.5.1.4 shows the testing phase of machine learning algorithm with datasets from the Home lab. The algorithm is tested with the datasets of time T=10 minutes. These testing instances of datasets were not used in the training set.

Services Started, Created policy, counts the number of services started, created in DCE-RPC protocol in each dataset. The number of Services Started, Created are extracted from a set of normal and malicious datasets. These normal and malicious datasets ought to be of the same network. For the reasons mentioned earlier here the time is also taken as T= 10 minutes. So the number of Services Started, Created for time T=10 minutes in each dataset (normal and malicious) are used to train K-nearest neighbour algorithm. Figure 5.5.1.5 shows training of K-nearest algorithm with both normal and malicious datasets from Home lab. These datasets we generated in our Home lab and a threshold is learned. This threshold is fixed in Services Started, Created policy. If in any dataset the number of Services Started, Created in time T=10 minutes are above the learned threshold the dataset is flagged as anomalous. Figure 5.5.1.6 shows the testing phase of machine learning algorithm with datasets from Home lab. The algorithm is tested with the datasets of time T=10 minutes.

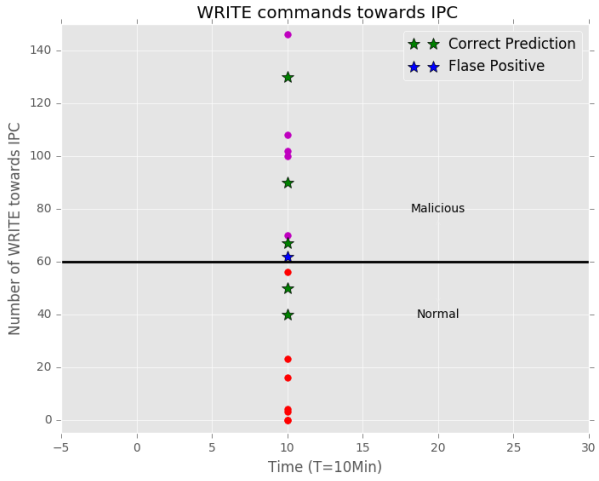

Figure 5.5.1.4: Testing of machine learning algorithm with datasets from Home lab that were not used in the training set.

These testing instances of datasets were not used in the training set.

5.6 Model Evaluation

This sections studies the performance of our detection model on different datasets. The evaluation is performed on the three datasets: the datasets we acquired from Home lab setup, advanced forensics challenge [98] and datasets from a real corporate network. Each of these datasets are large traffic, that contains both the normal behaviour and the attack. As our model works in batches so we divided all these large traffics into small 10 minutes datasets. To further clarify, for instance advanced forensics challenge [98] dataset time duration is two days. This dataset contains both normal behaviour and attack traffic. This large dataset is divided into 10 minutes small datasets and an alarm is triggered if the algorithm predicts there is an attack ongoing (in these 10 minutes). We analysed the performance of each policy individually. For empty session key policy, the prediction is 100% precise as shown in Table 5.4 because this characteristic is only present in malicious datasets. Hostname randomness policy raises false-positives when normal hostnames are complex and have high entropy. For example `WKS-WINXP32BIT` hostname is triggered as malicious although it legal hostname. The performance of the remaining three ploicies is directly proportional to the datasets used to train K-Nearest algorithm. If the training datasets are extensive, the prediction precision of the machine learning algorithm is better and ultimately very few the number of false positive and false negative.

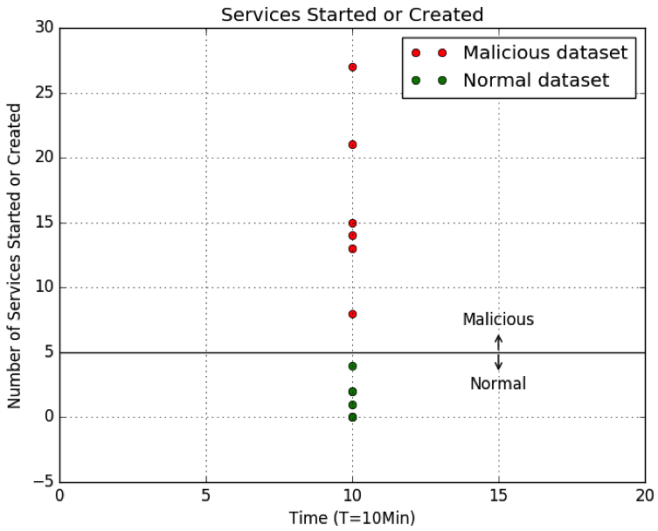

Figure 5.5.1.5: Training of K-nearest neighbour algorithm with datasets from Home lab. X-axis indicates Time (T=10Min) while Y-axis indicates number of Services Started, Created in different datasets.

