Journal of Information Security and Applications 67 (2022) 103167

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Insider-threat detection: Lessons from deploying the CITD tool in three multinational organisations

Arnau Erola *, Ioannis Agrafiotis, Michael Goldsmith, Sadie Creese

Department of Computer Science, University of Oxford, United Kingdom

ARTICLE INFO

Keywords:
Insider threat
Anomaly detection
Cyber security

ABSTRACT

Insider threat is a persistent concern for organisations and business alike that has attracted the interest of the research community, resulting in numerous behavioural models and tools to tackle it. However, the effectiveness of detection of these tools has scarcely been demonstrated in real environments. In order to fill this gap, we collaborated with three multinational commercial organisations who trialled our anomaly detection system, and worked with us to understand performance constraints for insider threat detection deployment and innate weaknesses in their operational contexts. During a period longer than a year, we were provided access to real data in their premises and interacted with their cybersecurity analysts to understand their systems, validate the results and identify best practices for mitigating insider threat. In this paper, we provide details on the architecture used in our tool, the methodology followed to validate its performance and we elaborate on our experiences in implementing the tool in the three corporate environments. We present the results obtained from deploying the detection system in real network infrastructure over a period of six months, the lessons learned, issues experienced, and potential limitations.

1. Introduction

The insider-threat problem is constantly growing in magnitude, resulting in significant damage to organisations and businesses alike. The overgrowing implications of the insider threat problem has attracted the interest of the research community over the last twenty years. The in-depth knowledge which insiders possess of the internal security controls, the monitoring practices and the modus operandi of organisations allows for targeted and sophisticated attacks with dire consequences. Detecting and preventing insider attacks raises significant challenges and insightful research has been focusing on understanding the human element, identifying patterns of attacks and creating conceptual models of insider behaviour [1–3].

In this context, the Corporate Insider Threat Detection project (CITD) was created to explore how automatic detection of insider threats could be improved via a combination of automatic anomaly detection and a deepened understanding of the human behavioural aspects. It also considered the operational context within which such a detection approach would exist since this could impact upon successful deployment. In the context of our research an insider threat is anyone with privileged access (e.g., an employee, contractor, client or business organisation) to an organisation's data, systems or infrastructure, that intentionally abuses this access for some gain.

Extensive research has been conducted in the area of insider threat, resulting either in theoretical frameworks modelling the human behaviour or on detection tools. Despite the fact that many defence mechanisms have been proposed, their effectiveness has rarely been yet demonstrated on real case studies, relying mainly on synthetic datasets for validation. Scarce research approaches have used anonymised versions of datasets from real organisations, but there has been no comparison on the detection rate or utility before and after the data generalisation [4–6]. To address the gap of real case studies, we provide insights into the performance of the anomaly-detection approach developed for the needs of the CITD project [7] within corporate environments.

Our aim is twofold: to validate if our anomaly-detection approach can be applied effectively to any real available network and host data (where to be effective means to be able to detect insider threats with minimal false-negatives, manageable levels of false-positives and to scale up to handle large and diverse datasets); transfer knowledge to the community and vendors of protective-monitoring software from our experience in deploying an anomaly detection system to different environments, with diverse needs and datasets.

* Corresponding author.
E-mail addresses: arnau.erola@cs.ox.ac.uk (A. Erola), ioannis.agrafiotis@cs.ox.ac.uk (I. Agrafiotis), michael.goldsmith@cs.ox.ac.uk (M. Goldsmith), sadie.creese@cs.ox.ac.uk (S. Creese).

https://doi.org/10.1016/j.jisa.2022.103167

Available online 19 April 2022

This paper documents the results of our research in addressing the above requirements, by conducting industry trials with **three commercial organisations**, all of which were concerned about insider threat, and all with an existing security capability. Given the lack of ground truth on the data accessed, our findings rely on interviews with security analysts of the organisations, who have deep knowledge of their systems and the insider attacks they faced in the past. To our knowledge, this is the first time that insights and lessons learnt from testing an insider threat detection system across multinational organisations have been presented. We identify lessons learnt by comparing our procedures and results across all three implementations of the system. These lessons can apply horizontally to any corporate environment and propose solutions on addressing limitations in developing detection systems that emerge from current security practices.

In what follows, Section 2 provides an overview of the literature on insider threat, Section 3 describes the high-level architecture of CITD detection tool, while Section 4 elaborates on the methodology we have adopted to conduct the trials. Section 5 presents the result of our trials and Section 6 identifies lessons learnt from our experience. Finally, Section 7 concludes the paper.

2. Reflecting on insider threat research

The significant challenges in detecting insider-threat actors, compared to external threats, has attracted the interest of the research community over the last twenty five years. This maturing in the field of academic output has been captured by scholars who have constructed systematic reviews of the relevant literature [8–10]. These reviews conclude that insightful discourses can be categorised in attempts to highlight common elements in insider attacks [1,11], research into identifying human behavioural factors resulting in high-level explanatory models and works that have designed and implemented anomaly-detection systems based on machine learning or policy violation [3,6,12].

2.1. Conceptualising the insider-threat problem

The CERT research programme at Carnegie Mellon University pioneered research in conceptualising insider threats and actors [1,11,13,14]. Researchers there analysed an extensive number of real insider threat cases and categorised these attacks into four classes: namely, Information Technology (IT) sabotage, Intellectual Property (IP) theft, data and financial fraud, and espionage. Using system dynamics as a methodology, critical paths were highlighted that determined based on qualitative characteristics the motivations of employees to commit these attacks (i.e, disgruntlement or dissatisfaction, stress, narcissism). The CERT work paved the way to understanding the nature of insider threats, resulting in a series of studies.

The CERT work has inspired scholars who have designed complementary models to describe behavioural aspects and motivations of insider actors. Sarkar [15] elaborate on factors that can be considered as precursors for an attack (i.e, historical behaviour, psychological traits) and distinguish between actors who act maliciously on purpose and those that unintentionally facilitate an attack due to negligence or inconvenience. Similarly, Nurse et al. [16] provide a framework that explains the steps an insider may follow before executing an attack and identify critical precursors. The US Department of Homeland Security has shed more light on how personality traits may reveal predispositions for insider threat and discuss prevention measures [17], while Pfleeger et al. [18] designed a taxonomy to capture actions that insiders follow by eliciting elements from the environment in which an organisation functions and the systems exploited.

Similar frameworks elaborate on factors which may be considered as precursors of an attack. Schultz [19] proposed a framework for characterising indicators of insider threat activity, including technical and sociotechnical markers, which are aggregated in a final score based on the scrutiny of real insider cases. The work does not provide, however, this scrutiny to validate the framework. Greitzer et al. [20] provides another framework integrating technical data and psychological data with the use of dynamic bayesian networks to recognise patterns at higher semantic levels. Caputo et al. [5] use a similar approach, combining contextual information and technical data (e.g. office job title, project, ...), utilising a bayesian network constructed from expert judgement. All three studies agree that the combination of technical data and organisational data (psychological or sociotechnical) is needed to create a knowledge base of precursors of insider activities to allow prevention and early detection.

2.2. Detecting insider threats

Traditionally, literature has emphasised on implementing and validating systems to automatically detect the presence of insider threats based on the digital footprint that these actors leave [21]. Such systems may benefit from the behavioural elements and precursor activities captured by high-level models mentioned in the previous section, when these are mapped to changes in employees' digital profiles.

A simple approach to automatic detection of indicative insider activity is monitoring of ICT policy violations through rule-based approaches. Zhang et al. [22] describe a state machine system able to capture such violations, as well as monitor for patterns of insider threat activity. The most common method used to detect insider threats is the detection of anomalies in the digital profiles of employees. Systems build a profile over a period of benign behaviour based on observed sequences of actions, which is then considered the normal behaviour. Actions that deviate from the baseline are considered indicative of anomalous activity. Parveen and Thuraisingham [12] used unsupervised learning techniques to process large volumes of streaming data and by introducing the term of "concept-drift" indicate drifts in employees' behaviour which might be indicative of suspicious activity.

The majority of the literature, focuses on designing baselines to profile employees' behaviour by extracting features which are indicative of insider threat activity. Features differ based on the data that organisations collect and a variety of algorithms is used to decide deviations of interest [23–25]. A noteworthy example is presented in [26] where authors acquired access to data from employees' laptops. To validate their approach, collected data was considered benign and the authors inserted logs that were considered as malicious activity. More than 100 features were implemented and seventeen algorithms were tested, however, the authors provide little evidence on how features were identified or why the logs are representative of insider activity. Rashid et al. [27] describe a system for insider detection utilising Hidden Markov Models (HMM). The authors design features based on the CERT dataset, provide a thorough analysis of the accuracy of the HMM algorithm and identify the values for hyper-parameters that optimise the system's effectiveness. Their work, however, highlights the performance issues of the algorithm since it does not scale-up efficiently with large datasets.

Researchers have attempted to infer psychological traits and behavioural aspects from users' digital activity and incorporate this information to anomaly detection algorithms [28,29]. It is worth noting the work from Brdiczka et al. [30], who rely on psychological profiling of employees to filter false positive alerts from a detection system. In a similar vein, Arulampalam et al. [31] use Bayesian networks to infer the behavioural traits of users based on sentiment of texts and social network analysis. Chen et al. [32] focus on cases where insiders collude to execute an attack. They propose a "user relationship network" based on unsupervised learning techniques to observe suspicious collaborative activity. They validate their model on access logs from an electronic health records system in a medical centre.

One of the main drawbacks for insider threat detection is that data is usually represented in natural language and it is seen as highly sensitive for the organisations. Costa et al. [33] presented an ontology

of potential indicators of insider activity created from 800 insider cases using natural language processing techniques. Network activity and human resources data are combined to create insider profiles. An interesting advantage is that insider data is abstracted in a way that would allow sharing with other organisations without disclosing sensitive data. Greitzer et al. [34] also combines technical and organisational factors to create richer individual profiles, and describes them in an extension of the previous ontology. They validate the approach with a case study and the elicitation of expert judgement to determine values of indicators of risk.

Finally, research has focused on formalising the problem of insider threat to provide insights for modelling. Agrafiotis et al. [35] provide a grammar to allow organisations to unequivocally capture policies to restrict insider threats and to enable the description of patterns of insider activity. Bishop et al. [36] formalise the link between tasks and agents to describe processes. These processes can be analysed to identify signs of compromise and can highlight counter measures to enhance the resilience to insider attacks.

A key observation in the reviewed literature is that the design and implementation of detection systems is drived by the different types of data available to the research community, rendering key to the success of a system the variety and quality of the data in which these tools are validated. Most of the published work is validated on synthetic data, failing to reflect the complexities of real corporate environments. In the scarce occasion where actual data from organisations is available, malicious activity is inserted due to lack of ground truth. This practice may distort the actual effectiveness of these systems in real cases of insider threat, because it relies on inserting malicious data that are representative of actual insider threats. Therefore, results may differ significantly if a system is tested in data from a real environment where an actual malicious behaviour exists in this dataset. Scarce examples of such studies which used real data in anonymised form for validating systems are [4,5]. In [4] they use multimodeling to evaluate insider threat inference capability on 16 behaviour indicators. To provide privacy, the data used in the study is generated to mach data statistics from a real organisation, or aggregate into summary statistics and histograms. Caputo et al. [5] captured data from a real corporate intranet where a red team was acting as malicious insider.

Our work in this paper endeavours to fill this gap by deploying a detection system in real organisations. To the best of our knowledge, this is the first time such work is reported and we provide novel lessons learnt from our experience that apply not only to the validation of the tool but to the whole process of deploying a detection tool in a large network.

3. CITD anomaly detection system

The CITD anomaly detection system is an investigative tool to detect potential insider threat behaviour developed with the support of the UK National Cyber Security Programme in conjunction with the Centre for the Protection of National Infrastructure (CPNI). The preliminary versions of the system [6,37] have been refined and enhanced in performance. Additional functionalities to detect violation of policies (tripwires) tailored for insider-threat behaviour and to monitor for well-known patterns of insider threat have been implemented. The following describes the system architecture, how the detection system works and foundations underpinning the approach.

The architecture of CITD system, as illustrated in Fig. 1, is composed of several modules. The *Parser Module* ingests logs that are captured from the organisation, removes redundant data (cleaning task) and converts these to a readable CSV format. Logs derive from a variety of sources (e.g. user devices, servers or SIEM tools), therefore, ad-hoc parsers are written for each type of log, creating a wide variety of CSV structures with fields of interest. The cleaning task, does not only involve the extraction of data from the logs into fields in the CSV format, but performs complex tasks by triangulating different logs to derive the required information in a new field (e.g., link usernames to IP addresses to determine online activity per user and not per IP address, execute timestamp corrections for activities performed in servers with different time zones to the servers that record them). Once the logs are in a tool-readable format, they are send to the *Activity Parser* where they are enriched with extra content, if available, for example metadata or email keywords, and classified by user and corresponding role. Role describes the positions held by users in the organisation (i.e., IT, finance, HR) and enables the comparison of behavioural characteristics of users who perform similar tasks in an organisation.

The system creates and maintains two profiles for every user; the first is a personal unique profile and the second contains data from all users who hold the same role. Once all logs are parsed and reformatted, data corresponding to every user is appended to their two profiles. Once this process is concluded, the anomaly detection mechanism starts. CITD uses a three-layer approach as alerting system. The first layer (L1) is context-specific and consists of alerts that violate policies indicative of insider threat behaviour or capture activity similar to well-known patterns of insider threat activity. Policies are created by identifying interesting situations which can potentially lead to insider threat (i.e. detecting VPN access using the same username from two different locations, which is physically impossible), or flag violations in corporate procedures (i.e. prohibition of usage of non-authenticated USB devices on corporate systems). The second type of L1 alerts, captures patterns of insider attacks. Our tool relies on a framework for characterising insider attacks to provide a default number of patterns [38]. To build the framework itself, we adopted a grounded-theory approach and examined a dataset of 80 real insider-threat cases. Once we examined these cases, we set about identifying key aspects of the insider attacks relating to the attackers, their background, their behaviours, and the attacks that they conducted. The culmination of our analysis was a framework where we defined how each of the key aspects relate to each other and the general progression of an insider attack.

