TWOS: A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition

Athul Harilal, Flavio Toffalini, John Castellanos, Juan Guarnizo, Ivan Homoliak, Martín Ochoa
ST Electronics-SUTD Cyber Security Laboratory
Singapore University of Technology and Design
{athul_harilal,ivan_homoliak,martin_ochoa}@sutd.edu.sg
{flavio_toffalini,john_castellanos,juan_guarnizo}@mymail.sutd.edu.sg

ABSTRACT

In this paper we present the design and outcome of a gamified competition that was devised in order to obtain a dataset containing realistic instances of insider threats. The competition simulated user interactions in/among competing companies, where two types of behaviors (normal and malicious) were incentivized. For the case of malicious behavior, we designed sessions for two types of insider threats (masqueraders and traitors). The game involved the participation of 6 teams consisting of 4 students who competed with each other for a period of 5 days, while their activities were monitored considering several heterogeneous sources (mouse, keyboard, process and file-system monitor, network traffic, emails and login/logout). In sum, we obtained 320 hours of active participation that included 18 hours of masquerader data and at least two instances of traitor data. Additionally to malicious behaviors, the students explored various defensive and offensive strategies, such as denial of service attacks and obfuscation techniques, in an effort to get ahead in the competition. The TWOS dataset is publicly accessible for further research purposes.

KEYWORDS

Malicious insider threat, masquerader, traitor, dataset, multi player game, user behavior monitoring.

1 INTRODUCTION

In today's world, insiders have the potential to bring harm to the organization in which they work [1, 15, 53]. Considering knowledge of an insider, Salem et al. [41] divided insider threat into two categories: masqueraders and traitors. A masquerader is a type of malicious insider who performs illegal actions on behalf of a legitimate user of a system [43]. On the other hand, a traitor is a malicious insider who misuses his own privileges to perform malicious activities. Traitors have full knowledge of a targeted system and its resources and can perform malicious activities without significantly deviating from their normal profiles.

As an effort to combat these threats, several works have proposed the analysis of user behavior using various features (i.e., file-system interaction [13, 14], biometric behavior by means of mouse [20] and keyboard usage patterns [26], among others). In order to perform such analysis, researchers rely on datasets that contain normal and malicious behaviors; a few such datasets have been made available to the research community.

However, there are number of challenges with obtaining high-quality datasets. Information about incidents of real insider attacks are kept confidential, since revealing the details could harm the reputation of the organizations involved [15]. On the other hand, even if organizations would be willing to share data related to user activity, it is often challenging to discern normal from malicious behaviors [15], which is crucial information in order to evaluate the performance of detection algorithms.

In particular, the detection of **masqueraders** has been studied actively since the work of Schonlau et al. [46], who profiled the interaction of various users in the Unix operating system by recording commands issued in a shell – by mixing sequences of commands issued by one user (say user A) with the ones belonging to one or more other users in the dataset (say B,C,D etc.), a labeled dataset was obtained. Then, the challenge of a detection algorithm was to distinguish the normal user behavior from the simulated masquerade behavior. However, the behavior labeled as malicious (normal behavior of other users) lacked malicious intent. Other datasets often used in masquerader detection do not explicitly provide malicious classes – *e.g.,* Greenberg's [22] and Purdue University [39] datasets of Unix commands, MITRE OWL [33] dataset of MS Word commands.

Such datasets were often used to evaluate algorithms that addressed user authentication, which is related but not equivalent to masquerader detection. On the other hand, there exist datasets in which malicious data were collected either by synthetic (e.g., WUIL dataset [12]) or by interactive (e.g., RUU dataset [44]) simulation of malicious intent.

In comparison to the masqueraders case, the compilation of useful datasets for **traitor** detection is a more challenging task due to the heterogenous nature of traitor activity that can be highly context dependent. Publicly available datasets containing traitor data involve the Enron [11] company and CERT datasets [16].