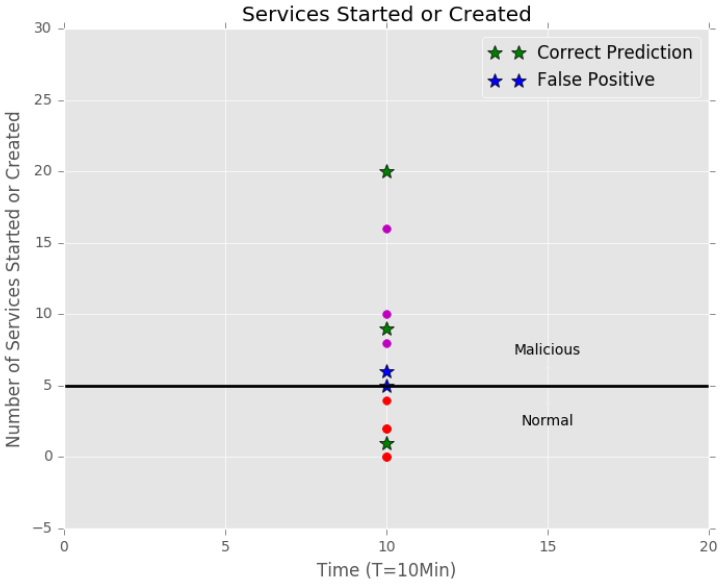

Figure 5.5.1.6: Testing of machine learning algorithm with Home lab datasets that were not used in the training set.

5.6.1 Datasets from lab setup at Home

This section describes the evaluation performed on the datasets we get from lab setup at Home. These datasets are described in Section 5.4. We have tested our model with three different types of model validation techniques. Table 5.4 shows one type of the validation in which the model is trained with both normal and malicious datasets and then test the model with datasets that were not used in training set. The model behaviour depends on the number of datasets used for the training and the distribution of the datasets used for the training (number of normal datasets used in training and number of malicious datasets used in training). That is why we get mostly false alarm on the test datasets that are very near to the threshold value and the algorithm decides for the class that has more datasets near the threshold. Table 5.2 shows second type of model validation technique called cross validation. Cross validation is explain in Section 3.3.1. Few number of false alarms are generated it is because the test datasets' values comes in the range where the opposite class has more values near to it and thus the algorithm decides the test case for the wrong class. Table 5.3 shows the third type of model validation technique called percentage split which is discussed in Section 3.3.1. The reason why there are no false negatives is because all the test datasets have values which are either far bigger than the threshold or very smaller than the threshold thus the algorithm decides for the correct class. For Home lab datasets, in 13 hours of network traffic we generate 7 false alarms. Thus we get 1 false alarm in 2 hours which is considerably good performance. Empty session key and random hostname techniques do not need training, so that is why the number of training datasets in Table 5.4 for these techniques are zero.

5.6.2 Datasets from Corporate Network

In these datasets, among normal datasets the number of Write commands towards IPC$ is zero, the number of Services Started and Created is zero, while the number of error messages in some datasets are zero except in one dataset which is really high although the dataset is normal. This huge number of errors messages is because a legit user is unintentionally violating access control policies. As in this case the normal datasets has always zero number of Write commands towards ipc and service, except in one case the number of error messages are also zero. So we have always only one normal test dataset in which the number of Write towards IPC$ is zero, number of error messages is zero, and number of services Started, Created is zero. The number of true negatives and true positives depends on the test datasets. If there are few true negative it is because the normal test dataset is very small as in this case in which we have only one normal dataset. Table 5.6 shows a technique of model validation, in which the model is trained with both normal and malicious datasets and then test the model with datasets that were not used in training set. The model behaviour depends on number of datasets used for the training and the distribution of the datasets used for

Detection Technique	Number of datasets	Cross Validation (folds)	Accuracy (%)	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
WRITE commands towards IPC$	12	3	100	0	0	5	7
WRITE commands towards IPC$	12	5	83.3	1	1	4	6
WRITE commands towards IPC$	12	7	83.3	1	1	4	6
Error Messages	13	3	92.3	0	1	5	7
Error Messages	13	5	100	0	0	6	7
Error Messages	13	7	92.3	0	1	5	7
Service Started Created	11	3	90.90	0	1	3	7
Service Started Created	11	5	100	0	0	4	7
Service Started Created	11	7	100	0	0	4	7

Table 5.2: Cross Validation Method for Home lab datasets

Detection Technique	Number of datasets	Percentage Split (%training, %test)	Accuracy (%)	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
WRITE commands towards IPC$	12	80 , 20	100	0	0	4	6
WRITE commands towards IPC$	12	70 , 30	87.5	1	0	2	5
WRITE commands towards IPC$	12	60 , 40	87.5	1	0	2	4
Error Messages	13	80 , 20	90	1	0	4	5
Error Messages	13	70 , 30	88.88	1	0	3	5
Error Messages	13	60 , 40	100	0	0	3	5
Service Started Created	11	80 , 20	88.88	1	0	3	5
Service Started Created	11	70 , 30	100	0	0	3	5
Service Started Created	11	60 , 40	100	0	0	2	5

Table 5.3: Percentage Split Method for Home lab datasets

Detection Technique	Number of training datasets	Number of test datasets	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
Empty Session key	0	7	0	0	4	3
Random Hostname	0	10	2	0	6	2
WRITE Commands towards IPC	6	6	1	0	3	2
Error Messages	9	4	0	2	1	1
Services Started Created	6	5	2	0	2	1

Table 5.4: False Positive and False negative generated for Home lab datasets

the training (number of normal datasets used in training and number of malicious datasets used in training). Malicious dataset of corporate network contains one attack. As we have divided the dataset in smaller pieces of 10 minutes, some small datasets do not contain any information about the attack despite being labelled as an attack. Since we have evaluated this in this way, we expect to have false-negatives. Also the high number of false negative in table 5.5 and in 5.7 is for the same reason. Our model does not generate a lot of false positives and the one that is generated is assigned to a user misbehaviour which is indeed an anomaly. So the attack and as well as the user misbehaviour is successfully detected. For corporate network datasets, we get 7 false alarms in 9 hours network traffic. Which is approximately 1 false alarm in 1 hour thus the performance of our model for corporate network datasets is good. Empty session key and random hostname techniques do not need training, so that is why the number of training datasets in Table 5.6 for these techniques are zero.