The next layer (L2) is focused on detecting when users deviate from what is considered by the system as their normal behaviour. To that end, the system extracts features for every user and role profile. These features represent users' daily activities which may relate to insider threat behaviour if such behaviour changes (e.g., number of new files accessed in a given day). Features are clustered to create different sets (one feature may appear in more than one sets) that are representative of the type of activity they describe (e.g., file access, online browsing, login activity). We refer to these sets of features that intended to detect specific insider behaviour as *anomaly* sets. For example, in order to detect data exfiltration we probably ought to look for intensive file activity and/or increase in network usage, so we will group these features in a single *anomaly* labelled "data exfiltration". Features are extracted from the user profile in daily basis before the anomaly detection starts. We follow a semi-supervised approach, thus we need to train the model with certain amount of days of activity in order to create an accurate representation of the behavioural norm. Principal Component Analysis (PCA) is used to reduce the dimensionality (number of features in the set) of the *anomalies* and cluster similar behaviour (data from the training period) closer to the origin of the Eigenvectors. Then the values of the features describing user's daily activity in every set are plotted in the reduced-dimensional space, and if these are considered abnormal (deviating from the clustered normal activity) alerts are raised.

Lastly, the third layer (L3) monitors for deviation in the alerts generated in the previous layer (L2). The rationale behind this layer is that for systems where certain types of alerts are very common, such as alerts raised by an IDS after failed login events, the high number of false positives in L2 can conceal true positives. Over the course of our research, we found that performing anomaly detection to find different sets of alerts that hide insider activity behind an obfuscating screen of false positives works very well in reducing the number of false positives and generates alerts worth investigating further [6]. For example, a

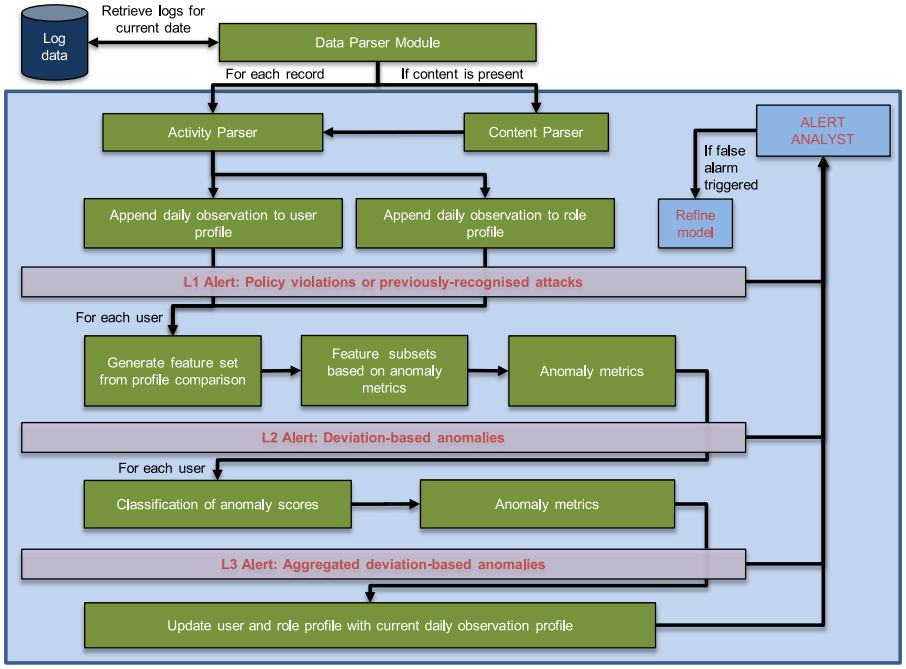

Fig. 1. Overview of the architecture of CITD anomaly detection system. The system process incoming log data records and construct daily profiles of user and role behaviour to assess the risk posed by their actions. The system rises three types of alerts: policy violations, deviation of users' behaviour and deviation of generated alerts. Alerts are dealt with by analysts who may provide feedback on the validity of the alert and reconfigure the sensitivity of the system.

user may be raising low severity alerts for a number of days, which may be considered of little concern when examined individually, but when these alerts are examined in tandem, the overall severity of the situation may be different. This is particularly true for insiders who possess excellent knowledge of the deployed detection systems and try to "fly under the radar" in their activities.

Alerts in L2 and L3 are classified according to their severity. L2 Alerts considered of medium severity are those that the deviation of the user's activity and user's profile is between 1 and 3 standard deviations, while alerts of high severity have a standard deviation greater than 3. We coin them *orange* and *red* respectively. To differentiate L3 alerts in our visualisations (see Section 5), we colour blue L3 alerts and refer to them as "blue alerts" in some instances.

As described previously, during an assessment the CITD system can trigger an alert to the analyst to notify that an abnormal behaviour is observed. Alerts are provided to the analyst in a visual format as shown in Section 5. The analyst can investigate the alert and decide to act if it is correct or discard it as a false positive. Should the analyst observe that too many false positives are generated of this type of alert (anomaly), the parameters of the system can be refined to avoid repeated alerts in future. Parameters include the training period, the dimensionality reduction, the number of features included in sets or even the scale of deviation before a change in behaviour is characterised as abnormal. This, however, implies that the system will restart as the anomaly detection metrics have to be recalculated using the new parameters.

4. Methodology

In this section, we describe the methodology that we adopted in order to conduct the trials in a comprehensive and concise manner. For every trial we spent six months deploying the system, and obtaining and verifying the results. Our approach consists of a description of the set-up phase required to get the tool in place, the pre-processing period to design the parsers and create the necessary features, and a framework capturing how we validate the performance of the tool during the trials.

4.1. Pre-processing period

A minimum of four-weeks was established as necessary to install and configure the tool for every organisation. Meetings were held once or twice a week with organisations' senior security officers, complemented with desk research off-premise and email communication. During the first week, we aimed to obtain a good understanding of the logs available, the information they contain and their format. Details were sought to better understand the threats that organisations had experienced in the past and the assets prominently targeted by insiders.

The second week was dedicated to identify the data needed from the logs we had access and in writing the appropriate parsers to capture this data. Extensive testing was required to ensure that all parsers collected the desired data without errors. Distinguishing the redundant fields was critical to the performance of the system, since the amount of data to be processed was reduced dramatically, boosting the speed of the training and detection period.

The third week revolved around implementing the insider threat policies in the tool. It was crucial to capture the security policies which organisations had in place. For every organisation we obtained access to internal security documents and highlighted the policies which were relevant to insider threat. We then updated our library of policies and implemented those which were relevant to the data we had at our disposal. Additionally, when we accessed logging activity which was new for the anomaly-detection system, we implemented bespoke anomaly alerts for these activities. This highlights that the different operational needs of organisations are reflected in the diverse datasets that we found across them. Therefore, it is critical for any detection system to be able to adapt to the operational environment in which it is deployed.

The fourth and final week involved configuring a bastion machine, testing that the code and parsers worked as intended, and placing the machine into the facility of the organisation. Once this process was concluded, the organisation provided the data or access to the data and the trial commenced. We should note that for all the organisations once data was received, it was then stored locally on the hard drive of the machine, allowing to reconfigure the tool's parameters and re-run the system without waiting again for data to be retransmitted. We

visited the premises of the organisations on a weekly basis to check the results of the system and to discuss these with the security analysts of the organisations. The insights acquired were then used to refine the configuration of the CITD tool and introduce changes if needed. When physical access was not possible, we were provided with remote connection to a bastion host, and we had continuous communications with their analysts. We received ethics approval from the University of Oxford for these trials and we signed Memorandum of Understanding (MoUs) with all three organisations. MoUs detailed how we would use the data anonymously, abiding to privacy regulations, whereas it was clearly stated that results of the detection system do not provide evidence of malicious behaviour and cannot be used to incriminate any of the employees.

4.2. Processing period

For a period of approximately five months, we visited our partners on a weekly basis in order to validate the anomalies generated, and refine the configuration of the system. It is worth noting that the data never left the organisation, so these visits were necessary to access the results. The methodology followed aimed at validating the results obtained for each organisation (intra-organisation methodology), and also, where possible, at comparing the results between organisations (inter-organisation methodology).

The hypotheses distilled from relevant literature that we intend to examine with our experiments across the three organisations are:

- Running PCA with more dimensions yields a smaller number of alerts
- Real data exhibits deviations not observed in synthetic datasets and higher thresholds are required to keep the number of alerts manageable
- Anomalies with a higher number of different types of features reduce the number of false positive alerts
- The presence of L1 alerts is usually correlated with the presence of L2/3 alerts
- Thresholds for L1 alerts vary depending on the context in which organisations operate
- The performance of the system has a linear relation to the number of data processed
- The memory usage has a linear relation to the amount of data processed

4.2.1. Intra-organisation methodology

Elaborating on the hypotheses presented in Section 4.2, we experiment with different configurations of the CITD system with the aim of elucidating the soundness of our anomaly detection method. In particular we focus on:

- How to reduce the number of false positives while increasing the number of true positives. This is not always possible, due to the lack of ground truth in the data.
- How to identify which policies may be more indicative of insider-threat activity and which may be deemed unhelpful due to the large number of users who violate them (and are presumed to not be indicative of malicious behaviour)
- How the performance of the system may be improved, in terms of memory consumption and time for processing the data

Focusing on the reduction of false-positive alerts, we construct anomalies with different numbers of features for every run. We analyse the results, consult the cybersecurity experts of the organisations regarding the number and type of false-positive alerts, and reconstruct the anomalies for another run based on this feedback.

We observed during our tests with synthetic datasets [6] that increasing the number of different types of features in an anomaly (i.e., features describing file activity, email activity, web activity) reduces the number of generated false-positive alerts. There is a compromise, however, in designing these anomalies, because we observed cases where a real attack was missed; thus there is a need for more specific anomalies to capture nuanced attacks. These types of anomalies contain features of the same type (i.e., number of files accessed, number of files written etc.). We also seek to understand whether the presence of multiple anomalies with a relatively low deviation value (these are captured as L3 alerts) may be a better indicator of malicious activity than the presence of a single anomaly with a high deviation value (“red” L2 alerts). Moreover, we experiment with different thresholds to determine what is the optimal way to automatically classify the severity of alerts. This comes from an observation that real data exhibits much greater deviation than synthetic datasets where the commonly used threshold of three standard deviations suffice [6].

Regarding policy violations and tripwires (yielding L1 alerts), we run the system with all the policies for which the requisite data is available and try to determine which of these policy violations may be indicative of insider activity. In addition, we try to understand how to gain insights from combining policy-violation alerts with the presence of L2/L3 anomalies flagged by the detection system. We have created anomalies comprised of features which are policy violations indicative of organisational commitment (activities in out-of-office hours), indicating how many times and how often users violate policies. The number of policy-violation alerts also allows us to conclude if some policies are regularly violated by many users. Such policies should then be reviewed by the organisation, to either be judged unhelpful and removed, or to be modified to conform with the daily practice of the employees. Alternatively, users can be trained to become aware of the importance of complying with the security policies which are neglected and often violated.

We also constantly monitor and try to improve the performance of the system, in terms of memory and speed demands. We observed during the first trial [6] that the number of features which constitutes the anomalies and the reduction of dimensions in the PCA algorithm are decisive factors for the performance of the system. An increase in the number of features comprising an anomaly raises the time required for the system to process the data superlinearly, but at the same time reduces the number of false positive-alerts, as more and different information is captured. We investigate how to reach a fine balance in the trade-off between the accuracy of the detection system and the performance of the detection engine.

Finally, we evaluate the validity of the alerts generated and in distinguishing users who present a potential insider-threat behaviour. We consult security analysts of the organisations to explore the alerts generated and identify if they could be indicative of potential insider behaviour or if they were false positives. Given the high number of alerts generated due to the extremely high number of employees per organisation, we could not explore all of them. Therefore we focus on the alerts that are rarer or with higher deviations, together with a random sample of alerts, as it would happen in a real setup. When possible, we look at alerts generated by specific anonymised individuals that are suspicious of insider behaviour (in cases where malicious information was created on purpose in the dataset), to explore if our system generates any alerts for them and if these are communicated to analysts in a useful and identifiable manner.

4.2.2. Inter-organisation methodology

Another aspect of our methodology focuses on comparing the results between the organisations. Since all three organisations operate in different contexts, comparing the results from all the organisations may determine to what extent the environment within which organisations operate (and the data that they make available) influences the effectiveness of the detection system. Literature suggests that organisations experience at least four different types of insider threat [1,13,14,28,39]. We seek to shed light into how security culture and

the functionality of an organisation may render certain anomalies and the existence of specific policy violations more important than others. Establishing which alerts are crucial for the detection of specific attacks is an important step towards the design of a successful detection system.

Firstly, we investigate whether certain policies, which may be deemed critical in one organisation, are still relevant in another organisation. More specifically, our aim is to implement the same security policies and examine the number of policy violations generated by the system across all three organisations. When there exist significant differences in the total number of policy violations (adjusted to the number of employees) across the three organisations, conclusions about the security culture of an organisation and the suitability of these policies may be formed.

Aside from policy concerns, and focusing on alerts generated due to deviation in employees' behaviour from the norm, we explore whether certain alerts which are identified in the intra-organisational methodology, are indicative of a specific attack in every context. We hypothesise, based on the findings from our initial work [6], that certain anomalies are indicative of a specific type of insider activity (i.e., IP theft, sabotage, fraud). We validate, whether the environment may alter the manner in which attacks are executed, and thus change the effectiveness of these anomalies in indicating such attacks. We also apply the same reasoning to anomalies which generate false-positive alerts, and explore whether the context in which organisations operate may provide a larger or smaller number of false positives (due to differences in employees' activities, which may be the cause of variation in the data). We should note that these latter comparisons are feasible only when similar log data is available.

5. Trials: Tool experimentation results

This section presents our findings from the trials. For each organisation we describe the data which was available, the different configurations of the tool that we experimented with and the results we obtained. Lastly, we reflect on the research objectives and hypotheses described in Section 4 both for intra- and inter-organisation methodologies.

