International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 10 Issue 5, May, 2021, Impact Factor: 7.751

A Study of Detection of Structurally Anomalous Logins Within Enterprise Networks

Md. Ashraf Siddiqui

Department of Computer Science, A.M.U. Aligarh

ABSTRACT

To detect lateral motions that target remote vulnerabilities, often network intrusion detection systems use byte sequences. This method, attackers circumvent anti-tampering controls by acquiring legitimate keys and using them to relay data from two separate devices without triggering irregular network traffic. In Credential-based Lateral Movement, we name this method. We use the capability of our technologies to recognise lateral activity of this kind.

CCS Concepts
• Security and privacy → Intrusion/anomaly detection and malware mitigation; Network security; • Computing method-ologies → Anomaly detection;

Keywords: Network Lateral Movement, Anomalous Logins, Pattern Mining

INTRODUCTION

Corporate networks are fully informed of their susceptibility to data transfers and sabotage. The one simple idea communicated by these assaults is that the attacker has decided to make a methodical hacking advancement that finishes in the machine. A individual breaks into and uses keys of someone else to get unauthorized access to the computer or device after which they take off with property or assets. For this reason, an interrupter can begin by using a single computer and by controlling an email connection to the network. The attacker then steals network users' passwords and uses them to connect to other devices. The attacker then switches between devices lat-erally before reaching confidential data contained deeper in the network. We call this type of attack the Lateral Ovation (CLM) Credential (CLM). In many ases of data breaches attackers used this technique, which included the JP Morgan Chase [29] and the Target hacks [16].

NIDS detects malicious network traffic that indicates the execution of a remote exploit. CLM (i.e. network traffic substance) is indistinguishable from a benign login; NIDS (Network Intrusion Monitoring System) is also useless for CLM detection. Meanwhile, owing to its various obstacles, such as a tangled IT architecture, control strategy and tools such as ACLs and Active Directory face a number of roadblocks when seeking to minimise the paths of lateral movement in a market sense. Unrestricted access is given for business sustainability reasons and such that if they fail, information systems may be recovered. Because of this, in the worst-case situation, the website needs logins that will not normally be needed. A previous analysis by Sinclair and colleagues[30] showed that in terms of what they can access, almost half of business network customers are over-entitled. Because of this condition, attackers are able to fly easily across a network to capture their target destinations using stolen passwords. We describe a novel method in this article to detect fake logins that are likely to be used in company networks. Two relevant assumptions are contained in our methodology. The first step of the device login control of

An organisation is liable for coordinating contact practises that are almost entirely repetitive. For example, accounting department employees connect to a server that holds an accounting application, while staff from the human resources department connect to a server that has an HR application activated. The second issue with CLMs is that computers that are not organised as part of an enterprise network are always linked. In general, by utilising stolen credentials to log in from computers in the HR department, a hacker can achieve access to computers in separate departments. Since the hacker would only be allowed to use stolen keys that he already has, they must still move on, as well as already-compromised computers. Thanks to the base rate fallacy, locating unusual logins is complicated. [2] The This, along with the theory of the Network Login Structure, explains why we are implementing a Network Login Structure that defines the usual login patterns within

a given network. To model a network login setup, we build a structure in which the patterns for login entries are automatically extracted. This pattern segment illustrates how most individuals from various computers log in to this unique scenario. To support our theory, deviations are then included. Detection technique to track fraudulent logins that are compatible with a business network's login framework. We use a semi-supervised anomaly-based methodology to put it another way, creating a one-class classifier to detect malicious logins. To extract patterns describing standard network logins, we propose a pattern mining algorithm. We use a market-basket analysis in our model [14]. When you look at a pattern, one of the things you may notice is that some of the consumer (the communication device), the source machine (the computer that is logging in), and the destination computer (the computer from which you are logged in) would be the same. The layout of the network login is defined by the login patterns set together. If a new username is not consistent with the login system, we mark it. It is in general that one can describe the article as follows:

When explaining the idea of detecting Credential-based Lateral Movements using login anomaly detection inside an enterprise network, we examine the possibility of using login anomaly detection to detect Credential-based lateral movements. In this paper, by applying standard login patterns, we present a methodology for the simulation of enterprise network login architectures. In addition, we have an algorithm that extracts consumer login patterns with relative ease from a large data collection. Using real login data sets, we validate our method. In order to categorise logins, protection researchers use stickers. Owing to the natural dynamics of network logins and transitions in institutions, the system has sources of false positives in addition to human review. We answer these complex problems and offer recommendations for other organisations to enhance the accuracy of the method. To the best of our knowledge, the structure and dynamics of logins in an enterprise network have never before been encountered.