5.6.3 Dataset from advanced forensics challenge [98]

In these datasets, the number of Write commands towards IPC$, the number of error messages, and the number of services Started, Creates in normal datasets are equal to zero. So all the normal datasets have value equal to zero, that's why the number of true negatives are very few. While in the malicious datasets these numbers are not zero. As in these datasets, there is huge difference between the normal and malicious datasets (normal datasets have values zero, malicious datasets have high value) and the test datasets either have zero or very high values so the model correctly classifies the test datasets. The model generates almost no false alarm as shown in table 5.8 and table 5.9. The false positives in table 5.10 is because the way datasets are divided for training and testing by percentage split method. As in this case we have very small set of normal datasets so not using these datasets for training generates false positives. For advanced forensics challenge datasets, we get 1 false alarm in almost 48 hours of network traffic. In this case, our model performance is very good. Empty session key and random hostname techniques do not need training, so that is why the number of training datasets in Table 5.8 for these techniques are zero.

Detection Technique	Number of datasets	Cross Validation (folds)	Accuracy (%)	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
WRITE commands towards IPC$	10	3	90	0	1	6	3
WRITE commands towards IPC$	10	5	90	0	1	6	3
WRITE commands towards IPC$	10	7	90	0	1	6	3
Error Messages	16	3	68.7	3	2	11	0
Error Messages	16	5	75	3	1	12	0
Error Messages	16	7	68.7	3	2	11	0
Service Started Created	7	3	85.7	0	1	2	4
Service Started Created	7	5	85.7	0	1	2	4
Service Started Created	7	7	85.7	0	1	2	4

Table 5.5: Cross validation method for corporate network datasets

Detection Technique	Number of training datasets	Number of test datasets	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
Empty Session key	0	3	0	0	2	1
Random Hostname	0	3	1	0	2	0
WRITE Commands towards IPC	6	4	0	1	2	1
Error Messages	8	8	0	3	4	1
Services Started Created	4	3	0	2	0	1

Table 5.6: False positive and False negative generated for corporate network datasets

Detection Technique	Number of datasets	Percentage Split (%training, %test)	Accuracy (%)	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
WRITE commands towards IPC$	10	80 , 20	87.5	0	1	5	2
WRITE commands towards IPC$	10	70 , 30	83.3	0	1	5	0
WRITE commands towards IPC$	10	60 , 40	83.3	0	1	5	0
Error Messages	16	80 , 20	76.9	3	0	10	0
Error Messages	16	70 , 30	72.7	3	0	8	0
Error Messages	16	60 , 40	70	3	0	7	0
Service Started Created	7	80 , 20	50	0	3	0	3
Service Started Created	7	70 , 30	80	0	1	1	3
Service Started Created	7	60 , 40	75	0	1	1	2

Table 5.7: Percentage Split Method for corporate network datasets

Detection Technique	Number of training datasets	Number of test datasets	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
Empty Session key	0	2	0	0	1	1
Random Hostname	0	2	0	0	1	1
WRITE Commands towards IPC	7	5	0	1	3	1
Error Messages	5	3	0	0	2	1
Services Started Created	7	4	0	0	3	1

Table 5.8: False positive and False negative generated for advanced forensics challenge datasets[98]

Detection Technique	Number of datasets	Cross Validation (folds)	Accuracy (%)	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
WRITE commands towards IPC$	12	3	91.6	0	1	8	3
WRITE commands towards IPC$	12	5	91.6	0	1	8	3
WRITE commands towards IPC$	12	7	91.6	0	1	8	3
Error Messages	8	3	100	0	0	5	3
Error Messages	8	5	100	0	0	5	3
Error Messages	8	7	100	0	0	5	3
Service Started Created	11	3	100	0	0	8	3
Service Started Created	11	5	100	0	0	8	3
Service Started Created	11	7	100	0	0	8	3

Table 5.9: Cross validation method for advanced forensics challenge datasets[98]

Detection Technique	Number of datasets	Percentage Split (%training,%test)	Accuracy (%)	Number of False Positive	Number of False Negative	Number of True Positive	Number of True Negative
WRITE commands towards IPC$	12	80 , 20	70	3	0	7	0
WRITE commands towards IPC$	12	70 , 30	62.5	3	0	5	0
WRITE commands towards IPC$	12	60 , 40	85.7	0	1	4	2
Error Messages	8	80 , 20	100	0	0	4	2
Error Messages	8	70 , 30	100	0	0	4	2
Error Messages	8	60 , 40	100	0	0	4	1
Service Started Created	11	80 , 20	66.6	3	0	6	0
Service Started Created	11	70 , 30	100	0	0	6	2
Service Started Created	11	60 , 40	100	0	0	5	2

Table 5.10: Percentage Split Method for advanced forensics challenge datasets[98]

5.6.4 Types of attacks we can detect with our model

Following lateral movement techniques can be detected with our model.

(i) **Application deployment Software**

Intruders perhaps try to install malicious software to systems within the network utilizing application deployment systems which is employed by admin. In most of the cases the system administrators use specific domain credentials for application deployment systems which is different from the Administrator credentials. If the attackers mistakenly enters administrator credentials, our Error detection policy will trigger an alarm.