Three commercial organisations participated in our trials. We will name them A, B and C due to confidentiality agreements. All organisations provided data from their systems which traced back to activities of their employees. We used pseudo-usernames to track the activities of the same user throughout the different logs provided. Organisations did not maintain a structured record of the positions held by employees that we could use to build role profiles, so we opted to assign all users into the same role, aiming at finding behaviours that are unique across the organisations. As expected, due to the wide variety of employees' activities, we did not find any interesting observations to report on this. Similarly, organisations did not provide with organisational information that could help in extracting individual, sociotechnical or psychological traits, mainly because they do not capture such information in a digital structured format. Therefore, all features extracted are related to technological indicators. We combined them to infer user behaviours related to insider threat.

To ensure that the baseline of each user profile have data for an equal number of days, the system kept track when new users would appear in the dataset. In many cases, organisations hired employees whose activities started several days within the indicated training period. We ensured that the detection phase for every profile would start only when the number of days indicated in the training parameter are reached. Therefore, detection results for some employees appeared later. We also, distinguished user activities between working days and weekends.

Given that we applied our system on three real environments, the ground truth was not known as it is the case in synthetic datasets. The lack of ground truth rendered statistical rigorous analysis for the validation of our results impossible. Therefore, we relied on more qualitative analysis by examining our alerts with security analysts from the organisations and where possible we complemented this evaluation with statistical inputs.

More specifically, all three organisations differ in the their understanding of insider threat activity in their environments. The first organisation had not experienced attacks from insiders, the second had a clear understanding of past attacks and created use cases of interest, whereas the third one were suspicious of certain employees' activities. To validate our results we opted for three different methodologies. For Organization A, in which there were no known cases of insider activity, our validation approach was to consult their security analysts to gain feedback on the alerts that we generated. Organisation B had created use cases of insider activity in which employees performed acts to insert specific target scenarios of insider threats into the network data. Our approach was then to determine if our analysis could identify the malicious behaviour in these scenarios. For Organisation C, in which there was no ground truth but interest in detecting suspicious insider activities such as sabotage or exfiltration, our validation approach relied on feedback from security analysts to validate if the alerts generated were indicative of insider threat behaviours of interest.

5.1. Organisation A

Organisation A is a multinational organisation offering IT services globally. Organisation A used the SIEM tool Splunk[1] for monitoring their network activity and implementing security policies. Splunk is used to store and query data, log events of interest, as well as perform statistical analysis with focus on external threats. Organisation A lacked, however, a system able to detect insider threat activity. To their knowledge, they had not experienced any insider incidents. However, they were concerned with some increased activity they witnessed from employees based outside UK regarding the use of USB drivers. They also raised concerns regarding their employees' activity on encrypting and decrypting sensitive files.

5.1.1. Data sources

During our first meetings we identified and requested the appropriate type of logs and data collected by the detection system, and we also enriched our L1 policy library by examining their internal security policy documentation and consulting their security analysts. Organisation A provided full access to Splunk that allowed us to explore the events logged and their format. They also highlighted the logs which capture information critical to the functionality of the organisation. We decided to focus on *email exchange logs, removable devices logs, main repository logs* and *file encryption logs*. Our decision was informed by the fact that all these logs capture important events for the organisation and contained user identifiers; unlike logs such as those capturing web activity, which record only an IP address and further triangulation would be required to link a specific activity to a particular user (we could not obtain access to DHCP logs to perform this task). Also, organisation A have strict control access on web activity. All websites should be white-listed before employees are allowed to access them. Unfortunately, to our knowledge, instances where access to a particular website is blocked are not recorded. Such information would have been rather useful for identifying employees who violate security policies regarding web activity.

Initially, organisation A provided a small sample of logs to facilitate the process of implementing parsers for the detection tool. Once the parsers were in place and a lengthier period of data was requested, organisation A experienced restrictions in obtaining the data from their SIEM provider. They were obliged to request specific permissions in order to retrieve a three-month period of data, however, even with the additional permissions difficulties arose due to the Splunk API's

[1] https://www.splunk.com/

Table 1
Details on the logs acquired from organisation A.

Type of log	Size	# Records	Data length
Encryption	30 MB	206k	5 months
Email	1.7 GB	10M	5 months
USB Device	5.9 MB	74k	5 months
Repository	37.8 GB	197M	5 months

limitation on the size of events.[2] We created Python scripts to recursively query their SIEM, paginate the queries and retrieve data for the lengthiest possible time period. Table 1 details the size, start date and length of the date acquired. Unfortunately, Splunk limited the queries to the past 90 days Therefore, we were able to obtain data spanning over a period of five months (90 days and two months of collecting live data).

We restricted the number of fields per log that are queried from Splunk to those identified as useful for insider threat detection. Logs stored by Splunk comprise of several fields, some of which are duplicates while others provide information about the employee (i.e. phone number) that was deemed irrelevant for the detection system. Downloading the desired number of fields resulted in obtaining much less data, with a decrease of around 70% in volume. This practice also reduced the time needed to retrieve the logs to 1 h, and 8 days in case of the repository.

We wrote parsers for all the different types of logs and created adequate anomalies based on the information the logs contained. We had first to do some additional formatting to match usernames with email addresses and to extract multiple file names and paths in single event logs.

5.1.2. Investigation into L1 anomalies and tripwires

Organisation A granted us with access to internal documents detailing the security policies which employees comply to. They had implemented via their SIEM tool, named Splunk, a number of these policies and had logged events which constitute a violation of these policies. Unfortunately, due to the complexity of the SIEM tool, the analysts were not necessarily aware of what constitutes a violation of a policy. As an example, they had implemented rules to identify events of brute-forcing passwords; however, when we asked how many times someone must insert the password wrongly in order to trigger an alert, they could not provide an answer. We were not given access to logs which indicated a violation of policy.

The policies implemented in the detection tool were specially crafted based on the available datasets and the information we acquired during our informal interviews with the staff of organisation A. The rationale behind the aforementioned policies was to detect possible data exfiltration attempts. The policies we were able to implement, given the absence of VPN and web logs, are summarised in bullet-points below:

- Exceed threshold (4 MB) of emails with big attachments.
- Exceed threshold (300) of number of emails with big attachments sent to unusual email addresses.
- Exceed threshold of file decryptions (300) and send an email with a size more than 4 MB to a personal email address (e.g. gmail, yahoo, aol, ...)

Fig. 2 shows that we generated a total of 10,273 alerts for 1166 users during 140 days, i.e. 73 alerts per day on average. From those, 10,078 were unique for the day and user, i.e. the user generated a single alert that day; revealing that it is uncommon to generate more than one alert per day. Specifically for each alert, we generated 6215 alerts for decrypting files, 3797 alerts for sending big emails and 261 for sending emails to new addresses. Regarding the users that generated more than one alert per day, they generated a big size email alert or a decryption of files alert combined with a new email destination (an email sent to a new recipient of an external email address). No users generated more than two alerts per day. Focusing on the email alerts, we detected that many were generated by company's automatic emails (i.e., no-reply@organisationA, alerts@organisationA), thus we further processed data to discard these emails.

The user with most alerts generated 23 alerts within 120 days, with a maximum value of 35,4230 policy violations. Only 13 alerts were generated for decrypting more than 10,000 files, belonging to 10 different users. We should also note that when examined in detail, the same user raised a policy violation of 5,1000 files being decrypted in a day. We were able to identify that during the same days for which we raised policy violations for this employee, alerts from the detection system regarding deviant decrypt activity, as well as email activity, were triggered. We validated the significance of our findings with analysts from organisation A, who where surprised about the large number of file decryptions and were worried that they did not spot that before. After further investigation they concluded that the user had been copying files to different hard drives for legitimate purposes. Their key reason for not being alerted by this unusual behaviour was the fact that the files were not sensitive. A possible enhancement to our tool would be to keep track of sensitive files and discard alerts that contain only the downloading or decryption of non-sensitive files. Unfortunately, we did not have access to such information. The alerts generated by the other 9 users were also false positives, as only non-sensitive files were handled. The analysts pointed that their behaviour was rather unusual, as well. In fact for some, it seemed as if they had just started working (despite the fact that they were long time employees).

Table 2
Details on the different configurations of the tool for organisation A.

Data length	Training	# PCA	Time	Memory	Users
5 months	30 days	6	180 m	<4 GB	4330
5 months	60 days	6	180 m	<4 GB	4330

5.1.3. Usability and utility assessment of L2/L3 anomaly-detection algorithms

Our assessment of the validity of L2/L3 alerts focused on testing and optimising the different parameters of the detection system. Therefore, we tested the system with different training configurations and examined the occurrence of scalability limitations given the large dataset we had at our disposal.

We ran the system with four different training periods, as depicted in Table 2 which also presents the configuration options. Due to longer training periods, the testing data was reduced accordingly. As expected, running time,[3] memory usage and number of users which logs exist for are the same since they are independent of the duration of the training period. The running time is less than three hours for a dataset spanning over five months, and no scalability issues due to the volume of data or the number of users were observed.

Table 3 provides an overview of the number of anomaly alerts generated by the detection system. We can observe that the average number of alerts per day is very similar in both cases. The number of alerts per day may look overwhelming in the first instance, we should note that for testing purposes we created a much bigger number

[2] Splunk returns a maximum number of 500,000 events per query.

[3] For longer training periods there should be a small difference in the required time to finish processing all the data. However, during the trials, due to the limited time we had available at the premises of the organisation we run in parallel multiple configurations of the system. As a result, any small differences in the running time could not have been detected.

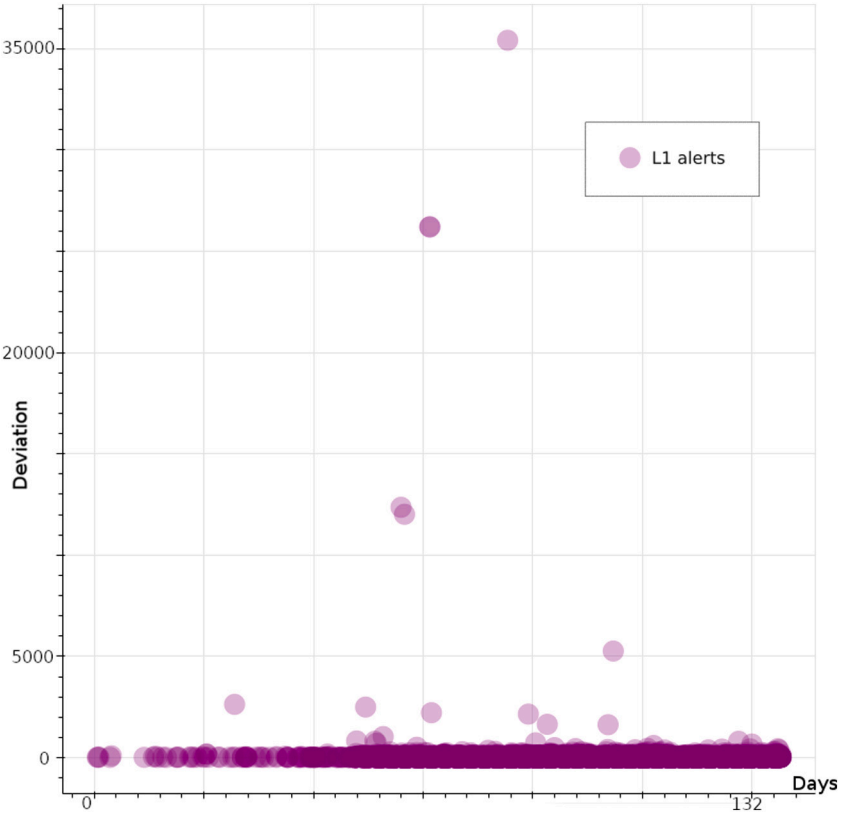

Fig. 2. Policy violations generated for organisation A over a period of 5 months.

Table 3
Alerts raised for organisation A with different training periods.

Training	Orange severity	Red severity	Blue (L3)	Total	# Users	# Days	Alerts/Day
30 days	5698 (22.8%)	14840 (59.6%)	4413 (17.7%)	24951	3284	70	356
60 days	2782 (21.9%)	6646 (52.3%)	3273 (25.7%)	12701	2439	35	362

of anomalies than the usual. More specifically, for every activity, we implemented anomalies based on activities that had never been observed before, activities that are observed frequently and a mix of both, resulting in 27 different anomalies. We introduced an "out-of-office hours" anomaly, in an attempt to capture behaviours which may indicate organisational commitment or anxiety. A list with the number of anomalies and the generated alerts can be found in Appendix.

In order to investigate the anomalies we began from considering the statistical data. We identified the users who generated the largest number of alerts and used visualisation tools to identify those who created the largest deviation for anomalies. We mainly focused on anomalies which indicated deviant behaviour in many actions, as well as, aggregate activity. We randomly selected a few more to ensure that the remaining alerts with very low deviation values are false positives.

When the system was configured to run over a 30-day training period, we started our analysis from the user alerts to identify those with the highest deviation. Fig. 3 illustrates all the user alerts the system raised over a period of four months. Fig. 4 presents all the alerts for the user with the highest user activity score. We were able to identify that this high score was due to decrypting file activities and the vast majority of alerts were related to decryption activity.

We also focused on the user who generated the most alerts during a period of four months. Fig. 5 presents all the alerts raised in the system. Overall, there were 54 different alerts. In stark contrast to the employee with the highest score, this person raised a variety of alerts in a time interval of two days. More specifically, they raised email, file activity and device alerts, which is behaviour that seemed related to data exfiltration. Analysts from organisation A investigated these alerts further and were deemed as false positives. However, this behaviour was related to an unusual working schedule for the employee.

Finally, we tried to combine information obtained from L1 alerts. We searched for L1 alerts generated for a user with the highest value of L2 alerts, however, there were no policy violations identified for this individual.

In a similar vein, we processed the data for which a training period of 60 days was used. We should note that the user who generated the highest number of alerts was not the same person as the employee in the 30-day training period. Upon further investigation, the majority of the alerts raised for this employee were still in the training period for the 60-day configuration run.