Any of the latter is laid out in the following manner. This chapter will provide a description of the structure we have built and the different components that make it up. In section 3, our login pattern mining algorithm is defined below. The username classifier is addressed in Section 4. Section 5 includes our assessment of the process. We also tracked login data from a real network to see how it works in order to make a detailed assessment. The classifier's accuracy, true positive, and false positive are also used. The function of section 6 assessments relevant to this section. Section 7 ends with a review of shortcomings of our present analysis and a glance forward to potential studies in it. Our paper finishes in section 8.

OVERVIEW OF THE SYSTEM

Attackers utilise stolen credentials in a variety of fashions each of which calls for a specific defence strategy. The goal of this paper is to examine deviations from the standard pattern of logins, such as when the user, source, or destination of the login is out of the ordinary. Here, we formally state the problem and show how we have developed a detection technique.
Here is the problem statement: Lateral movement by means of stolen credentials is referred to as credential-based lateral movement (CLM). This method of intrusion utilises a stolen credential to log into a new computer on the network, compromising it. The computer will thus be linked to a chain of other hacked computers. These types of attacks are usually started by phishing attempts that infiltrate corporate networks and compromise a user's workstation. Attacker goal is to compromise high-value assets such as a database so they can obtain credentials and perpetrate a compromise.

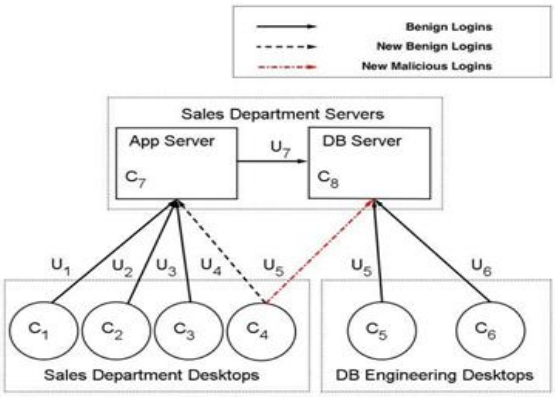

Figure 1: This diagram shows a simplified schematic of lo-gins within a network. Solid lines represent logins that are observed in a time interval in the past. Dashed lines are new logins, one of which is benign (dashed black) and another of which is malicious (dotted red). The benign login is consis-tent with normal login structure, but the malicious login is inconsistent.

This server could either be a financial-specific application server, or it could be a crucial part of the operations. The attacker must continuously harvest new credentials as he traverses the path from one workstation to a target server. Next, he uses those credentials to infiltrate a machine and extend the series of compromises. A set CC of compromised computers and a set CU of compromised user accounts represent a state of an attack (i.e., stolen credentials). An infected computer is one that is in an enterprise network, but which is also under attack by an attacker who attempts to execute an arbitrary programme on it (e.g., malware). Attackers using compromised computers as stepping-stones will then attempt to log in from a stolen credential u on a compromised computer s. Next, they will attempt to compromise a computer that is not already compromised, using the stolen credential u.

An attacker will use valid credentials to gain access to network resources, and some of his logins may not be in sync with standard network logins when it comes to user account information and computers. Although such inconsistencies are unavoidable, the attacker will only be able to use compromised computers and user/system accounts that he has already gained access to, for all of his login attempts to other machines. Using this finding, we can spot malicious logins.

Adversarial Model

An assault on our mechanism has one of the revealing signals. Attackers use passwords for network-wide machines. There is no connection between how an identity is robbed and how it is detected. A consumer can steal the password via a keylogger if they type in a login page. You can also use Mimikatz[8] to view passwords through the use of native memory attack techniques, and use them for local or remote authentication as a new person. Whether it's done by breaking into a computer and then logging on with the stolen credentials, or by gaining unauthorised access to a network and logging in with the stolen credentials, our detection method always finds when those credentials are being used. If program (such as a protocol for file sharing) is used to control the resources of a remote Device, passwords could be manually inserted in a logon window or credentials recovered from a cache. Anyway, the authentication case is recorded by the network incident logging and our system can use it to track events.

The identification of the intruder used to log in to the computer and then the target device is crucial in order to guarantee that our identification algorithm works. Our algorithm cannot identify fraudulent logins if the assailant uses a password typically used by two machines. The attacker cannot, regardless of the case, meet this obligation. For eg, an intruder must look at a helpline manager who logs into the infected computer from a remote source and re-logs it to a different system, then breaks in on another machine from the new machine. Since he steals the login credentials, our algorithm will identify him as he will log in to every other device utilizing the stolen credentials. In addition, attackers would presumably attempt to return to the machine of the back-end support desk manager to restart the method the algorithm is utilizing, since it takes care of the instructions of their login.

As far as identification is concerned, an intruder who is acquainted with applying our algorithm and understands regular network login connections is not an assurance of being found. An attacker must also have the right credential and be located on a computer that typically logs into a destination in order to take advantage of his knowledge and bypass our detection algorithm.