(ii) **Pass the hash attack**

This method bypasses standard authentication procedure that require a clear text password, instead uses hashes for authentication. Also Pass the hash allows anonymous logins. Our approach detects pass the hash with empty session key, and random hostname.

(iii) **Brute force attack**

In case the attacker doesnt have password of certain machine, he may try brute forcing the password. By analyzing the password brute forcing traffic, it generates a huge amount of traffic towards IPC share. Also with the Error Messages generated (Access Denied).

(iv) **Remote File Copy**

Attacker might transfer files to the victim computer. Such an act requires admin privileges and generate large amount of Write commands towards IPC$ share. And this malicious behaviour can be detected with our approach.

(v) **Remote services**

With Services started, created policy we can detect installation of malicious services.

(vi) **Windows Admin Share**

Windows systems have hidden network shares that are accessible only to administrators and provide the ability for remote file copy and other administrative functions. Example network

shares include C$, ADMIN$, and IPC$. Adversaries may use this technique in conjunction with administrator-level legitimate credentials to remotely access a networked system over server message block (SMB) to interact with systems using remote procedure calls (RPCs), transfer files, and run transferred binaries through remote execution. Example execution techniques that rely on authenticated sessions over SMB/RPC are Scheduled Task, Service Execution, and Windows Management Instrumentation. Adversaries can also use NTLM hashes to access administrator shares on systems with Pass the Hash and certain configuration and patch levels [68]. With our model exploitation of Windows Admin Share can be detected with Write commands towards IPC$ ploicy.

(vii) Exploitation of access control policies Any failed attempt to exploit access control policies generates error messages. So our model is capable of detecting such exploitation.

5.6.5 Why our approach works

Whenever an attacker want to exploit remote machines, he tries to remain anonymous, undetected, transfer and execute malicious code, create and start services, tries to bypass normal authentication procedures, guess wrong passwords and hashes until he gets the correct credentials. All of these behaviors enhance the possibility of malicious activities in the network. Even very smart attacker might find it hard to bypass our detection model. The reason is that our model is a multivariant which detects all these mentioned malicious activities.

Chapter 6

Conclusion and future work

6.1 Conclusion

The overriding purpose of this study is to research and develop an innovative approach to detect lateral movement attack and implement it in BRO network analyser. Detecting lateral movement is of high significance because such an attack can cause economical losses and breach privacy at larger scale. To accomplish this goal we developed a model that comprise of five BRO policies. The

model is collectively detecting five different types of anomalies related to lateral movement attack. Which makes our model one of the comprehensive detection approach compare to existing detection approaches. The model is trained and tested on three different datasets containing lateral movement attack which are from three different sources and the model is capable to detect the lateral movement attack in all these datasets. The model is equally proficient in detecting lateral movement attack performed through both SMB1 and SMB2. The main limitation of our approach is that in order to achieve better detection, the model has to be trained on large number of normal and malicious datasets and acquiring complete datasets is always fractious. So the performance of our model can be improved with large training datasets although we have analysed that the model detection performance is adequate in situation where the training datasets are very limited. Our main scientific contribution is providing unexplored detection mechanisms for sophisticated lateral movement attack. To conclude, with our model it is practically possible to detect lateral movement attack and can be easily implemented in any environment.

6.2 Future work

Some interesting further areas to be explored is to discovered the detection of lateral movement attack through SMB3. And to research the detection of lateral movement by extraction of RPC calls.

References

[1] Mark Russinovich. (May 2, 2014). PsExec v2.11, Available: https://technet.microsoft.com/en-us/sysinternals/bb897553.aspx Ding, W. and Marchionini, G. 1997. A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] Jose Barreto. Microsoft SMB remote file protocol. Available: http://www.snia.org/sites/default /education/tutorials/2012/fal l/file/JoseBarreto_SMB3_Remote_File_Protocol_revision.pd f.

[3] Richard Sharpe. (October 8, 2002). Just what is SMB? Available: https://www.samba.org/cifs/docs/ what-is- smb.html

[4] Microsoft SMB Protocol and CIFS Protocol Overview. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365233(v=vs.85).aspx

[5] What is BRO? (n.d). Available at: https://web.nsrc.org/workshops/2015/ pacnog17- ws/raw-attachment/wiki/Track2Agenda/bro-intro.htm

[6] Arun Hodigere (n.d). Intrusion Detection. Available: www.cs.utexas.edu/users/ygz/395T-01F /reading/arun.ppt

[7] Reusing credentials (n.d). Available: https://help.rapid7.com/metasploit/Content/credentials/ reusin g-credentials.html

[8] Testing Lateral movement effectiveness with the metasploit PRO credential Domino Metamodule. (n.d). Available: https://www.rapid7.com/resources/videos/credential-domino- metamodule-in-metasploit-pro.jsp

[9] Searching for credentials (n.d). Available: https://help.rapid7.com/metasploit/Content/credentials/ searching-credentials.html

[10] Credential Tutorial (n.d).. Available: https://help.rapid7.com/metasploit/Content/tutorials/working-with-credentials.html Task7

[11] Credentials Domino MetaModule (n.d). Available: https://help.rapid7.com/metasploit/Content/ metamodules/cre dentials-domino-metamodule.html

[12] Managing Credentials (n.d). Available: https://help.rapid7.com/metasploit/Content /credentials/mana ging-credentials.html