We started our analysis using a 60-day training period to identify those users with the highest deviation. Fig. 6 illustrates all the alerts the system raised over a period of four months. We focused on the user who generated the most alerts during a period of four months. Fig. 7 presents all the alerts raised by the system for this user. Overall, there were 37 different alerts in a short time interval. The alerts raised mainly revolved around email and file activity.

We also explored the alerts generated by the user with the highest user activity score (Fig. 8). Similarly to the previous analysis, this high score was due to decrypting file activities and the vast majority

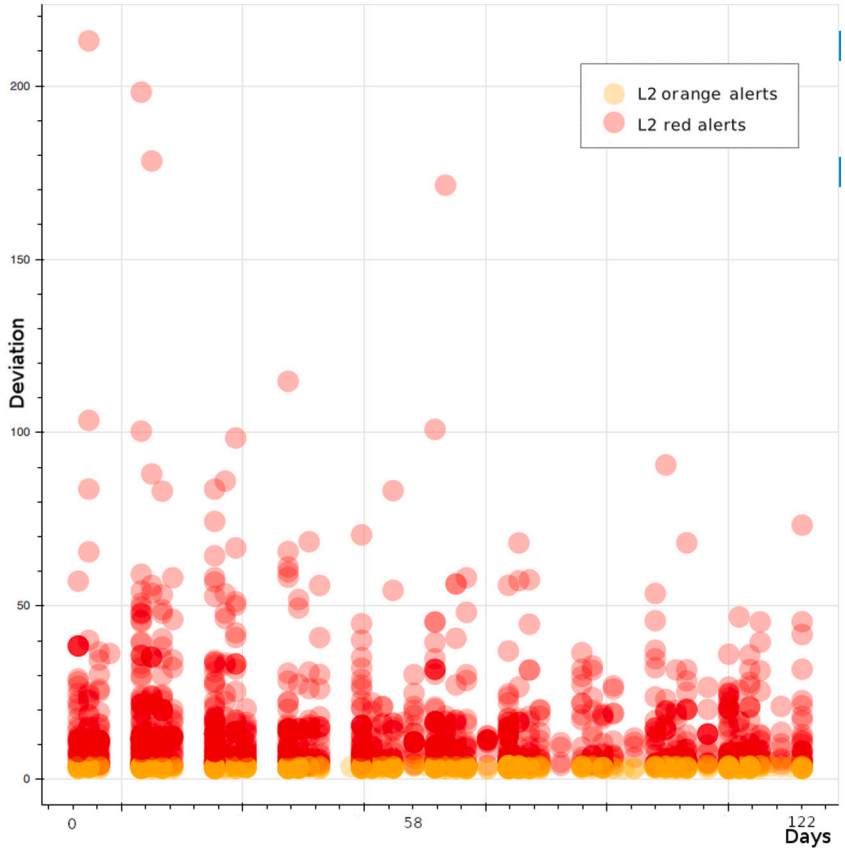

Fig. 3. User alerts for 30 days of training.

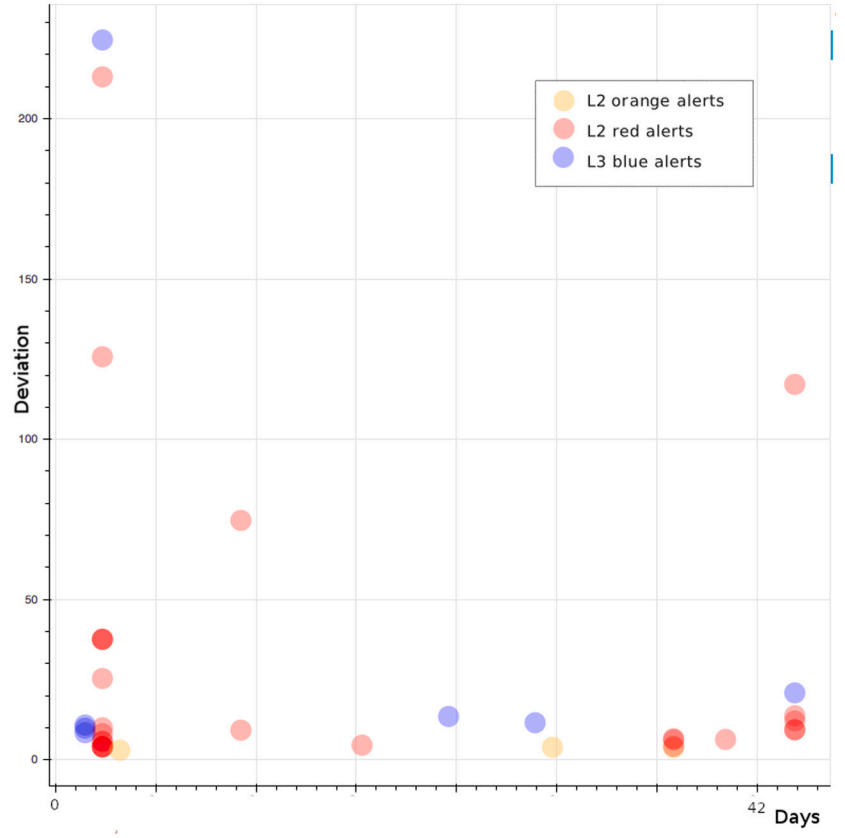

Fig. 4. User's activity with the highest value of deviation over a period for four months with 30 days of training data.

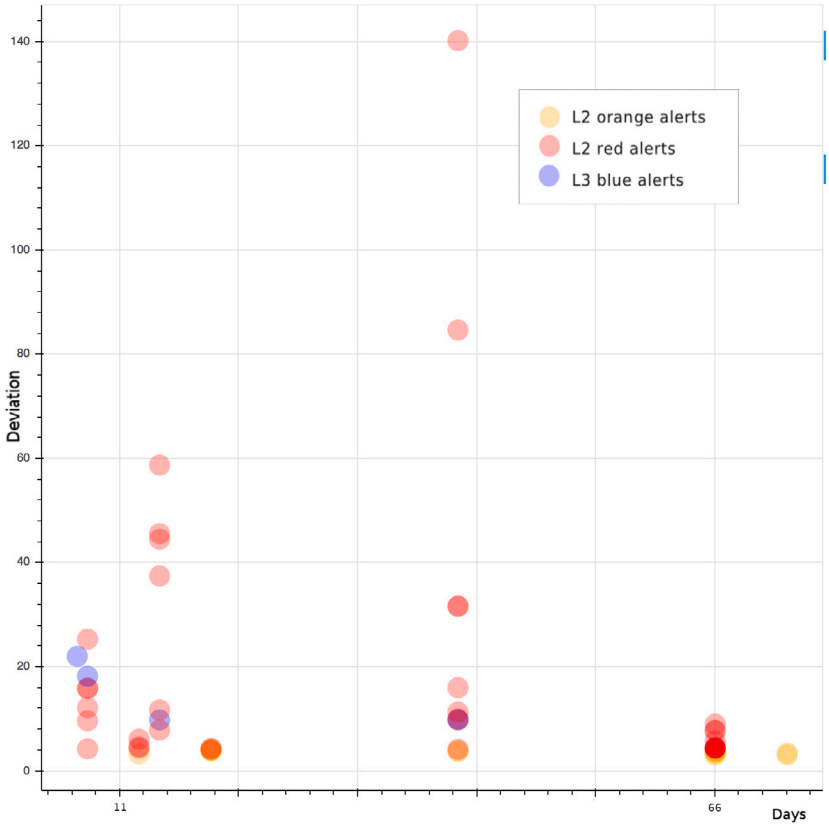

Fig. 5. User with the highest number of alerts over a period of four months with 30 days of training data.

of alerts, included the user activity alert, were related to decryption activity. Fig. 9 shows the policy violations for this user. As illustrated, there is a decrypt policy violation, which is consistent with the alerts raised by the system on decrypting files.

We also focused on the person identified in Section 5.1.2 for raising the highest number of policy violations regarding decrypting files. As seen, the system has generated alerts regarding decryption activity for this individual (see Fig. 10).

From the results presented above we can confirm that the presence of L1 alerts is correlated with the presence of L2/3 alerts. Also, the number of alerts with higher deviation values generated during the trial is much higher than these generated with synthetic dataset, providing evidence that real data create deviations not observed in synthetic datasets and higher thresholds are required to reduce the number of alerts. Upon consultation with the security experts of organisation A, the user who raised the highest number of different alerts, was recently hired. The activity captured in their logs appeared as if they were inactive for a large period of time and suddenly started working on their daily tasks. They could not verify if this behaviour was malicious or why the user did not have activity before.

Overall, organisation A considered the alerts generated by the system of utility for event triage, and although no alerts were linked to insider activity, they found them worth further investigation.

5.2. Organisation B

Organisation B is a multinational organisation operating globally in the energy sector. To our knowledge, they did not possess a system dedicated for insider threat detection, however, they expressed an interest in purchasing one. In particular, they were running an internal Insider Threat program where they evaluated several detection tools available on the market, with the aim to adopt the most effective one. To that end, they created 8 different scenarios with real data from their SIEM tools. Instead of adding malicious information, they acted while appearing to work normally as insiders, imitating previously observed attacks. The attacks corresponded to different insider threats.

Organisation B experienced insider attacks in the past mainly from disgruntled employees, aiming to sabotage the function of the organisation. They had particular interest in detecting cases of radicalised employees who may be motivated to sabotage the company's infrastructure.

5.2.1. Data sources

Organisation B provided a sample dataset comprised of 13 types of logs. These logs were CEF formatted and all contained usernames. We wrote the parsers for all log types and configured a machine that was installed in their premises. Because of physical access restrictions, we were provided with remote access to the machine.

The types of logs we had access to, differ greatly from the other organisations, as they do not represent the activities of the users, but alerts generated on users' activities by their security tool. Specifically, we were provided access to these logs: *Crosslink*, *Crowdstrike*, *CWS proxy*, *DNS*, *FireEye*, *IP data*, *Juniper NSM*, *McAfee EPO DLP*, *McAfee EPO HIPS*, *McAfee EPO VirusScan*, *Print logs*, *SourceFire*, and *Windows events*.

We acquired data from organisation B for a period of 4 months. All data obtained from organisation B linked logs with usernames and no further triangulation was required. Due to huge volume, data was send to our system overnight to avoid disruption in their networks. Each week of data weighted approximately 550 GB before cleaning, that reduced to 60 GB/week once fields of interest were extracted. Each original log contained approximately 50 field names, we informally met with staff from their security personnel to understand the information provided in these fields and decide which data to parse into the tool.

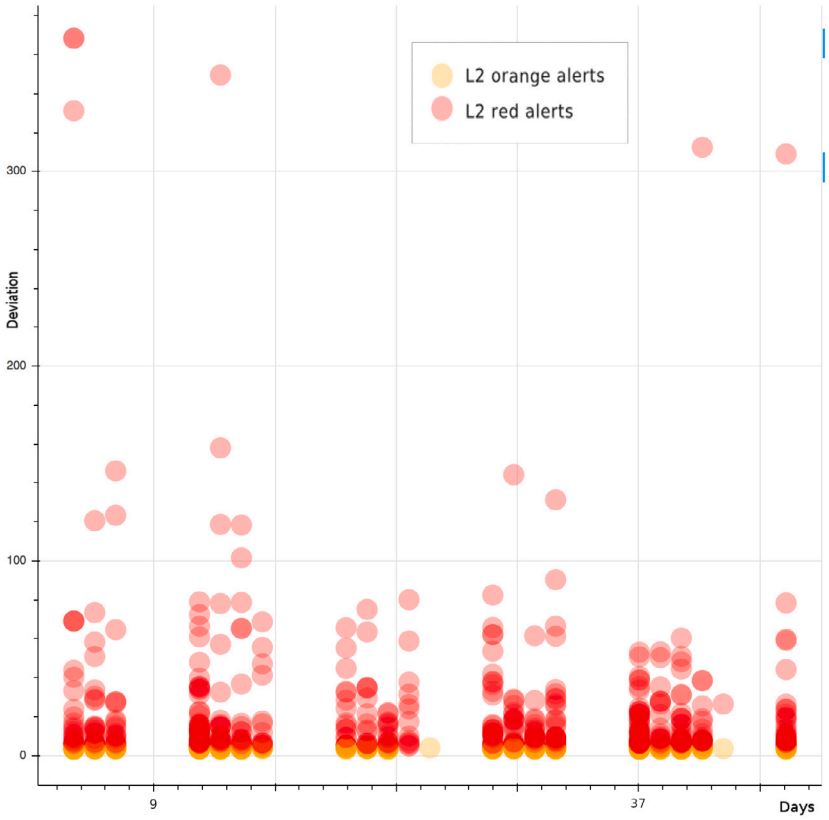

Fig. 6. User alerts for 60 days of training.

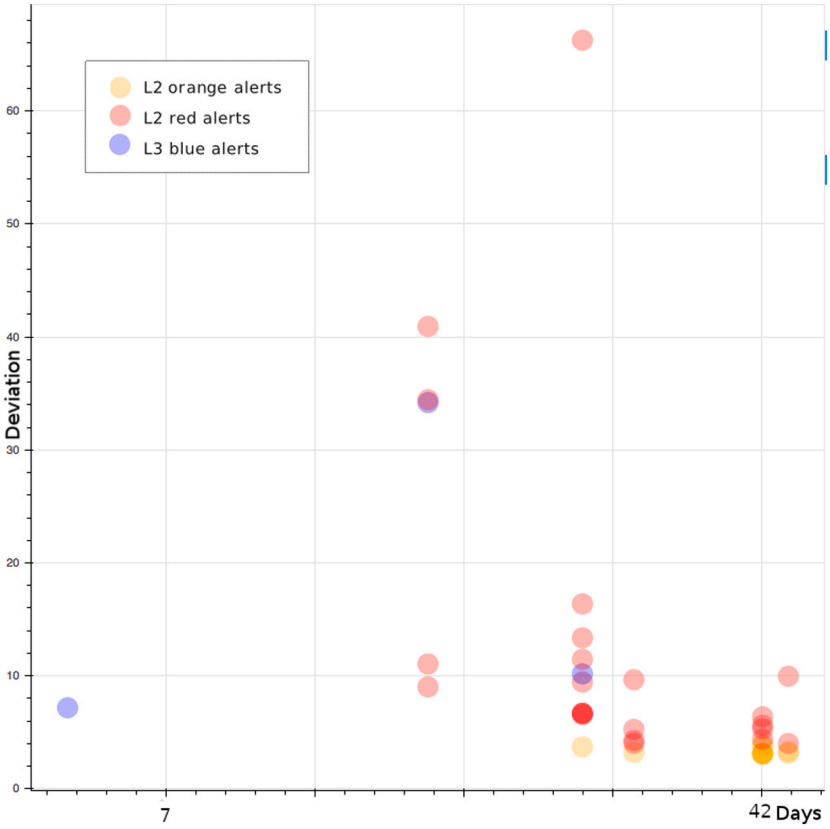

Fig. 7. User with the highest number of alerts over a period of four months with 60 days of training data.