Pattern Mining Algorithm

Our pattern mining algorithm relies like consumer basket research on association laws. It requires two separate processes to obtain network login patterns. The first phase of this algorithm has just been completed and contains the number of each applicant login pattern in the background of the user H. The second stage requires the usage of an algorithm and the amount of t

Session F2: Log(in) Analysis

A single username alluding. For starters, we don't think about patterns that just enforce a login type, without other attributes like the user's name. This is why the amount is

In addition, the login patterns produced by U*, S* and D* are allocated to and picked first (|u | - 1) kilometric (|S - 1) kilometer |D - 1 - 1) The login patterns generated will be the same. Furthermore, if we want to login ("Sales"

Algorithm 1 This algorithm generates all pattern candidates from a given login. The operator * computes power set of a given set.

```
1: procedure EnumeratePatterns(u, s, d)
2:     get <U , S, D >
3:     gen-powerset < U *, S*, D*>

4:     for Ub∈ U * do

5:         for Sb∈ S* do

6:             for Db∈ D* do
7:                 emit-candidate (<Ub, Sb, Db>)
```

The computer orientation scores are computed. We will find the P-shape, named the Ub, Sb and Db Axes, by calculating the degree of alignment between the pattern and each of the three axes.

Rate perfect This score indicates the source direction of the pattern. In this measure, we first find out how many computers have S and are presented in a case.

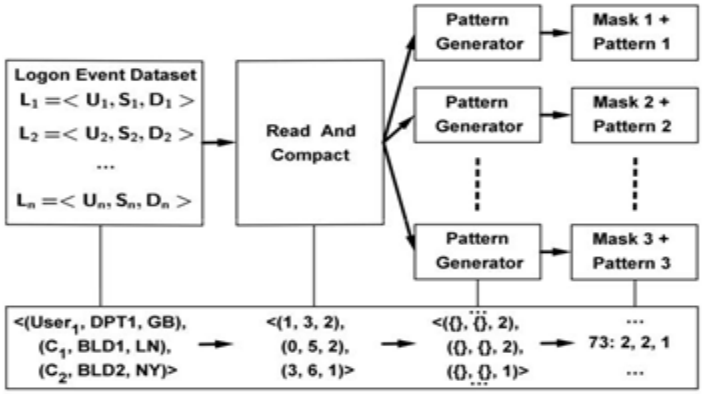

Figure 2: This diagram shows the process of encoding and parallelization of the algorithm for generating the candidate login patterns.

Pattern Matching. A pattern matching classifier first generates all possible combination of attributes related to a login L with attributes=< U, S, D> the same form used to list network login patterns for can-didate. The classifier classifies the username as benevolent if a configuration sequence of network login patterns represents the network layout fits at least one combination of login attributes. This implies that, whenever one trend exists, the relation l =< u, s, d> would be listed as benevolent.

EVALUATION

We assessed our framework over a five-month span using a real data collection of all logins of a multinational financial firm. First of all we look at the network configuration and complexities of the company network logins in this segment. Then, based on hand labels from a variety of security experts, we test the consistency of our warnings. Finally, we measure the incorrect positive and t

Dataset

The login dataset we used encompasses any Login entry, which marks each one of two machines as a special login case. A login link involves the user name, root name and login destination. The data collection contains the regular amount and form of username per each login. The protocol used for authentication is seen by a login sort. Windows netwo is one of the login styles The login dataset we used encompasses any Login entry, which marks each one of two machines as a special login case. A login link involves the user name, root name and login destination. The data collection contains the regular amount and form of username per each login. The protocol used for authentication is seen by a login sort. Windows network is one of the login styles Destination shift covering 2-6 percent of all logins was the most common type of modifications. Approximately 80% of the shift of destination contained a server. Our review reveals that a source shift is

used about 1-2,5% of all logins. Equal portion of source modifications are reflected on servers and desktop. The smallest types of username adjustment is user change and complete change.

On the basis

Provide explanations of habits of members. We start with the collection of all subsets, and each is generated centered auf a set of login entity attributes, to produce three power sets from all subsets. We refer to them as U*, S*, and D* in order to represent the power sets we described. We create the cartesian U – S – S – S – S – S – and D – product, which reflects the complete portfolio, after all nominee trends have been created.

Log(in) Analysis: Session F2

A single username alluding. For eg, trends that enforce only login types that do not include other attributes such as the name of a user are not regarded. This is why the amount is Also, if U*, S*, and D* are assigned logins and the first is chosen, the generated login patterns would equal $(|U*| - 1) \times (|S*| - 1) \times (|D*| - 1)$. In addition, if we want to identify a login ("Sales","Staff") of a user ("Desktop","Sales") who works on a ("Sales-Dept","Server") (see Figure 1), there are 27 different patterns that could be considered (three non-empty subsets of attributes for each element). Depends on each login attribute's number of unique values. The abridged variant of this algorithm is seen in Algorithm 1. network and organization. The goal of our technique is to distinguish benign from malicious login changes.