[13] W. Ballenthin and M. Graeber (n.d). Windows management Instrumentation Offense, Defense and Forensics. Available: https://www.fireeye.com/content/dam/fireeye- www/global/en/current-threats/pdfs /wp-windows- management-instrumentation.pdf

[14] Florian May 12 2015. An Introduction to Pass the Hash. Available: https://bogner.sh/2015/05/an-introduction-to-pass- the-hash/

[15] Local Security Authority Subsystem Service (n.d). Available https://en.wikipedia.org/wiki/Local_Security_

Authority_Sub system_Service

[16] David Lladro. (November 4, 2013). Plaintext passwords with Procdump and Mimikatz Alpha. Available: http://www.securityartwork.es/2013/11/04/plaintext- passwords-with-procdump-and-mimikatz-alpha/

[17] David Maloney. (December 9, 2015). Dumping active directory password hashes explained. Available: https://www.rapid7.com/resources/videos/dumping-active- directory-password-hashes.jsp?utm_medium=blog- cta&utm_campaign=blog-cta&CS=bouncex

[18] Thu Pham. (February 26, 2015). Inside Retail Hack: Lateral Movement & Credential Harvesting. Available: https://duo.com/blog/inside-a-retail-hack-lateral-movement- and-credential-harvesting

[19] Alfred Lee. (September 11, 2009). Microsoft SMB2 Vulnerability. Available: http://researchcenter.paloaltonetworks.com/2009/09/microsof t-smb2-vulnerability/

[20] CVE-2009-2532. (October 14, 2009). Available: https://web.nvd.nist.gov/view/vuln/search- results?adv_search=true&cves=on&cpe_version=cpe%3A% 2Fo%3Amicrosoft%3Awindows_vista%3A-%3Asp2&startIndex=200

[21] Microsoft Windows Server Service RPC handling remote code execution vulnerability. (October 22, 2008). Available: https://www.symantec.com/security_response/vulnerability.j sp?bid=31874

[22] CVE-2012-4774. (June 9, 2012). Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012- 4774

[23] Redirect to SMB. (n.d). Available: https://cdn2.hubspot.net/hubfs/270968/SPEAR/RedirectToSMB_public_whitepaper.pdf?t=1458691799493

[24] Jose Pagliery. (December 29, 2014). What caused Sony hack: What we know now. Available: http://money.cnn.com/2014/12/24/technology/security/sony- hack-facts/

[25] Malware Identified from Attack on Sony. (January 7, 2015). Available: https://securityfyi.wordpress.com/tag/us-cert/

[26] Tim Hornyak. (Febraury 4, 2015). Hack to cost Sony $35 million in IT repairs. Available:

http://www.networkworld.com/article/2879814/data- center/sony-hack-cost-15-million-but-earnings - unaffected.html

[27] Vern Paxson, BRO A system for detecting network intruders in real time, in: Proceedings of the 7th USENIX Security Symposium San Antonio, Texas, January 26-29, 1998

[28] Matt Graeber. (2015). Abusing Windows Management Instrumentation to build a persistent Asynchronous, and fileless backdoor. Available: https://www.blackhat.com/docs/us-15/materials/us-15- Graeber-Abusing-Windows-Management-Instrumentation- WMI-To-Build-A-Persistent%20Asynchronous-And- Fileless-Backdoor-wp.pdf & https://attivonetworks.com/documentation/Attivo_Networks-Lateral_Movement.pdf

[29] Christian Callegrari. (May 9, 2009). Statistical Approaches for Network Anomaly Detection. Available: http://www.iaria.org/conferences2010/filesICIMP10/ICIMP_ Tutorial_Christian_Callegari.pdf

[30] Yu Gu, Andrew McCallum, Don Towsley. Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation. Department of Computer science, University of Massachusetts, Amherst, MA 01003, 2005.

[31] Animesh Patcha, and Jung-Min Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, United States, 2007, PP 34483470.

[32] S.E Smaha, Haystack: An intrusion detection system, in: Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, 1988, PP 37-44.

[33] D.E. Denning, P.G. Neumann, Requirements and Model for IDES A Real-time Intrusion Detection System, Computer Science Laboratory, SRI International, Menlo Park, CA 94025-3493, Technical Report 83F83-01-00, 1985

[34] T.F. Lunt, A. Tamaru, F. Gilham, R. Jagannathm, C. Jalali, P.G. Neumann, H.S. Javitz, A. Valdes, T.D. Garvey, A Real- time Intrusion Detection Expert System (IDES), Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Final Technical Report, February 1992.

[35] D. Anderson, T. Frivold, A. Tamaru, A. Valdes, Next- generation intrusion detection expert system

(NIDES), Software Users Manual, Beta-Update release, Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-0, May 1994.

[36] D. Anderson, T.F. Lunt, H. Javitz, A. Tamaru, A. Valdes, Detecting Unusual Program Behavior Using the Statistical Component of the Next-generation Intrusion Detection Expert System (NIDES), Computer Science Laboratory, SRI International, Menlo Park, CA, USA SRI-CSL-95-06, May 1995.

[37] S.A. Hofmeyr, S. Forrest, A. Somayaji, Intrusion detection using sequences of system calls, Dept. of Computer Science University of New Mexico Albuquerque, NM 87131-1386.