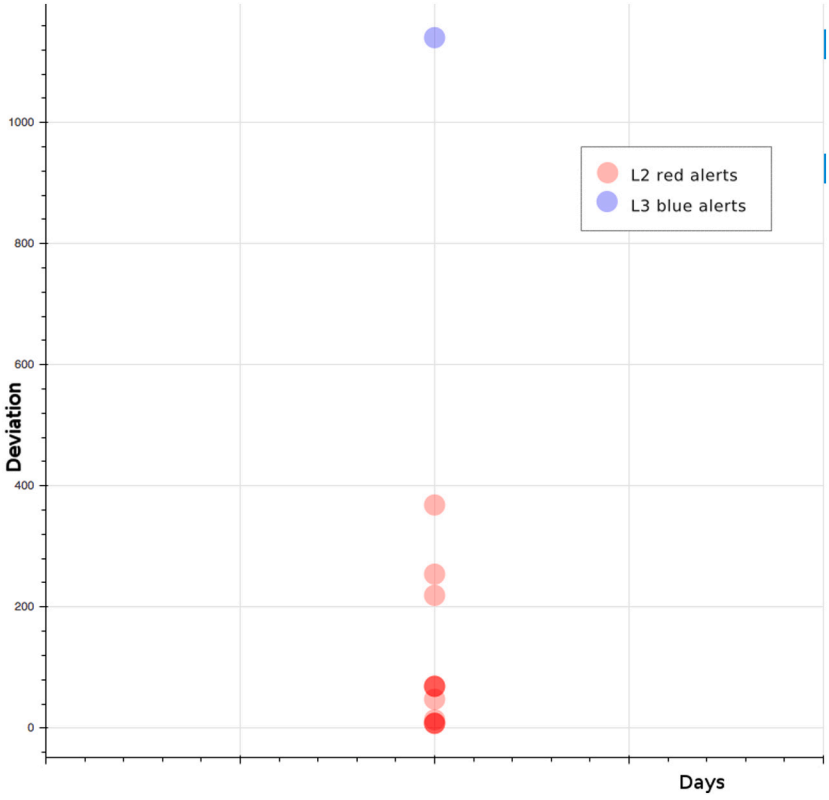

Fig. 8. User's activity with the highest value of deviation with 60 days of training data.

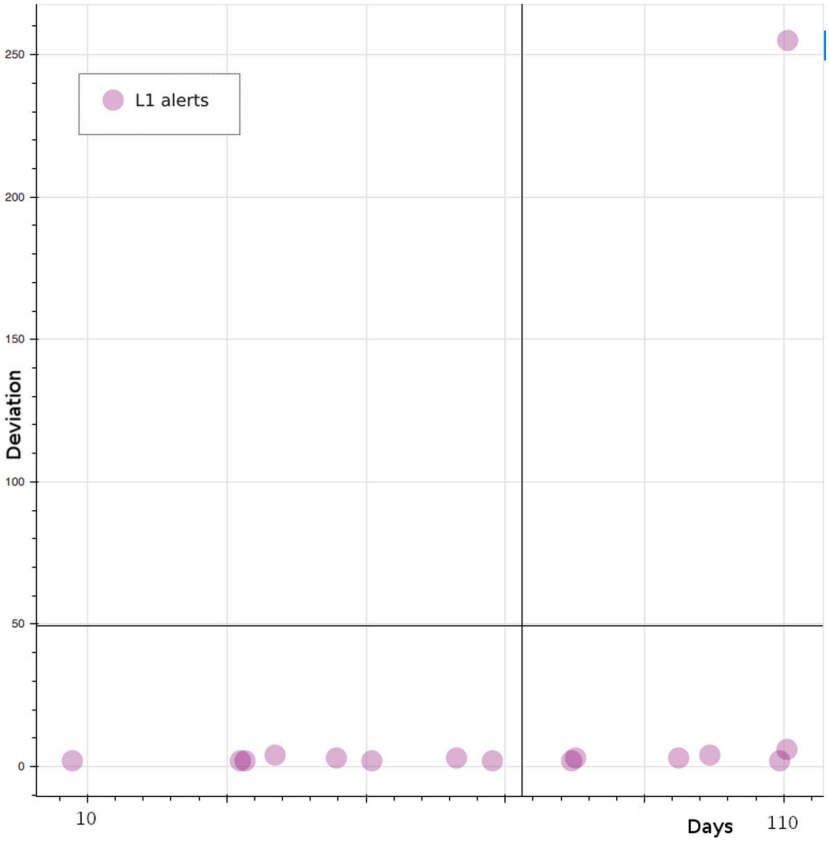

Fig. 9. L1 alerts generated by the user with the highest value of L2 alerts with 60 days of training.

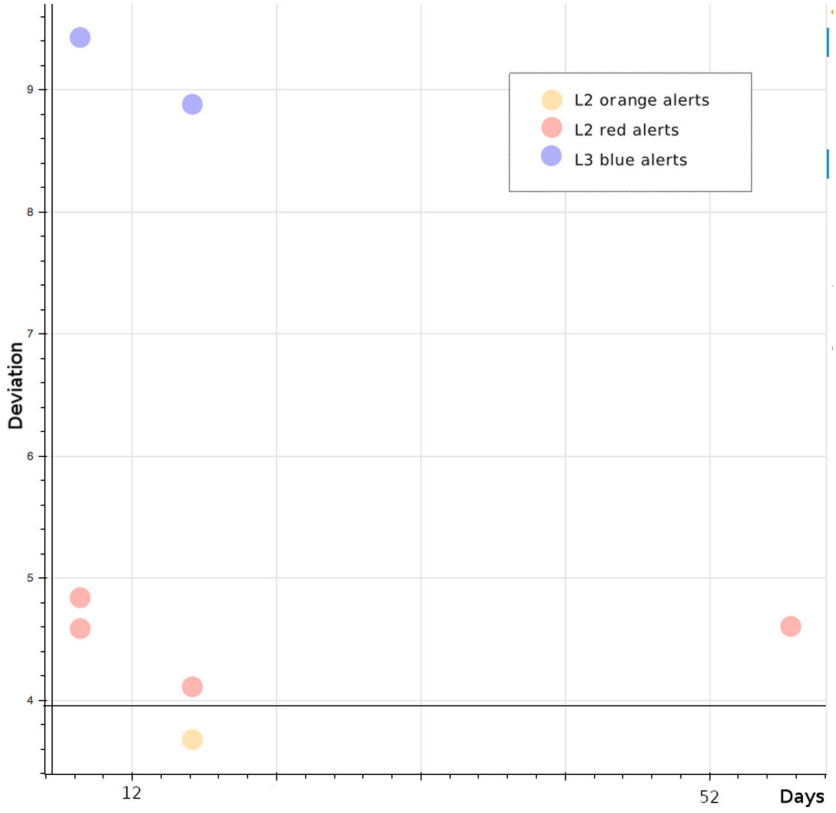

Fig. 10. User's activity with the highest value of L1 alerts with 60 days of training data.

Table 4
Alerts raised for organisation B for a 60 days training period.

Training	Orange severity	Red severity	Blue (L3)	Total	# Users	# Days	Alerts/Day
60 days	50,996 (24.1%)	137,848 (65.1%)	22,810 (10.8%)	211,654	19,786	57	3,713

5.2.2. Investigation into L1 anomalies and tripwires

Organisation B had implemented a number of security policies via a SIEM tool and they were not interested in exploring this space further. We were not provided with access to their internal security policies, nor with logs of when these policies are violated.

5.2.3. Usability and utility assessment of L2/L3 anomaly-detection algorithms

To perform the anomaly detection, we implemented anomalies that were specifically designed to tackle the use-cases that the organisation B created, but we also designed generic anomalies corresponding to insider behaviour identified in the literature that we also implemented in the other organisations.

We identified 19,786 unique usernames corresponding to valid users across all logs from Partner B. We trained the system for a period of two months and run the detection algorithm on the rest of the dataset. Overall, we generated a large number of alerts which is not surprising given that we implemented a big number of unique anomalies. Our decision to create a big number of anomalies was informed by the fact that we had to identify features that better indicate suspicious behaviour, but at the same time create alerts for the use-cases of organisation B.

Table 4 provides an overview of the number of anomaly alerts generated by the detection system. Over a period of 57 days, we generated 211,654 alerts, of which 50,996 were of medium severity (orange) and 137,848 of high severity (red). Fig. 11 shows these alerts. In addition, we generated 22,810 alerts that indicate overall suspicious behaviour (blue) of which 19,000 indicated events that were observed for the first time (i.e. a user accessing a new file repository).

Fig. 12 shows the alerts generated for the anomalies that are specific for the use-cases. In this case the number of alerts dropped significantly to 21,584. Of those, 30.3% were of orange severity and 69.7% were of red severity (Table 5). We validated these alerts and we found that one of the real insiders from one of the test cases generated several alerts of high severity for different anomalies. This insider was the only user that generated more than two alerts of high severity (easily identifiable in the visualisations) in the same day, totalling 5, which was confirmed as a real insider.

5.2.4. Performance

Executing the CITD anomaly detection system took two hours and used <30 GB of memory to process four months of data and 19,786 users. We conclude that no scalability issues were experienced by the system. We generated a large number of blue (L3) alerts for this experiment. We attribute it to the large number of anomalies created and the variance of the data, which means that the majority of blue alerts were generated by highlighting activities that were observed for the first time.

We were able to identify one insider in the eight use-cases that organisation B created. Further investigation revealed that the data related to the undetected cases was not included in the data provided, or the incident was included in the training data, making their detection impossible. As there were no similar incidents in the test dataset, we were not able to examine the effect of including malicious data in

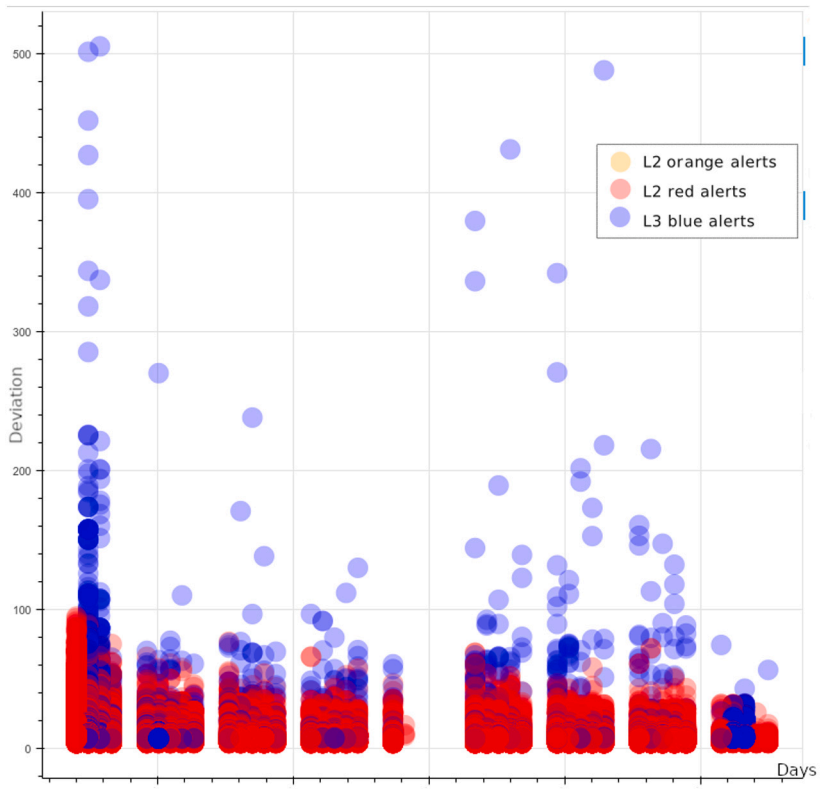

Fig. 11. All alerts generated during the trials with organisation B. Each block of alerts correspond to a natural week. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

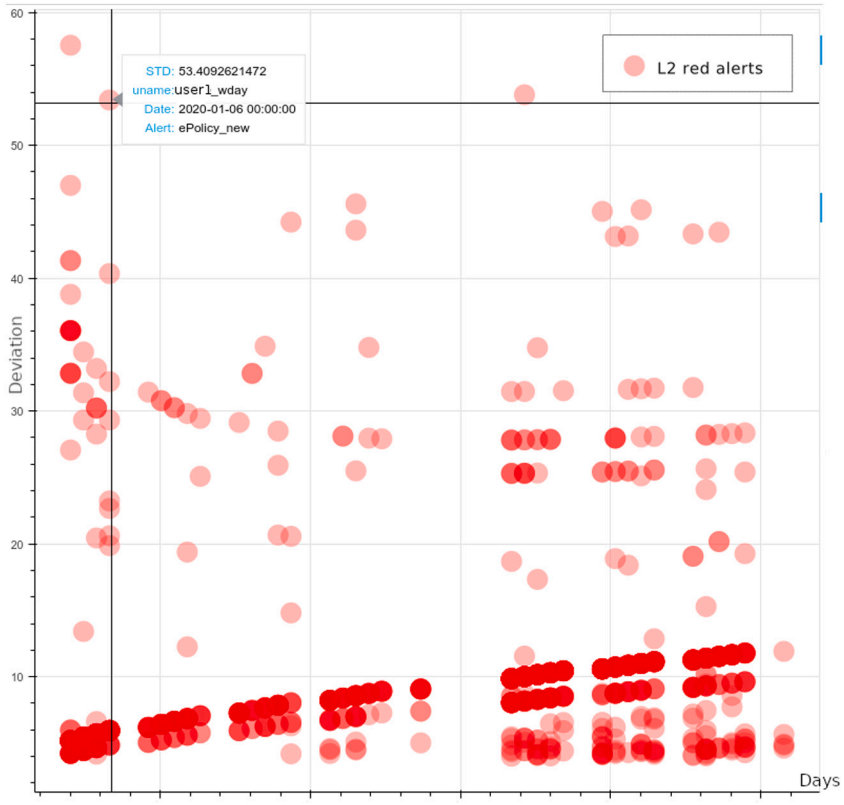

Fig. 12. Alerts generated that tackled the specific use-cases of organisation B. Alerts generated during a natural week form column-like clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Alerts raised for organisation B for the specific use-cases.