Experiment by Security Analysts

We also requested several security experts from the same financial institution to provide us with the login information for the evaluation of the logins that our device identifies as malicious if a collection of malicious logins are not present. In this section we explain the device configuration, the assessment technique and the experimental outcomes.

Installation of the device. Our algorithm classifier consists of two components

(a) The sys-tem had first to reduce login patterns in order to allow the second part, the pattern match. We then enter the logins of the last four months and machine and user information in the algorithm of pattern mining. In total, 8 separate attributes were used to define login attributes, including two for users and three for each device. the algorithm of mining pattern

Experimenting with Synthetic Attack traces

Our evaluation based on feedback from security analysts was lim-ited mainly because without knowing the actual number of mali-cious logins we can not measure the number of malicious logins that our algorithm misses. To overcome this problem, we evaluated our system based on several synthesized malicious logins injected into real traces of logins from the enterprise network. Benign logins. We used five consecutive months of login data set to set up and evaluate our approach. We split this dataset into two

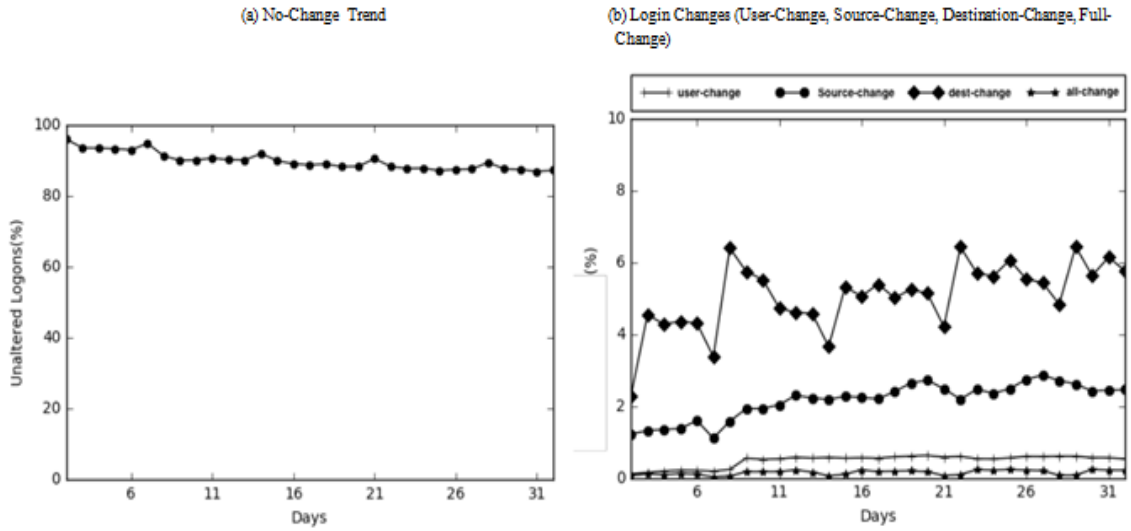

Figure 3: (a) shows the percentage of no-change logins relative to a collection of logins of the past four months. (b) shows the percentage of each category of login change. Most of the changes are related to destination-change. The y-axis is shortened for readability

A wall (or a partition) acts as a buffer between the space and the outdoors. We learned our anomaly detection algorithm to discover the network structure by using four months of login results. We run a test to figure out the false positive rate of our algorithm in the last month of logins.

Trace proof of an assault. To replicate a lateral movement assault inside an enterprise network, we simulated an attack that sometimes happens during penetration tests and used remnants of actual malicious logins. Based on this study, we assume the following to be true:

Already inside a corporate network, the intruder has breached a workstation and is attempting to switch to another.
The intruder is able to intercept every logged-in user's password. This covers machines on which the account was used to log in or from which the account was accessed.

Attacks may be kept out of a network by network access controls. Administrative control for non-admin accounts is the most relevant since only such accounts will obtain it. The intruder must have admin privileges to hack a computer.

A stealthy intruder uses as little logins as necessary because further logins mean that an alarm would be generated. To award the at-tackers we simulated credit, it should be noted that they only tried to log into five other systems on the network.

To randomly select workstations as a source of malicious logins, we used a random number generator to choose a range of workstations. Finally, we searched all machines and found all passwords that were still involved. An intruder may use these credentials to log into other machines and hack them. Next, we agreed to pick five random destination computers from all of the compromised computers and order each one to execute a malicious login to a nearby target machine. This method produced 150 malicious logins that were in the form of traces.