[38] J.E. Dickerson, J.A. Dickerson, Fuzzy network profiling for intrusion detection, in: Proceedings of the 19th Inter- national Conference of the North American Fuzzy Information Processing Society (NAFIPS), Atlanta, GA, 2000

[39] D. Barbara , J. Couto, S. Jajodia, N. Wu, ADAM: a testbed for exploring the use of data mining in intrusion detection, ACM SIGMOD Record: SPECIAL ISSUE: Special section on data mining for intrusion detection and threat analysis 30 (2001)

[40] Mohammad M. Masud, Latifur Khan, Bhavani Thuraisingham, Xinran Wang, Peng Liu and Sencun Zhu, A Data mining technique to detect remote exploit. Available at: https://pdfs.semanticscholar.org/7bbe/792a233b513ad84ff552a41430471f9e8bb2.pdf

[41] Matthew V. Mahoney and Philip K. Chan, PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic Department of Computer Sciences Florida Institute of Technology Melbourne, FL 32901, Florida Institute of Technology Technical Report CS-2001-04

[42] Ke Wang and Salvatore J. Stolfo, Anomalous Payload-Based Network Intrusion Detection, Computer Science Department, Columbia University 500 West 120th Street, New York, NY, 10027

[43] Stephanie Forrest Steven A. Hofmeyr Anil Somayaji, Thomas A. Longstaff , A Sense of Self for Unix Processes, Dept. Of Computer Science University of New Mexico Albuquerque NM 87131-1386, CERT Coordination Center Software Engineering Institute Carnegie-Mellon University.

[44] A Fast Automaton-Based Method for Detecting Anomalous Program Behaviors by R. Sekar M. Bendre D. Dhurjati State University of New York Stony Brook NY 11794 P. Bollineni Lowa State University Ames, IA 50014

[45] Forrest, S., Warrender, C., and Pearlmutter, B, Detecting intrusions using system calls: Alternate data models. In Proceedings of the 1999 IEEE ISRSP. IEEE Computer Society, Washington, DC, USA, 1999.

[46] Matti Mantere, Mirko Sailio and Sami Noponen, Network Traffic Features for Anomaly Detection in Specific Industrial Control System Network, VTT Technical Research Centre of Finland, Kaitovayla 1, Oulu 90571, Finland.

[47] Inter-process communication. 26 March 2016. Available: https://en.wikipedia.org/wiki/Inter-process_communication

[48] Remote Desktop protocol. 11 May 2016. Available: https://en.wikipedia.org/wiki/Remote_Desktop_Protocol

[49] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, Alfredo De Santis, Network Anomaly Detection with the restricted Boltzmann machine, Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Neurocomputing 122 (2013) 1323.

[50] Harinee.k, Veeramuthu.A Sivagangai, Tamilnadu, Intrusion Detection System Based on Fuzzy Association Rule with Genetic Network Programming, Dep of Information Technology, K.L.N College of Information Technology, India Dep of Information Technology, Sathyabama University, Chennai,Tamilnadu, India.

[51] Nong Ye and Qiang Chen, : An Anomaly Detection Technique Base on a Chi-Square statistic for Detecting intrusions into information systems, Department of Industrial Engineering, Arizona State University, Tempe, AZ 85287- 5906, USA.

[52] Emmanuele Zambon, Damiano Bolzoni, Sandro Etalle, Pieter Hartel, POSEIDON: a 2-tier Anomaly-based Network Intrusion Detection System University of Twente, Distributed and Embedded System Group, P.O. Box 2100, 7500 AE Enschede, The Netherlands, Universita CaFoscari di Venezia, Dipartimento di Informatica, Via Torino 155, 30172 Mestre (VE), Italy.

[53] A Chi-square testing-based intrusion detection Model. Nasser S. Abouzakhar and Abu Bakar School of Computer Science, The University of Hertfordshire, College Lane, Hatfield AL10 9AB, Hertfordshire, UK N.Abouzakhar, A.Bakar@herts.ac.uk

[54] Jiong Zhang and Mohammad Zulkernine, : Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection, School of computing Queens University, Kingston Ontario, Canada K7L 3N6.

[55] Marin J, Ragsdale D, Surdu J : A Hybrid Approach to the Profile Creation and Intrusion Detection. Proceedings of the DARPA Information Survivability Conference and Exposition DISCEX 2001, June 2001.

[56] Philip K. Chan, Matthew V. Mahoney, and Muhammad H. Arshad. Learning Non-stationary Models of Normal Network Traffic for Detecting Novel Attacks, Department of Computer Sciences Florida Institute of Technology, Melbourne, FL 32901, 2002.

[57] Bailin Xie, Qiansheng Zhang. Application-layer Anomaly Detection Based on Application-layer Protocols Keywords. Cisco School of Informatics Guangdong University of Foreign Studies Guangzhou, China. International conference on computer science and network technology Changchun, China 2012.

[58] Christopher Krugel, Thomas Toth, Engin Kirda. Service Specific Anomaly Detection for Network Intrusion Detection Distributed Systems Group Technical University Vienna A- 1040 Vienna, Austria. 2012.

[59] Damiano Bolzoni, Sandro Etalle. Approaches in anomaly- based intrusion detection systems. University of Twente, P.O. Box 2100, 7500 AE Enschede, The Netherlands.