Training	Orange severity	Red severity	Blue (L3)	Total	# Users	# Days	Alerts/Day
60 days	6,538 (30.3%)	15,046 (69.7%)	– –	21,584	19,786	57	378

Table 6
Details on the logs acquired from organisation A.

Type of log	Size	# Records	Data period
Email logs - server 6	17 GB	73M	3 months[a]
USB Device	4.2 GB	12M	2 months[a]
Email logs	103.3 GB	201M	3 months[b]
Proxy logs	10 GB	50M	3 months[b]

the training dataset, and the extent to which such data would inhibit detection of these events.

5.3. Organisation C

Organisation C is a multinational organisation operating globally in the energy sector. Organisation C utilised ArcSight for monitoring their network activity and implementing security policies, while they had outsourced parts of their network activity to be monitored by a third party. ArcSight was used to store events and investigate cases which were flagged as suspicious external activity. To our knowledge, they did not possess a system for insider threat detection. They had experienced insider attacks in the past, mainly from disgruntled employees aiming to sabotage the functions of the organisation, either by corrupting information or by committing fraud. They had also experienced two incidents of insider activity where employees tried to exfiltrate sensitive data.

5.3.1. Data sources

During our first meetings we identified and requested the appropriate type of logs required by the system. Unfortunately, we were not granted access to their internal security policies. Within the first few weeks we agreed with organisation C to get access to: email exchange logs, removable devices logs, proxy logs and DHCP logs (those logs were crucial to triangulate an IP address from the web logs with a user identity). In the end we did not get access to DHCP logs because they were managed by a third party that imposed restrictions in accessing them.

Table 6 details the logs we acquired, the corresponding time period as well as the overall size. We should note that for the email logs organisation C used six different servers to store the data. From those, only two months of email activity overlapped the same period for device activity logs (USB)[a], and the other email logs overlapped the proxy logs[b]. Proxy logs did not contain a username, only an IP address, and provided no useful information, since the field containing the URL indicated only if a search engine was used.

We wrote parsers for all three type of logs and created adequate anomalies based on the information the logs contained. We had to format the usernames of the logs to match system usernames and email addresses. We did so by extracting the username from the email addresses and converting following guidance from the security analyst. It is also worth mentioning that email logs were provided in a number of zip files and further processing was required to merge them into a single file, as they were overlapping in time.

Furthermore, organisation C provided logs directly from their SIEM tool, ArcSight in a binary format. They stated that the logs contain information for the following events: *Windows Member Servers*, *Windows Domain Controllers*, *ESX*, *Linux*, *Juniper FW*, *Juniper SSL VPN*, *Cisco ASA*, *CheckPoint*, *BlueCoat*, *RACF*, and *SourceFire*. However, the lack of clarity on the content of those files and the fact that they were in a proprietary-binary format, made it impossible for us to retrieve the desired information.

Table 7
Details on the configuration of the system for organisation C.

Data period	Training	# PCA	Time	Memory	Users
2 months	30 days	5	157 m	3 GB	33542

5.3.2. Investigation into L1 anomalies and tripwires

For organisation C, we implemented three policies related to the use of email. The first one raises an alert when a user sends more than one email with an attachment greater than 5 MB. The second policy violation is triggered when a user sends more than 1000 emails in a single day. This threshold was determined after experimenting with lower thresholds that generated a big number of L1 alerts. The last policy is flagged when the tool detects that an email with a size greater than 5 Mb has been send to a personal email address, such as gmail, yahoo, etc. We run the system once with these policies against the email logs provided, but unfortunately we were generating many L1 alerts. This was clearly a problem of tuning of the thresholds, as we did not get an input from the company, and relied on parameters from Organisation A.

An interesting observation is that the thresholds we tried on organisation C were much higher than the thresholds we implemented the policy alerts for organisation A. Nonetheless, we generated a much bigger number of policy violations for organisation C compared to those generated for organisation A. This is a clear example why policies have to be implemented considering the particularities of each environment.

5.3.3. Usability and utility assessment of L2/L3 anomaly-detection algorithms

We decided to commence the trial for organisation C by running the system with the email logs and USB logs. Table 7 presents the details for the configuration of the system. Overall, the system processed approximately 20 GB of data in 157 min, consuming 4 GB of memory. The number of unique usernames the system created profiles for was 33,542.

Table 8 provides details on the number of alerts generated by the system for a period of 30 days. Although the number of alerts generated on average per day may seem overwhelming, there is a large number of users for whom an alert has been generated for. To further elaborate on the number of alerts, the ratio alerts per day over the number of users is very similar to that observed for organisation A.

Fig. 13 illustrates the overall number of alerts generated. From this picture it is obvious that some users exhibited a significant deviant behaviour. Fig. 14 shows the alerts the system generated for the user with the highest deviation over a period of 30 days. All the alerts relate to device activity, some of which was observed during outside office hours.

We should note that due to the network structure of organisation C, it is not evident if the time when the event occurred corresponds to the local time, where the user is based, or corresponds to the time where the server capturing the event is based. Therefore, we cannot be certain that these activities occurred in out-of-office hours. It may well be the case that the employee exhibited this behaviour during working hours, but the events were recorded from servers which are based in different parts of the world. We were not given access to data to correct this issue.

Fig. 15 presents the alerts raised for the user with the second highest deviant behaviour. As with the previous user, seven alerts were raised

Table 8
Alerts raised for organisation C during a run of the system.

Training	Orange	Red	Blue	Total	# Users	# Days	Alerts/Day
30 days	14,236 (18.9%)	32,540 (43.3%)	28,437 (37.8%)	75,213	29,492	30	2,350

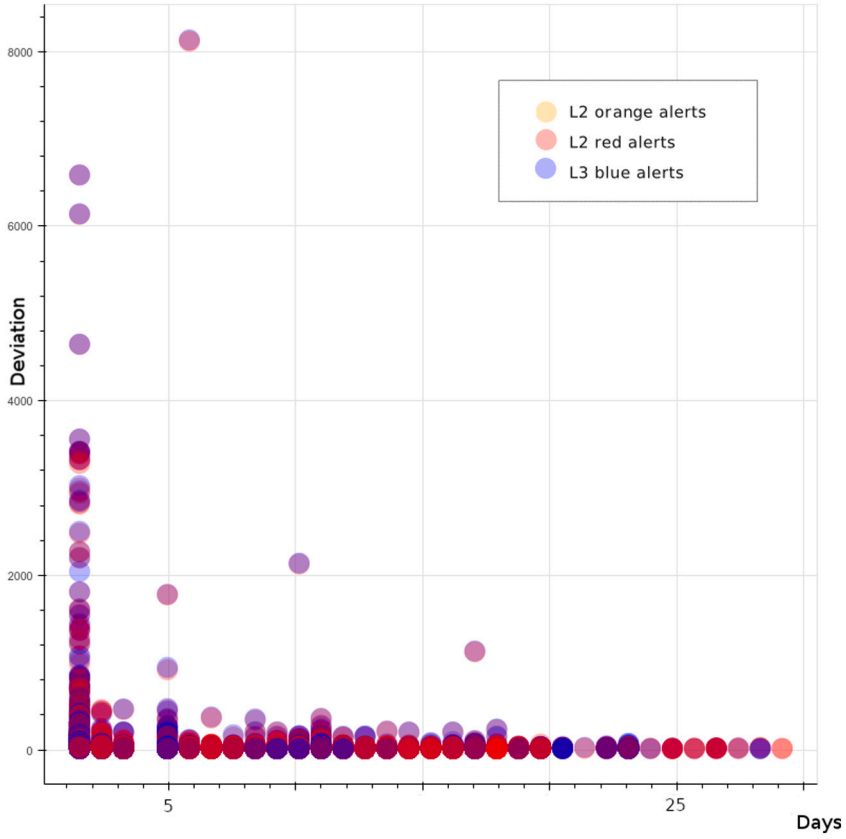

Fig. 13. L2/3 alerts generated during the run for organisation C.

for this employee, all of which for device activity both for office and out-of-office hours.

Finally, similarly to organisation A, we emphasised on the employee who generated the highest number of alerts. Fig. 16 illustrates the alerts raised for this individual. In total eleven alerts were triggered all of which relate to device activities.

Unfortunately we could not validate our findings due to lack of ground truth. Organisation C did not allow security analysts to provide feedback on the anomalies we produced due to privacy concerns, however, the analysts considered them worthy of further investigation. In fact, analysts mentioned in early meetings that they had concerns about usage of USB drives by employees in other countries, a concern that is matching our findings.

5.4. Reflections on results across all trials

For organisations A and C the performance of the system is very efficient. For a similar volume of data, the system requires roughly the same time to process all the data and produce alerts. In both trials, we used less then 4 GB of memory even though the number of users was extremely high. For organisation B, we used much more memory due to ingesting much more data, while the execution time was proportionally less as the number of anomalies implemented was lower. Based on our findings we can confirm the hypotheses below:

- The performance of the system has a linear relation to the amount of data processed
- The performance of the system has a linear relation to the amount of users present into the system
- The performance of the system has a polynomial relation to the number of features fed into the anomalies
- The memory usage has a linear relation to the amount of data processed
- The memory usage has a linear relation to the number of dimensions in the PCA algorithm

Regarding the process of analysing the results, in organisations A and C we mainly focused on those who generated either the highest number of alerts or exhibited the highest deviant behaviour. It is an approach that when combined with visualisations and access to the raw data, may quickly identify the reason why these alerts occurred and if this behaviour is indicative of insider threat activity. For organisation B, we were able to identify the employee who purposefully performed insider activity in the only scenario that we were given access and contained malicious behaviour.

Despite the large amount of alerts the system generated, the use of visualisations and the tuning of parameters to decrease the sensitivity of the system can reduce rapidly the number of false positive alerts. Furthermore, we identified features for each organisation that exhibit less variation in the training period and better capture the abnormal activities. By reducing the number of anomalies and the number of features that comprise these sets we have another strategy to reduce false positives.

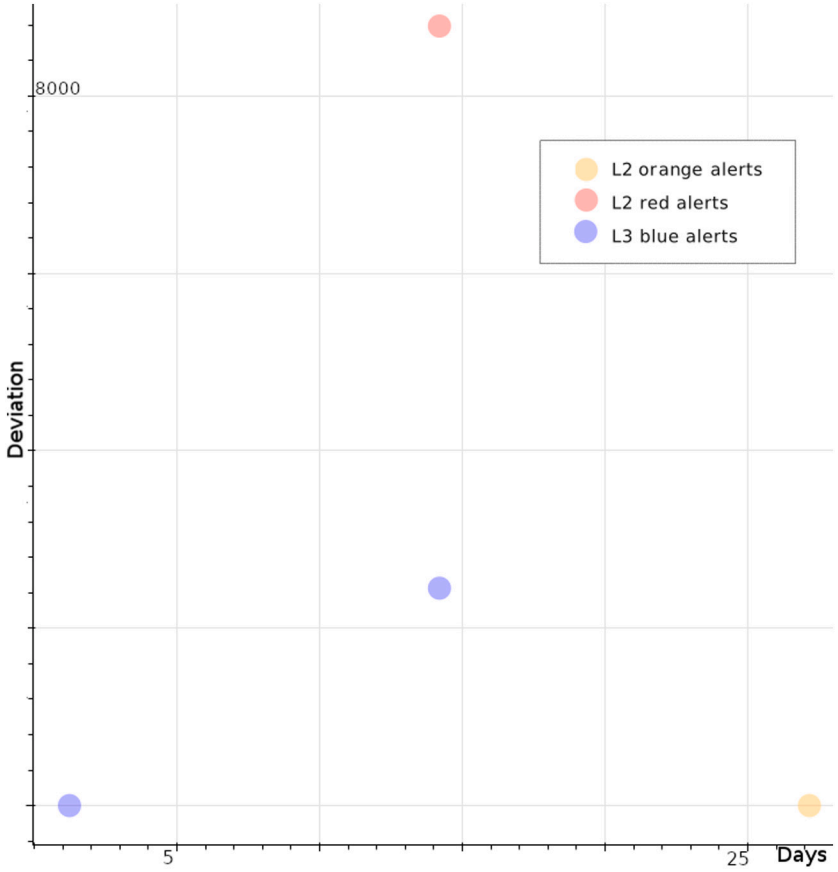

Fig. 14. User's activity with the highest value of L2 alerts.

Finally, to briefly comment on the L1 alerts, we initially tried the same thresholds for policy violation in organisations A and C. We observed that a significantly larger number of alerts was generated regarding email activity. We can safely conclude that it is in the culture of employees in organisation C to send many emails daily which potentially are larger in size. Therefore, there is some initial evidence to confirm our hypotheses that thresholds for L1 alerts vary depending on the context of each organisation. This is particularly true when organisations have different data available, as it is the case for organisation B.