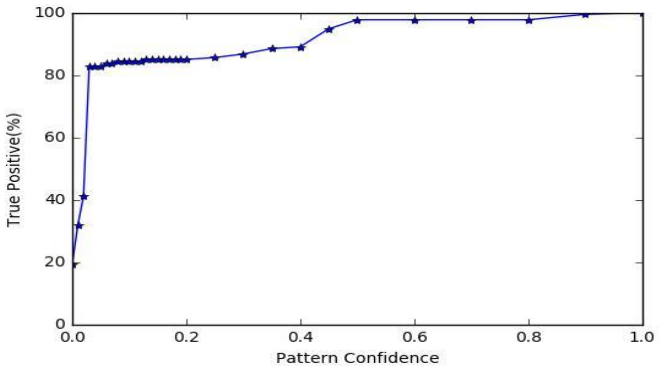

Figure 4: True positive rate of the algorithm based on differ-ent threshold for pattern confidence.

made up of <u, s, d>Ses optimistic examples from our dataset were included in this series of logins. We used the malicious logins that we inserted into the data set to evaluate the algorithm's efficiency.

An significant factor when settling on a design is the consistency of the patterns. If we use subpar patterns for classification, our success would be based on how well our detection algorithm operates. Trust indicators act as a strong predictor of trends' quality. Higher OE scores denote a greater degree of reliability, so trends of higher OE scores have higher confidence. In this segment, we report the system's success concerning different pattern trust thresholds. Positive statistical value. A crucial component of any de-detection algorithm is the capacity to recognise malicious logins. The correct positive rating is called “com-posite positive rate.sprinkled with T P To test this, we counted the amount of occurrences of the letter 'T' and the letter 'F.'

malicious research data logins found by the classifier. If the trust score for the classifier contains a login pattern in its meanings, the true positive rate differs. Orientation ratings with a higher threshold produce trends with higher confidence. When utilising a higher trust threshold, our algorithm is less inclined to align a malicious pattern with a valid pattern wrongly. See Figure 7, which depicts the proportion of true positives compared to the trust level for detecting trends. This will allow our framework to capture more malicious logins.

FALSE POSITIVES Levels An algorithm for identifying defective equipment is only feasible if it creates only a small amount of false alarms. This system will generate varying amounts of false alerts based on how confident it is that it has

identified a pattern. To lower the system's threshold, more login patterns would be considered acceptable. Also, as a result, new benevolent logins would be more likely to generate a matching pattern, and this results in a reduced incidence of false positives. The findings from Figure 8 display the algorithm's false positive rate at various thresholds. The lower the level for pattern belief, the lower the false positive rate. The ROC curve is a test of the efficiency of various kinds of contact in relation to different goals. The pattern trust threshold the method uses for selecting qualified patterns influences both the true positive rate and the false positive rate. We used the receiver operating characteristic (ROC) curve to demonstrate this relationship and evaluate a balancing threshold. On the X-axis of a ROC curve, the classifier's false positive score is plotted against its real positive rate on the Y-axis. We arrived at our data points by looking at different confidence levels in patterns. The ROC curve for our classifier as seen in Figure 9. We will identify 82% of fraudulent logins thus providing 0.3% of false alarms using this graphic.

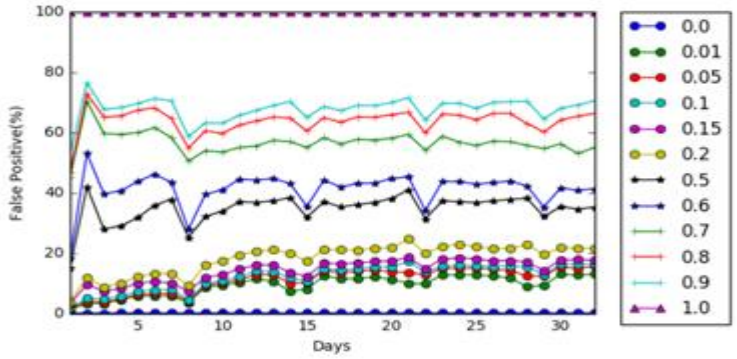

Figure 5: False positive rate relative to different thresholds for pattern confidence.

RELATED WORK

Using the notion of user groups and artefacts, network administrators may authorise or refuse a community of users access to a set of resources. This mechanism is effective in prohibiting an employee from obtaining knowledge or services that they could otherwise have access to. Access is given to properties that a consumer can require access, even though the user seldom requires it. In reality, the key explanation for granting further access than is needed at any specific moment in time is business continuity. As a consequence, 50-90% of consumers are over-entitled [4]. This unnecessary permissions allow an intruder to travel inside a network almost openly. This work complements the processes for access management and is ideal for a corporate setting where strong emphasis is provided to business continuity. Where a login is uncommon, our device produces an alarm. Security researchers can verify that an extremely unlikely login is not malicious by in-examining the warnings.