[60] Pavel Laskov. Intrusion detection and malware analysis. Signature based IDS. Wilhelm Schickard institute for computer science. Avaialbe at: http://www.ra.cs.uni- tuebingen.de/lehre/ws10/ids-malware /05-signature-ids.pdf

[61] Pavel Laskov. Intrusion detection and malware analysis. Signature based IDS. Wilhelm Schickard institute for computer science. Avaialbe at: http://www.ra.cs.uni- tuebingen.de/lehre/ws10/ids-malware/05-signature-ids.pdf

[62] Anomaly detection. 30 April 2016. Available at: https://en.wikipedia.org/wiki/Anomaly_detection

[63] Arnt Brox. 2002. Signature based or anomaly based Intrusion detection: The Practice and Pit-

falls. Available at: http://www.scmagazine.com/signature-based-or-anomaly- based-intrusion-detection-the-practice-and- pitfalls/article/30471/

[64] Sony pictures entertainment hack. May 2016. Available at: https://en.wikipedia.org/wiki/Sony_Pictures_Entertainment_ hack.

[65] Network Security: Restrict NTLM: NTLM authentication in this domain. November 15,2012. Available at: https://technet.microsoft.com/en-us/library/jj852241(v=ws.10).aspx

[66] Pros and Cons of Disabling NTLMv1. 19 August 2008 Available at: http://www.windowsnetworking.com/kbase/WindowsTips/Windows2003/AdminTips/Security/ProsandConsofDisablingNTLMv1.html

[67] PSExec Demystified . 9 May 2013. Available at: https://community.rapid7.com/community/metasploit/blog/2013/03/09/psexec-demystified

[68] Lateral movement. 27 June 2016. Available at: https://attack.mitre.org/wiki/Lateral_Movement

[69] IPC Share Exploit: January 20,2003. Methodology of Chinese Attackers by Bernard Kan. Option 2 : Cyber Defense Initiative. Version 2.1a

[70] Protecting windows Networks Defeating pass-the-hash. 8 November 2015. Available at: https://dfir-blog.com/2015/11/08/protecting-windows-networks-defeating-pass-the-hash/

[71] Detecting Lateral Movement From Pass the Hash Attacks. By Chris Martin. 2 Feb 2015. Available at: https://logrhythm.com/blog/detecting-lateral-movement-from-pass-the-hash-attacks/

[72] Detection of Phishing Attacks: A Machine Learning Approach Ram Basnet, Srinivas Mukkamala, and Andrew H. Sung New Mexico Tech, New Mexico 87801, USA ram,srinivas,sung@cs.nmt.edu

[73] High-Performance Content-Based Phishing Attack Detection Brad Wardman, Tommy Stallings, Gary Warner, Anthony Skjellum Computer Forensics and Research Laboratory, University of Alabama at Birmingham bwardman, tds2, gar, tony@uab.edu

[74] A Framework for Detection and Measurement of Phishing Attacks Sujata Garera Johns Hopkins University Baltimore, MD 21218 sgarera@cs.jhu.edu Niels Provos Google Inc. Mountain View CA 94043 niels@google.com Monica Chew Google Inc. Mountain View CA 94043 mmc@google.com

Aviel D. Rubin Johns Hopkins University Baltimore, MD 21218 rubin@jhu.edu

[75] Rule-Based Phishing Attack Detection Ram B. Basnet a,b,* , Andrew H. Sung a,b, Quingzhong Liu c Computer Science & Engineering Department, New Mexico Tech, Socorro, NM 87801, USA Institute for Complex Additive Systems Analysis (ICASA), New Mexico Tech, Socorro, NM 87801, USA Department of Computer Science, Sam Houston State University, Huntsville, TX 77341, USA *Corresponding Author, rbasnet, sung@cs.nmt.edu, qxl005@shsu.edu

[76]Pass-The-Hash Toolkit for Windows Implementation & use. By Hernan Ochoa. 29 October 2008. Available at: https://www.coresecurity.com/system/files/publications/2016/05/Ochoa_2008-Pass-The-Hash.pdf

[77] NTLM. (n.d). Available at: http://www.opengroup.org/comsource/techref2/NCH1222X.HTM

[78] XNTLM. (n.d). Available at: https://developer.gnome.org/evolution-exchange/stable/evolution-exchange-xntlm.html#xntlm-authenticate

[79] The NTLM Authentication Protocol and Security Support Provider. Available at: http://davenport.sourceforge.net/ntlm.html#theNullUserSessionKey

[80] Glossary. By Microsoft. Available at: https://msdn.microsoft.com/en-us/library/cc246484.aspx

[81] Glossary. By Microsoft. Available at: https://msdn.microsoft.com/en-us/library/cc233529.aspx#gt_4f67a585-fb00-4166-93e8-cf4abca8226d

[82] NEGOTIATE. By Microsoft. Available at: https://msdn.microsoft.com/en-us/library/cc236650.aspx

[83] Verifying the Signature. Available at: https://msdn.microsoft.com/en-us/library/cc246762.aspx

[84] Sony Hack Attack: Cybersecurity Expert Reveals How Massive Breach Might Have Happened. By Travis Reilly. 4 December 2014 http://www.thewrap.com/sony-hack-attack-cybersecurity-expert-reveals-how-massive-breach-might-have-happened/

[85] Target's Data Breach: The Commercialization of APT. By Tal Be'ery. 19 March 2014. Available at: http://www.securityweek.com/targets-data-breach-commercialization-apt

[86] JPMorgan Chase Breach Puts Renewed Focus on Malware Attacks at Large Organizations. By Dana Tamir. 8 October 2014. Available at: https://securityintelligence.com/jpmorgan-chase-breach-puts-renewed-focus-on-malware-attacks-at-large-organizations/

[87] Increased Activity Targeting Windows Shares. March 11, 2003. https://www.cert.org/historical/advisories/CA-2003-08.cfm?