6. Lessons learnt

The process of obtaining access to organisations and testing the detection system revealed invaluable lessons regarding the scalability and performance of the system, as well as for the deployment of the tool and its integration with existing SIEM tools. We faced difficulties in obtaining the appropriate datasets and in formatting the data. These difficulties provided an opportunity to understand the problems which organisations face when deciding which events to be logged and helped us to check the integration of SIEM tools into CITD system.

6.1. Validation of scalability for the anomaly-detection algorithms

Focusing on the improvements in the scalability and performance of the tool, we are still to observe a failure of the system due to shortage of memory or storage space. On the contrary, we were able to verify the improvements in performance and scalability that we observed when testing the tool with the synthetic datasets; at no point during the trials did the system use more than 30 GB of RAM, and that was the case after ingesting 1.5 TB of logs from organisation B. Note that this setup was running the tool in a standalone machine, but the execution could be distributed in several machines as there are no dependencies with the anomaly calculations for users and/or anomalies. Therefore, we do not expect any limitations that impede implementation in other organisations.

Elaborating further on the performance of the tool, we were able to validate our theoretical hypothesis for the complexity of the L1, and L2 algorithms, as well as for the algorithms which construct the tree-profiles for each user. The complexity of the system was deemed linear to the number of users and the number of data logs and polynomial to the number of features which comprise the anomalies. We observed that in all the configurations of the system there was a constant increase in the required time to process the data for a single day[4] and calculate the results of the anomaly detection. The constant increase in the amount of time indicates a linear cost for the time required to run the system for a day with the amount of days we have ran the system thus far. Therefore, a possible strategy to further improve the performance, is to store the state of the system (statistical data for profiles, some relevant historical information etc.) and create a cut-off point once a specific number of months have passed, deleting data beyond the cut-off point. The system will loose information related to judging new activities (i.e., some observed file access activity may be considered as new), however, it will not reach a point were the processing power will not suffice.

We further observed that when the number of features which are fed into the anomalies increased, the increase on the time required by the system to process the data was significantly higher, indicating a polynomial relation. For the datasets which contained data on more users and were much larger in volume, the constant increase in time

[4] We make a reasonable assumption that on average the amount of data the system processes per day is similar.

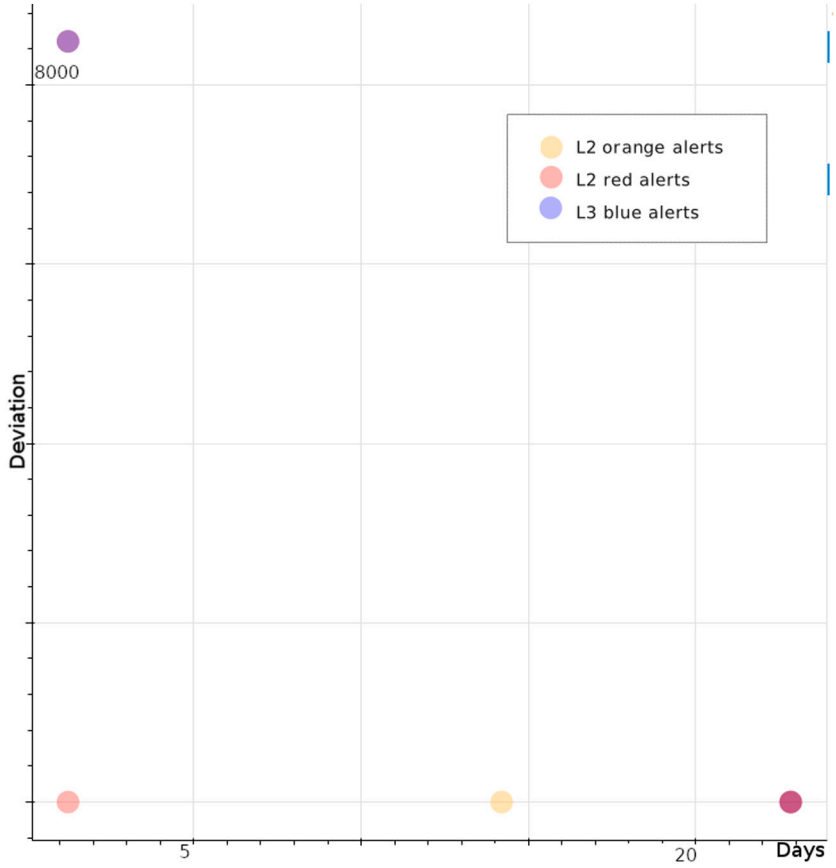

Fig. 15. User's activity with the second highest value of L2 alerts.

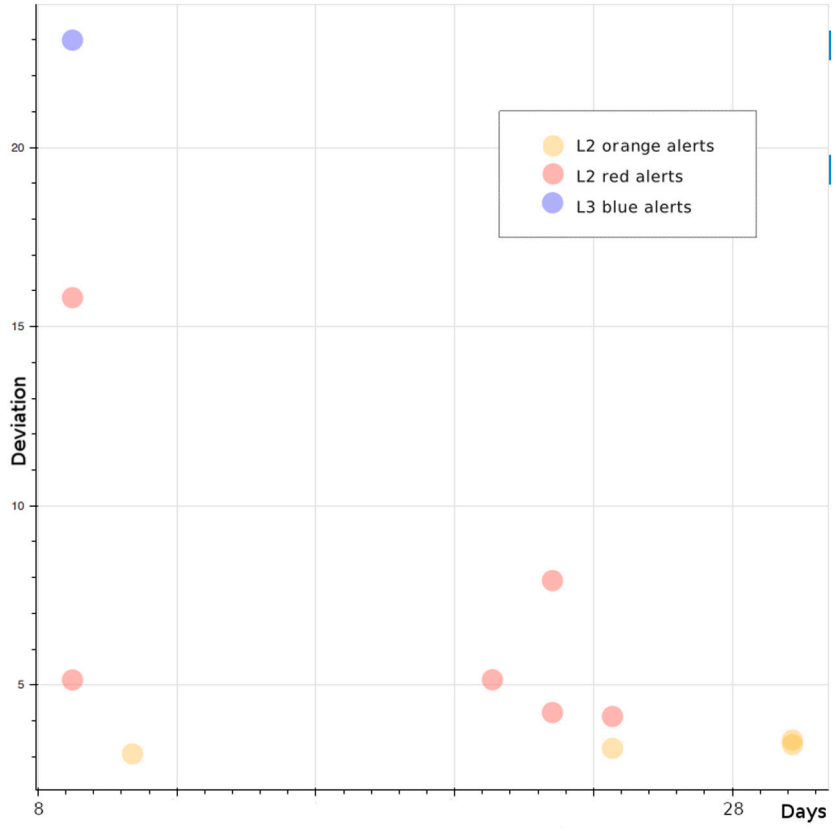

Fig. 16. User's activity with the highest number of L2 alerts.

required by the system to process data for a single day was not significantly higher than the interval required for the small datasets. In a similar vein, the time required to process data for the first day of the detection system was higher in the big datasets but not polynomial, indicating a linear relation between the time required to process data and the volume of data.

Finally, the algorithms building and accessing the user profiles, albeit linear in complexity in terms of time required per day to process the data, are a significant factor for the performance of the system. The system performance increases when the profiles are small in size. This observation is amplified when the system is constructing role-profiles[5] that are significantly larger than the user-profiles. We can safely assume that the complexity of the PCA, which in essence is reflected by the number of features which make up an anomaly, is the only factor which may increase polynomially the time required by the system to process data and produce results.

Regarding memory usage, we were able to verify our hypothesis that it is linear in the amount of data available and in the number of anomalies produced by the system. In a similar vein to time performance, memory usage is influenced by the size of profiles in a linear manner, due to the fact that these are stored in memory to speed up data access when calculating features. Finally, there is a linear relation with the number of users for whom the system creates profiles. We should note that some information, such as URLs, is stored in the profiles in hash form. This effectively allows the system to use less memory, while maintaining the same functionality.

6.2. Availability of data

During the trials, we experienced difficulties in obtaining the appropriate data for insider threat detection. The origin of the problems we faced was two-fold; the current monitoring practices which organisation adopt, and the permissions required from various authorities due to the sensitivity of the requested data.

Elaborating on the current monitoring practices, organisations mainly focus on configuring SIEM tools to detect external threats. In all organisations, the vast majority of the logged events revolved around the use of IP addresses, lacking a link to the individuals who were assigned these addresses. In a similar vein, events from Intrusion Detection Systems (IDS) were referring to devices or IP addresses. As a result, it is not straightforward to create a profile for an individual, a strong prerequisite for our anomaly detection system. In order to provide further information for the poorly-linked with individuals' events, triangulation of different types of log activity is required.

A common practice in industry is to outsource the monitoring function to third parties with expertise in IT services, or to use off-the-shelve tools to log, store and process events. The problem we encountered was that the data was not available to the organisations and permission to obtain the requested logs was always sought from third parties. In one case the organisation was requested to pay the security vendor to acquire access to their data. In addition, in cases where tools were used to log the events, the data was provided in a binary format creating difficulties in processing it. From our trials, we can identify a lock-in effect that organisations experience when contracting a security vendor to monitor their systems. It is in the interest of these vendors to prohibit organisations from processing security-related data outside the vendor's environment.

The majority of the logs provided by the organisations comprised more than 150 fields, most of which contained identical information in different formats (i.e., timestamp which the event occurred in more than one format). Identifying information relevant to insider threat detection, as well as parsing data formatted in complex ways proved to be a time-consuming task. Most fields were deemed redundant and from each type of log six to seven fields were processed by the system. Distinguishing the number of fields which are redundant is critical to the performance of the system, since the amount of data to be processed is reduced dramatically, boosting the speed of the training and detection period. From the acquired fields, there were logs which were well formatted and easy to write the parsers for, but, logs capturing file activity were not that well structured. This is due to the fact that in many logs, more than one attribute was captured in a single field, and this determined the existence of content from other fields. Missing information in logs raised additional issues and reduced the number of features which the system could produce. This was more evident in file logs, where occasionally the type of the action was not included in the log.

Departing from technical challenges, the sensitivity of the requested data required permission from several authorities within the organisation. Organisations had to provide information to the Human Resources departments, explain the purpose of the trial and ensure that at no point any of the employees would be under investigation. Similar information was provided to the union of employees. Additional constraints and time delays occurred due to the privacy legislation. All the organisations were multinational organisations and data from employees who were not in the UK was included. This proved problematic for one of the organisations, however, the manner in which their network architecture was structured did not allow them to discriminate the data for UK employees easily. Eventually, they received permissions from the appropriate authorities.

In order to overcome privacy concerns we were provided restricted access to personnel data, resulting in limiting the ability of the tool to generate role anomalies. Finally, one of the organisations was in a merge and acquisition process and they were restructuring their IT infrastructure. This resulted in additional delays due to the fact that it was imperative for the employees who were to start working for the new organisation not to be included in the dataset provided.

6.3. Limitations on tripwires and L2/3 alerts

The main research objectives regarding the tripwire alerts were two-fold; to enrich the library of security policies and validate whether their implementation may shed light into suspicious insider activity when combined with alerts from the detection system. Regarding the first objective we were given limited access to the internal security policies. All three organisations possessed systems which monitored for different types of security violations, thus their interests laid in the implementation of the anomaly-detection system. Nonetheless, we managed to enrich our library of policies, especially from our interaction with organisation A.

Regarding the second aim, the implementation of tripwire alerts depends on the availability of the appropriate data. As we explained in the respective sections, only a small number of policies were expressible with the logs that we were provided. In addition, the organisations filtered the web activity of their employees without, however, logging when they attempt to access black-listed sites. This filtering policy made the implementation of a number of web policies in our L1 library ineffective.

To reduce the number of false positives, implementing the same policies in organisations A and C highlighted the importance to customise detection thresholds determining when to trigger a policy violation in specific environments. In this case it was an effect of organisation's culture, but other factors, such as different business purposes or busy periods towards the end of fiscal year, can also affect the detection precision. It is therefore important to monitor and tailor detection thresholds in each environment to reduce the number of alerts to a desired and acceptable level. Another example that emphasised the importance to implement policy alerts was a user in Organisation A

[5] We should note that these profiles are not used for role anomalies. They enable the system to establish if an activity is new not only for the user under question but for any employee who has the same role.

generating a policy violation with value greater than 35,000 for decrypting files without being originally spotted by the security analysts. Documentation and implementation of these types of alerts will avoid activities going unnoticed by the security team.

Additionally, we were able to identify security policies that improve the accuracy of the detection system when combined with specific types of anomalies. For example, for organisation A, the number of violations regarding decryption of specific files, when combined with types of anomalies that aimed in identifying IP theft, yielded valuable results. The coexistence of policy violations and L2/3 anomalies for the same individuals and time frames allowed us to focus on actions that were more intriguing to the security analysts and provided a better understanding on the anomalies in employees' behaviour that the system was reporting.

Finally, validating the effectiveness of the L2/L3 alerts which form the core of the anomaly detection system is a very complex task. The greatest challenge that we faced was the lack of known insiders. Although organisations B and C experienced insider attacks, we were not provided datasets that contained known malicious behaviour. Therefore, our findings had to rely on verifying the alerts generated with the security analysts of the organisations.

Due to ethical reasons, we were not interested in verifying that individuals had actually committed insider attacks. We focused on identifying if the alerts that the system generated corresponded to abnormal, not necessarily malicious, behaviour. For organisation A, we went through generated alerts with their security analysts, specifically focusing on the user with the highest number of alerts and the user who triggered most policy violations. Analysts agreed that the alerts represented users who exhibited strange behaviour or act in unusual manners. For organisation B, we were able to spot the intended malicious activity that one of the security employees performed with relative success. For organisation C, we were not given any feedback on the detection by the security analysts, although it was pointed out that the types of alerts generated were related to some of their security concerns at that time.