Activity control inside Networks. Attackers also switched to indirect attack tactics in reaction to force-ened networks and servers that avoid overt external attacks. In one such process, utilising a phishing attack [5, 22, 34], the attackers compromise a desktop inside a network. Then they use this foothold to hack other machines or servers that otherwise could not reach important data they host. This method of attack motivates the development of malicious traffic-based surveillance and tracking tools inside corporate networks [11, 23, 23, 35]. Using sensors mounted on computers and networking equipment, these methods depend on a large volume of data obtained from network and host activities. Only compromised machines have become the subject of several identification approaches. Yen et al.[35] have

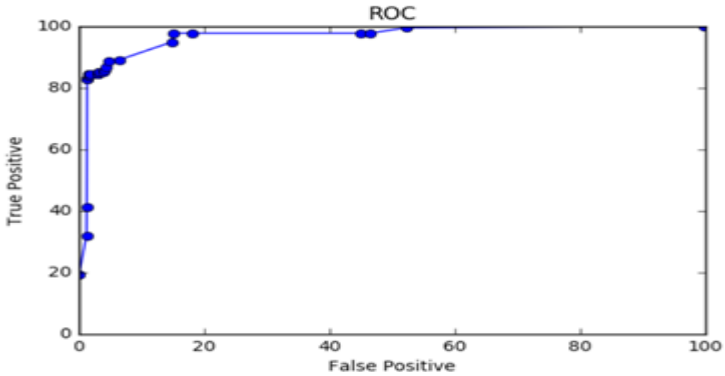

Figure 6: ROC diagram shows the false positive vs. True pos-itive of the detection method.

suggested a scheme for the identification of compromised workstations that automatically mines information from the log data provided by a large variety of protection products (e.g. anti-virus, firewall). A system for fusing data from multiple sources within a network to identify coordinated assaults, including lateral movement, has been suggested by Fawaz et al.[11]. Oprea et al. [23] suggested the replication of a belief Technique that decides the state of a machine, provided previous awareness of its past state and connections with external resources (i.e. benevolent vs. malicious) (e.g., external websites). They've been able to discover new hostile organisationsutilising this strategy. These methods do not use credential use details, and therefore the essential class of credential-based lateral movement we discussed in this paper can not be identified.

Malicious User Activities Identification. While remote exploits and zero-day vulnerabilities are used by attackers, these methods are overrated [24]. Instead, credential-based lateral movement (CLM) threats, utilising usernames and passwords to switch laterally across network computers[3], prevailed. Many recent works have examined credential-based assaults. Gonğalveset al.[13] used certificate usages focused on an unsupervised clustering technique to spot misbehaving machines. They used features such as the amount of logins that were active and failed, as well as detection data on admin logins. Their system is not capable of detecting CLM so no statistical anomalies such as repeated logins are seen. In addition, since it depends on the structure of logins instead of the frequency of them, our method will classify a single login of an intruder. A controlled computational approach for the classification of logins in client-server interactions was introduced by Freeman et al.[12]. To distinguish benevolent and hostile logins, they use several features, including IP reputations. Our strategy, in contrast, is connected to logins inside an enterprise network. Beyond the client-server framework, these logins require a more complicated series of connections between computers. Our methodology is also distinct from theirs, since our classifier does not require labelled data for preparation. Instead, we use a method of semi-supervised anomaly detection. Siadati et al., respectively.

[28] has used a signature-based approach for fraudulent login identification. To recognise and establish the signature of malicious logins, their method relies on iterative vi-sual login discovery inside the organisation.

Our method has possible applications in theft and prevention of insider attacks. Eberle et al.[9] have suggested a form of graph-based monitoring for the identification of anomalous behaviour linked to computer interactions within a network. In contrast with a model of interaction they construct atop the most common subgraphs of the relations, their method measures the adjustments of a graph of interactions. However, owing to network complexities from malicious ones, it does not accurately discern benevolent shifts that arise.

Methods of Assessment. Different techniques may be used to evaluate anomaly detection approaches, based on the availability of suitable test data [10, 17, 21, 25]. When the ground reality is available [18, 31], the perfect scenario is. We gathered a small data collection of branded logins based on the marking of some security researchers, analogous to [15], and evaluated the consistency of our device based on that. Another approved form, which is to establish synthetic attack traces and insert the traces into the benign records, was also used[6, 7, 20, 27, 33].
defines rules for access to network resources based on the role of users.

LIMITATIONS AND DISCUSSIONS

Escapes. In order to identify malicious logins that are not compatible with the usual login framework, our approach utilises the network login structure. An intruder who is aware of an application scheme and identifies a target company's login structure can attempt to avoid detection. The intruder must have the proper combination of username and device to connect to the target computer in order to mimic a valid login. For an intruder to fulfil certain requirements, it is not always possible. It is more challenging for an intruder to fulfil these requirements at the outset of a lateral movement assault to avoid detection since the attacker has less infected machines and stolen passwords. Our device would also be capable of identifying threats in their early stages. To escape detection, an attacker can combine the CLM method with a vulnerability-based lateral motion. It is also strongly advisable to use our methodology alongside others that identify bugs in remote applications.