[88] Basic BRO -R runtime examples and overview of important paths and files. 21 August. Available at https://www.syncurity.net/2013/08/21/broversity-lesson-2/

[89] Machine Learning in R for beginners. 25 March 2015. Available at: https://www.datacamp.com/community/tutorials/machine-learning-in-r#gs.dxuCegU

[90] k-nearest neighbors algorithm. 29 October 2016. Available at: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[91] M. Balduzzi, V. Ciangaglini, and R. McArdle, Targeted attacks detection with spunge, 2013.

[92] B. Bencs ath, G. P ek, L. Butty an, and M. F elegyh azi, Duqu: Analysis, detection, and lessons learned, in ACM European Workshop on System Security (EuroSec), vol. 2012, 2012.

[93] ROMAN JASEK, MARTIN KOLARIK, TOMAS VYMOLA. "APT detection system using honeypots". The Faculty of Applied Informatics Tomas Bata University in Zlin Nad Stranemi 4511, 760 05 Zlin CZECH REPUBLIC jasek@fai.utb.cz; martin.kolarik@email.cz; vymola@gmail.com

[94] "Preventing Pass-the-Hash and Similar Impersonation Attacks in Enterprise Infrastructures" Alexander Oberle, Pedro Larbig, Ronald Marx, Frank G. Weber, Dirk Scheuermann Fraunhofer Institute for Secure Information Technology (SIT) Rheinstrae 75, 64295 Darmstadt, Germany, Daniel Fages, Fabien Thomas Arkoon Netasq 1 Place Verrazzano, CS 30603 69258, Lyon Cedex 09, France

[95] "Detecting Malicious Logins in Enterprise Networks Using Visualization" Hossein Siadati New York University Bahador Saket Nasir Memon Georgia Institute of Technology New York University. 2016 IEEE SYMPOSIUM ON VISUALIZATION FOR CYBER SECURITY (VIZSEC)

[96] "Intrusion detection systems and multi-sensors data fusion" Tim Bass. COMMUNICATIONS OF THE ACM April 2000/Vol. 43, No. 4

[97] K-Nearest Neighbors. Accessed on 09-November 2016. Available at: http://www.statsoft.com/textbook/k-nearest-neighbors

[98] Advanced Forensics Challenge FOX-IT academy. Datasets from Machnetico company between 22/08/2014 and 23/08/2014.

[99] PsExec v2.11. By Mark Russinovich. May 2, 2014. Available at: https://technet.microsoft.com/en-us/sysinternals/pxexec.aspx

[100] Impacket. Michael Teo. (n.d). Available at: https://github.com/CoreSecurity/impacket

[101] Microsoft (2016) Windows management instrumentation. Available at: https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx (Accessed: 15 November 2016).

[102] Penetration testing software, pen testing security (no date) Available at: https://www.metasploit.com (Accessed: 15 November 2016).

[103] Microsoft (2016) WMIC - take command-line control over WMI. Available at: https://msdn.microsoft.com/en-us/library/bb742610.aspx (Accessed: 15 November 2016).

[104] Truncer, C. (2016) Veil - framework -. Available at: https://www.veil-framework.com (Accessed: 15 November 2016).

[105] Security, O. (2016) About the Metasploit Meterpreter. Available at: https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/ (Accessed: 15 November 2016).

[106] Wireshark go deep (2016) Available at: https://www.wireshark.org (Accessed: 15 November 2016).

[107] AB, N. (2010) NetworkMiner - the NSM and network forensics analysis tool. Available at: http://www.netresec.com/?page=NetworkMiner (Accessed: 15 November 2016).

[108] Scapy (2005) Available at: http://www.secdev.org/projects/scapy/ (Accessed: 15 November 2016).

[109] The Bro network security monitor. Available at: https://www.bro.org (Accessed: 15 November 2016).

[110] Study highlights Facebook Malware risks: 24% of companies affected - web filtering (2012) Available at: http://www.spamtitan.com/web-filtering/over-24-of-companys-hit-by-facebook-malware-many-unaware-of-malware-riskswe/ (Accessed: 15 November 2016).

[111] Cross Validation. 06 November 2008. By Lei Tang. Available at: http://leitang.net/papers/ency-cross-validation.pdf

[112] Lecture 13: Validation. By Ricardo Gutierrez-Osuna. (n.d). Available at: http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf

[113] Training and testing. By Uros Krcadinac. (n.d). Available at: http://ai.fon.bg.ac.rs/wp-content/uploads/2015/04/Training-and-testing-2015.pdf

[114] Lesson 2.2 Data mining with Weka. (n.d). Available at: http://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/transcripts/Transcript2-2.txt

[115] Credential harvesting. 07-Jan. By Tibor ukina. Available at: https://sgros-students.blogspot.nl/2016/01/credential-harvesting.html

[116] Persistence. 27 June 2016. Available at: https://attack.mitre.org/wiki/Persistence

[117] Netstat. Accessed date 22-November 2016. Available at:https://technet.microsoft.com/en-us/library/bb490947.aspx

[118] Honeypot. Accessed date 22-November 2016 .Available at: https://en.wikipedia.org/wiki/Honeypot_(computing)

[119] Microsoft Windows - NTLM Weak Nonce (MS10-012). 17-October-2010. By Hernan Ochoa. Available at: https://www.exploit-db.com/exploits/15266/