In order to reduce the number of false positives we interacted with security analysts in every run of the system and incorporated their feedback both on types of features used to calculate the anomalies and the thresholds. More specifically, the system allows the analysts to create tailored anomalies by combining a number of features. We first discussed with the security analysts to understand what types of behaviour were observed as suspicious in the past and how insiders would attack the organisation. Therefore, we created tailored anomalies for all three organisations. We ran the system a few times to calculate the values for every feature, and if the feature would skew the results (minimal activity that did not provide extra information but exacerbated any minimal activity of this type), it would be removed from the set of features comprising the anomaly to reduce the number of alerts generated. Furthermore, the system provides, alongside any type of alert, a deviation scoring. We then combined the different types of alerts, their frequency and deviation over a period of time into L3 alerts, which represent deviation of the anomalies. We discussed with the analysts our initial results for Organisation A and C (for B we had the ground truth), identifying that most suspicious behaviour occurred when an employee would generate an usual number of different types of alerts over a short period of three days or when an employee would constantly raise a small number of low-valued deviations over an extensive period of time (for more than 7 days). Therefore, it is rational to use all alerts generated in L2 to feed L3 anomalies, but set an acceptable threshold for L2 anomalies to not overwhelm analysts with anomalies that are recurrent.

The results suggest that despite the relative success of the system in generating true positive alerts for abnormal behaviour, the system provided a large number of alerts in total. As explained in Section 5, we purposefully created a large set of anomalies to test how representative are of specific suspicious behaviour and identify features that exhibit less variation in the training dataset. This exercise skewed the number of alerts that the system generated. Despite this, we believe that for large organisations with more than 20,000 employees generating around 300 alerts per day (as in our trials) is a reasonable amount of alerts for analysts to process. Especially for users who produce only one alert over a long period of time, (which covers the largest amount of alerts) these alerts can be considered as less suspicious.

We asked analysts of all three organisations about the utility of the visualisations provided. They all agreed that although they are fairly simple, they provide a straightforward way to identify alerts, as by hovering over the dots they get information such as username and type of the alert, which provides them a starting point for an investigation. Having alerts sorted by deviation (i.e. dots higher in the plot are alerts with higher deviation) also helped them to prioritise the alerts, although final human inference is still needed for a quick triage.

Our experiences from the trials has enabled us to identify specific features and anomalies tailored to each organisation, in order to reduce the number of alerts. Furthermore, we have identified that using visualisations on the data of the alerts is a good strategy to reduce the number of alerts an analysts needs to consider without compromising the effectiveness of the system. We have further seen that L3 alerts, the blue alerts, are a practical way to triage high number of alerts, as in most of the cases when a user was generating several L2 alerts, we had also an L3 alert. Remember that blue alerts can detect an unusual number of alerts over a period of several days, so they have the capacity to detect users that are hidden by "common" alerts that would go unnoticed by an analyst. By performing anomaly detection over these anomalies, we are able to detect when the deviation of the anomalies is even more anomalous. Although these results are quite promising, we believe the number of cases observed are not enough to draw some firm conclusions on the value of L3 alerts. As future work, we plan to further investigate the real utility of L3 alerts and better understand their limitations.

7. Conclusions

Insider threat is a heavily researched topic, given the impact that these attacks may have for organisations. Literature consists of several detection tools that have been validated on synthetic datasets but scarcely on real environments. To address this gap, we presented the results and the challenges we experienced when deploying our detection system into three large organisations.

Our results suggest that the system is scalable and able to consume the amount of data generated by large organisations and that the visualisations can assist analysts in identifying suspicious activity indicative of insider threat. Considering the alerts that were generated during the trials, we found that it is of paramount importance to consider the organisation's characteristics in the implementation of an insider threat system. Generic policies and anomalies generate large number of alerts that, in most cases, do not correspond to a real threat for an organisation. The creation of policies and anomalies to tackle specific organisations' needs is, however, a difficult task that should involve the study of practices of risk for the organisation.

We highlighted the difficulties that organisations experience to investigate cases suspicious of insider threat due to legislative (privacy) and technical limitations. Such difficulties are especially exacerbated in the case of large organisations that process huge volumes of data, which are not always well maintained and formatted. Finally, we provided useful insights for organisations that wish to deploy detection systems in their networks.

It is still early days to report on any stakeholder appreciation, given the fact that we deployed and used the tool for six months. We were able however, to reinforce our belief that there is a gap in the industry in terms of detecting insider activity with the use of anomaly detection algorithms. All three organisations welcomed our contributions and noted that they have started to invest in systems with machine learning functionality able to build profiles for users.

CRediT authorship contribution statement

Arnau Erola: Data curation, Methodology, Formal analysis, Investigation, Software, Validation, Writing. **Ioannis Agrafiotis:** Data curation, Methodology, Formal analysis, Investigation, Software, Validation, Writing. **Michael Goldsmith:** Conceptualization, Methodology, Funding acquisition, Formal analysis, Supervision. **Sadie Creese:** Conceptualization, Methodology, Funding acquisition, Formal analysis, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was conducted in the context of a collaborative project on Corporate Insider Threat Detection, sponsored by the UK National Cyber Security Programme in conjunction with the Centre for the Protection of National Infrastructure, whose support is gratefully acknowledged. The project brings together three departments of the University of Oxford, the University of Leicester and Cardiff University. We would like to thank all multinational companies for their continued collaboration on this project.

Appendix

This section presents the results per type of alert for the 15-day training-period for organisation A. These are:

- Device out-of-office hours: 497
- Blue first observed activity: 2899
- Blue: 1514
- Email out-of-office hours: 221
- File accessed in the past: 1207
- Decrypt for role: 1931
- Encrypt files accessed in the past: 1133
- Email to new addresses: 458
- Encrypt role: 207
- Device used in the past: 1517
- New files accessed: 558
- Overall new activities: 1448
- Decrypt files accessed in the past: 1961
- Encrypt files out of office hours: 88
- Files overall activity: 189
- Decrypt files out-of-office hours: 1451
- Email role: 84
- Email addresses used in the past: 1134
- Encrypt file overall activity: 178
- Email overall activity: 185
- Using new devices: 360
- File out-of-office hours: 314
- Decrypt new files: 1913
- File for role: 258
- Decrypt overall activity: 1148
- Overall activity: 1631
- Encrypt new files: 467

References

[1] Cappelli DM, Moore AP, Trzeciak RF. The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes. 1st ed.. Addison-Wesley Professional; 2012.
[2] Colwill C. Human factors in information security: The insider threat – who can you trust these days? Inf Secur Tech Rep 2009;14(4):186–96.
[3] Shaw ED, Stock HV. Behavioral risk indicators of malicious insider theft of intellectual property: misreading the writing on the wall. Technical report, Symantec; 2011.
[4] Buede DM, Axelrad ET, Brown DP, Hudson DW, Laskey KB, Sticha PJ, et al. Inference enterprise models: An approach to organizational performance improvement. Wiley Interdiscip Rev Data Min Knowl Discov 2018;8(6):e1277.
[5] Caputo D, Maloof M, Stephens G. Detecting insider theft of trade secrets. IEEE Secur Privacy 2009;7(6):14–21.
[6] Agrafiotis I, Erola A, Happa J, Goldsmith M, Creese S. Validating an insider threat detection system: A real scenario perspective. In: 2016 IEEE security and privacy workshops. 2016, p. 286–95.
[7] Legg PA, Buckley O, Goldsmith M, Creese S. Automated insider threat detection system using user and role-based profile assessment. IEEE Syst J 2015;11(2):503–12.
[8] Oladimeji TO, Ayo CK, Adewumi SE. Review on insider threat detection techniques. J Phys Conf Ser 2019;1299:012046.
[9] Kim A, Oh J, Ryu J, Lee J, Kwon K, Lee K. SoK: A systematic review of insider threat detection. J Wirel Mobile Netw Ubiquitous Comput Dependable Appl (JoWUA) 2019;10(4):46–67.
[10] Homoliak I, Toffalini F, Guarnizo J, Elovici Y, Ochoa M. Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. ACM Comput Surv 2019;52(2):1–40.
[11] Moore AP, Cappelli DM, Caron TC, Shaw E, Spooner D, Trzeciak RF. A preliminary model of insider theft of intellectual property. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst; 2011.
[12] Parveen P, Thuraisingham Bhavani. Unsupervised incremental sequence learning for insider threat detection. In: Intelligence and security informatics (ISI), 2012 IEEE international conference on. 2012, p. 141–3. http://dx.doi.org/10.1109/ISI.2012.6284271.
[13] Moore AP, Cappelli DM, Trzeciak RF. The "big picture" of insider IT sabotage across US critical infrastructures. In: Insider attack and cyber security. Springer; 2008, p. 17–52.
[14] Greitzer FL, Moore AP, Cappelli DM, Andrews DH, Carroll LA, Hull TD. Combating the insider cyber threat. Secur Privacy, IEEE 2007;6(1):61–4. http://dx.doi.org/10.1109/MSP.2008.8.
[15] Sarkar KR. Assessing insider threats to information security using technical, behavioural and organisational measures. Inf Secur Tech Rep 2010;15(3):112–33.
[16] Nurse Jason RC, Buckley Oliver, Legg Philip, Goldsmith Michael, Creese Sadie, Wright Gordon RT, et al. Understanding insider threat: A framework for characterising attacks. In: Security and privacy workshops (SPW), 2014 IEEE. IEEE; 2014, p. 214–28.
[17] National Cybersecurity and Communications Integration Center. Combating the insider threat. 2014, URL https://www.us-cert.gov/sites/default/files/publications/Combating%20the%20Insider%20Threat_0.pdf.
[18] Pfleeger SL, Predd JB, Hunker J, Bulford C. Insiders behaving badly: Addressing bad actors and their actions. IEEE Trans Inf Forensics Secur 2009;5(1):169–79.
[19] Schultz EE. A framework for understanding and predicting insider attacks. Comput Secur 2002;21(6):526–31.
[20] Greitzer FL, Frincke DA. Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation. In: Insider threats in cyber security. Springer; 2010, p. 85–113.
[21] Maloof MA, Stephens GD. Elicit: A system for detecting insiders who violate need-to-know. In: Kruegel Christopher, Lippmann Richard, Clark Andrew, editors. Recent advances in intrusion detection. Lecture notes in computer science, vol. 4637, Springer Berlin Heidelberg; 2007, p. 146–66.
[22] Zhang H, Agrafiotis I, Erola A, Creese S, Goldsmith M. A state machine system for insider threat detection. In: Cybenko George, Pym David, Fila Barbara, editors. Graphical models for security. Cham: Springer International Publishing; 2019, p. 111–29.
[23] Eldardiry H, Bart E, Liu Juan, Hanley J, Price B, Brdiczka O. Multi-domain information fusion for insider threat detection. In: Security and privacy workshops (SPW), 2013 IEEE. 2013, p. 45–51. http://dx.doi.org/10.1109/SPW.2013.14.
[24] Nguyen N, Reiher P. Detecting insider threats by monitoring system call activity. In: Proceedings of the 2003 IEEE workshop on information assurance. 2003.
[25] Myers J, Grimaila MR, Mills RF. Towards insider threat detection using web server logs. In: Proceedings of the 5th annual workshop on cyber security and information intelligence research: cyber security and information intelligence challenges and strategies. CSIIRW '09, New York, NY, USA: ACM; 2009, p. 54:1–4. http://dx.doi.org/10.1145/1558607.1558670, URL http://doi.acm.org/10.1145/1558607.1558670.
[26] Senator TE, Goldberg HG, Memory A, Young WT, Rees B, Pierce R, et al. Detecting insider threats in a real corporate database of computer usage activity. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining. ACM; 2013, p. 1393–401.

[27] Rashid T, Agrafiotis I, Nurse JRC. A new take on detecting insider threats: exploring the use of hidden markov models. In: Proceedings of the 8th ACM CCS international workshop on managing insider security threats. 2016, p. 47–56.
[28] Greitzer FL, Hohimer RE. Modeling human behavior to anticipate insider attacks. J Strateg Secur 2011;4(2):25–48.
[29] Magklaras GB, Furnell SM. Insider threat prediction tool: Evaluating the probability of IT misuse. Comput Secur 2002;21(1):62–73.
[30] Brdiczka O, Liu J, Price B, Shen J, Patil A, Chow R, et al. Proactive insider threat detection through graph learning and psychological context. In: Proc. of the IEEE symposium on security and privacy workshops (SPW'12), San Francisco, California, USA. IEEE; 2012, p. 142–9.
[31] Arulampalam M Sanjeev, Maskell Simon, Gordon Neil, Clapp Tim. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 2002;50(2):174–88.
[32] Chen Y, Malin B. Detection of anomalous insiders in collaborative environments via relational analysis of access logs. In: Proceedings of the first ACM conference on data and application security and privacy. ACM; 2011, p. 63–74.
[33] Costa DL, Collins ML, Perl SJ, Albrethsen MJ, Silowash GJ, Spooner DL. An ontology for insider threat indicators development and applications. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst; 2014.
[34] Greitzer F, Purl J, Leong YM, Becker DES. Sofit: Sociotechnical and organizational factors for insider threat. In: 2018 IEEE security and privacy workshops. IEEE; 2018, p. 197–206.
[35] Agrafiotis I, Erola A, Goldsmith M, Creese S. Formalising policies for insider-threat detection: A tripwire grammar. J Wirel Mobile Netw Ubiquitous Comput Dependable Appl 2017;8(1):26–43.
[36] Bishop M, Simidchieva B, Conboy H, Phan H, Osterwell L, Clarke L, et al. Insider threat detection by process analysis. In: IEEE security and privacy workshops. IEEE; 2014.
[37] Legg PA, Buckley O, Goldsmith M, Creese S. Caught in the act of an insider attack: Detection and assessment of insider threat. In: IEEE symposium on technologies for homeland security. 2015.
[38] Agrafiotis I, Nurse JRC, Buckley O, Legg P, Creese S, Goldsmith M. Identifying attack patterns for insider threat detection. Comput Fraud Secur 2015;2015(7):9–17.
[39] Hunker J, Probst CW. Insiders and insider threats – an overview of definitions and mitigation techniques. J Wirel Mobile Netw Ubiquitous Comput Dependable Appl 2011;2(1):4–27.