Assault by Poisoning. We use logins for training our classifier in a time in the past. With the intention of misleading the pattern miner module to include an unauthorised pattern in the collection of login patterns, it is possible for an intruder to build any logins. We just use logins that have existed often enough in the past to prevent this form of poisoning. More precisely, we measure the number of days in the history in which a password has existed. And if this percentage is above a certain level can we have a login in the preparation. We used logins in our experiment that happened in more than 10% of the days in the past. And if more than 10% of the days are signed in by an intruder, this incident would not simply mean a

new trend for our method. Instead, it must log in from a sufficient number of separate source computers of the type to a sufficient number of destination computers of the type to show trends with the minimum necessary orientation scores using suitable usernames. Therefore, without sacrificing detection, an intruder won't be able to contaminate the training results.

Restrictions. While we had access to a rare data collection of a corporation's millions of logins, our analysis is only confined to one company and one form of company. We are well conscious of the restrictions of generalising this paper's results to other networks, especially those with more login complexities and very distinct structures. In fact, the stability of the login system differs from business to company. In a production setting such as a software firm, for example, the login structure differs drastically over time as any adjustments to a project could bring major changes to the login structure. Therefore, the classifier changes can't keep up with the network dynamics.

Some of the innocuous logins triggered by abrupt shifts in a network, such as activation of a disaster relief centre, may not be classified correctly by our methodology. A possible solution to this issue needs the involvement of protection experts prior to or during the catastrophe recovery period to whitelist certain login trends. At the time of the disaster recovery evaluation, another potential fix is to train a separate model to use the system as a disaster happens.

Since we did not have access to such details, we have not analysed the impact of a longer time of logins as input for the pattern miner. We have not researched the optimum window at which the algorithm should be retrained for the same purpose. Nevertheless, it is advised to retrain the algorithm periodically, depending on the sum of improvement in logins from 5% to 15% after a month. Our quick algorithm allows the algorithm to be retrained efficiently in a short period of time.

Similar to all other study in the area of intrusion detection, our work suffers from canonical anomaly detection problems.

CONCLUSION

To the extent of our understanding, this is the first paper reporting an organisational network's internal and login dynamics. To build the notion of network login layout, we used the insights obtained from our research to model typical logins of business networks based on triplets of characteristics of users and machines participating in network logins. To remove such login patterns automatically, we have built a rapid and scalable pattern min-ing algorithm. We created a binary classifier to identify structurally anomalous logins by utilising the network login layout to model the class of benign logins. Based on the marking of security experts as well as synthetic assault traces, we tested our framework. Our assessment reveals that with 0.3 percent false warnings, our device will identify more than 82 percent of malicious logins utilising an acceptable data set for testing.

REFERENCES

[1]. APACHE. 2017. Spark: A lightning-fast cluster computing. https://spark.apache.org. (2017).
[2]. Stefan Axelsson. 2000. The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security (TISSEC) 3, 3 (2000), 186–205.
[3]. Schneier B. 2016. Credential Stealing as an Attack Vector. https://www.schneier.com/blog/archives/2016/05/credential_stea.html. (2016). [Online; accessed 15-Feb-2017].
[4]. Schneier B. 2016. Real-World Access Control. https://www.schneier.com/blog/archives/2009/09/real-world_acce.html. (2016). [Online; accessed 19-May-2017].
[5]. Businessinsider. 2014. How The Hackers Broke Into Sony And Why It Could Happen To Any Company. http://www.businessinsider.com/how-the-hackers-broke-into-sony-2014-12. (2014).
[6]. Baris Coskun, Sven Dietrich, and Nasir Memon. 2010. Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts. In
[7]. Proceedings of the 26th Annual Computer Security Applications Conference. ACM, 131–140.
[8]. Kaustav Das and Jeff Schneider. 2007. Detecting anomalous records in categorical datasets. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 220–229.
[9]. Benjamin DELPY. 2014. A little tool to play with Windows security. https://github.com/gentilkiwi/mimikatz. (2014).
[10]. William Eberle, Jeffrey Graves, and Lawrence Holder. 2010. Insider threat de-tection using a graph-based approach. Journal of Applied Security Research 6, 1 (2010), 32–81.
[11]. HadiFanaee-T and Joao Gama. 2014. Event labeling combining ensemble detec-tors and background knowledge. Progress in Artificial Intelligence 2, 2-3 (2014), 113–127.

[12]. Ahmed Fawaz, Atul Bohara, Carmen Cheh, and William H Sanders. 2016. Lat-eral Movement Detection Using Distributed Data Fusion. In Reliable Distributed Systems (SRDS), 2016 IEEE 35th Symposium on. IEEE, 21–
[13]. David Mandell Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. 2016. Who Are You? A Statistical Approach to Measuring User Authenticity. In NDSS, The Internet Society.
[14]. Daniel Gonçalves, João Bota, and Miguel Correia. 2015. Big Data Analytics for Detecting Host Misbehavior in Large Logs. In Trustcom/BigDataSE/ISPA, 2015 IEEE, Vol. 1. IEEE, 238–245.
[15]. Jochen Hipp, Ulrich Güntzer, and GholamrezaNakhaeizadeh. 2000. Algorithms for association rule miningâĂŤa general survey and comparison. ACM sigkdd explorations newsletter 2, 1 (2000), 58–64.
[16]. Jaeyeon Jung, Vern Paxson, Arthur W Berger, and Hari Balakrishnan. 2004. Fast portscan detection using sequential hypothesis testing. In Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on. IEEE, 211–225.
[17]. Krebsonsecurity. 2014. Target Hackers Broke in Via HVAC Company. http://krebsonsecurity.com/2014/02/target-hackers-broke-in-via-hvac-company/. (2014).
[18]. AnukoolLakhina, Mark Crovella, and Christophe Diot. 2004. Diagnosing network-wide traffic anomalies. In ACM SIGCOMM Computer Communication Review, Vol. 34. ACM, 219–230.
[19]. Richard Lippmann, Joshua W Haines, David J Fried, Jonathan Korba, and Kumar Das. 2000. The 1999 DARPA off-line intrusion detection evaluation. Computer networks 34, 4 (2000), 579–595.
[20]. Alistair G Lowe-Norris and Robert Denn. 2000. Windows 2000 active directory. O'Reilly & Associates, Inc.
[21]. Matthew V Mahoney and Philip K Chan. 2003. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In International Workshop on Recent Advances in Intrusion Detection. Springer, 220–237.
[22]. George Nychis, Vyas Sekar, David G Andersen, Hyong Kim, and Hui Zhang. 2008. An empirical evaluation of entropy-based traffic anomaly detection. In
[23]. Proceedings of the 8th ACM SIGCOMM conference on Internet measurement. ACM, 151–156.
[24]. NYTimes. 2014.Neglected Server Provided Entry for JP-Morgan Hackers. http://dealbook.nytimes.com/2014/12/22/entry-point-of-jpmorgan-data-breach-is-identified/. (2014).
[25]. Alina Oprea, Zhou Li, Ting-Fang Yen, Sang H Chin, and SumayahAlrwais. 2015. Detection of early-stage enterprise infection by mining large-scale log data. In
[26]. Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on. IEEE, 45–56.
[27]. Joyce R. 2016. USENIX Enigma 2016 - NSA TAO Chief on Disrupting Nation State Hackers. https://www.youtube.com/watch?v=bDJb8WOJYdA. (2016). [Online; accessed 15-Feb-2017].
[28]. Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and JD Tygar. 2009. Antidote: understanding and defending against poisoning of anomaly detectors. In Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, 1–14.
[29]. Jerome H Saltzer. 1974. Protection and the control of information sharing in Multics. Commun. ACM 17, 7 (1974), 388–402.
[30]. Taeshik Shon, Yongdae Kim, Cheolwon Lee, and Jongsub Moon. 2005. A machine learning framework for network anomaly detection using SVM and GA. In
[31]. Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC. IEEE, 176–183.
[32]. Hossein Siadati, Bahador Saket, and Nasir Memon. 2016. Detecting malicious logins in enterprise networks using visualization. In Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on. IEEE, 1–8.
[33]. Jessica Silver-Greenberg, Matthew Goldstein, and Nicole Perlroth. 2014. JPMor-gan Chase Hack Affects 76 Million Households. New York Times 2 (2014).
[34]. Sara Sinclair, Sean W Smith, Stephanie Trudeau, M Eric Johnson, and Anthony Portera. 2007. Information risk in financial institutions: Field study and research roadmap. In International Workshop on Enterprise Applications and Services in the Finance Industry. Springer, 165–180.
[35]. Robin Sommer and Vern Paxson. 2010. Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE symposium on security and privacy. IEEE, 305–316. Verizon RISK Team. 2017. 2017 Data Breach Investigations Report. (2017).
[36]. Florian Tegeler, Xiaoming Fu, Giovanni Vigna, and Christopher Kruegel. 2012. Botfinder: Finding bots in network traffic without deep packet inspection. In Proceedings of the 8th international conference on Emerging networking experiments and technologies. ACM, 349–360.
[37]. WSJ.2014. Home Depot Hackers Exposed 53 Mil-lion Email Addresses. http://www.wsj.com/articles/home-depot-hackers-used-password-stolen-from-vendor-1415309282. (2014).
[38]. Ting-Fang Yen, Alina Oprea, KaanOnarlioglu, Todd Leetham, William Robertson, Ari Juels, and EnginKirda. 2013. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In Proceedings of the 29th Annual Computer Security Applications Conference. ACM, 199–208.