Motivation for a New Dataset. We list three main points that emphasize the need for a new insider threat dataset: 1) Previous research works have indicated that datasets containing substituted masqueraders are less suitable for identifying masqueraders [41]

ACM ISBN 978-1-4503-5177-5/17/10…$15.00
https://doi.org/10.1145/3139923.3139929

(this is in contrast to datasets containing malicious tasks). 2) Although there is substantial research dealing with the masquerader detection problem, only few works use datasets specifically built for such purposes i.e., WUIL and RUU datasets. Amongst these datasets, WUIL contains only synthetically executed masquerade sessions that might be far from real user's behavior. 3) We have observed significant amount of research in masquerader detection, but fewer works related to the traitors. This can be explained by the argument that masquerader detection is simpler and more straightforward than traitor detection, as observed by Salem et al. [41], who mention that a masquerader is likely to perform actions inconsistent with the victim's typical behavior, and behavior is something that cannot be stolen.

Collected Dataset. We designed a multi player game, The Wolf of SUTD (TWOS), that encouraged user interactions in a simulated corporate environment, and whose purpose was to provide a comprehensive dataset containing interactive malicious insider threat activities involving both masqueraders and traitors. This was achieved by creating a gamified setting where sales departments of competing companies (represented by teams) contacted a common set of customers. Customers had different amounts of points they were willing to give, while they were committed to invest in the first sales team that "made a deal" with them. If a customer already made a deal, then he would not reply to any further requests from other teams. The goal of a team was to collect as many points as possible. We introduced masquerade sessions at specific time intervals in which each team was given access to a machine that belonged to another team's member. Masqueraders were motivated to steal the list of obtained customers from the victim's machine or to sabotage it, and thus prevent other teams from winning. Next, we also introduced the firing and hiring periods, where some participants were forced to change teams in the middle of the competition; this incentivized traitor behavior and enabled fired participants to steal the original team's data. Thus, unlike other datasets, our malicious data are not synthetic or injected, rather they followed from spontaneous user interaction with machines.

We have collected data from several heterogeneous sources as an attempt to study their cumulative effect for detection of malicious insiders. The dataset includes activity recorded from mouse, keyboard, process and file-system monitor, network traffic, email and login/logout of users. This dataset has been anonymized in order to not reveal any privacy sensitive information. In total, we captured 320 hours of activity from 24 users spanning across 5 days. Additionally, we obtained 18 hours of masquerader data and at least two instances of traitor data. During the competition, we also observed some interesting events, such as teams trying to automate the process of contacting the customers or teams deploying effective countermeasures that protected their "assets" from masquerade attacks.

Note: The dataset is available at URL: http://cyberlab.sutd.edu.sg/twos-dataset.

2 BACKGROUND

A scientific approach for tackling insider threat requires high-quality datasets that faithfully reflect real scenarios, and therefore enable researchers to develop effective detection algorithms. Nevertheless, collecting a useful dataset that contains insider attacks is challenging for several reasons. First, companies are not willing to share datasets involving malicious incidents due to matters of privacy issues. Second, it is difficult to simulate insider attacks under laboratory conditions because the simulations might not represent human induced actions in their natural form.

Even though there are complications to generate or obtain useful datasets, various research projects and institutions have published datasets addressing the context of insider threat [11, 12, 16, 19, 22, 33, 39, 44, 45]. However, these can be considered limited in functionality as they were created several years ago or they contain only a few data sources, such as network traffic, emails, or keystrokes (we discuss more about them and their shortcomings in Section 5). Thus, we identify a need for a new dataset that includes malicious insider scenarios from multiple data sources.

The following describes the attacker models employed in our experiment and why they have been chosen. Then, we outline the sources of data collected in order to achieve our goal.

2.1 Attacker Model

According to [48], we can classify an "insider attack" as one that is initiated by an entity already inside the security perimeter (the "insider"). It may refer to an entity that was previously authorized to access system resources but uses them in an inappropriate way. From this statement, we have defined our attacker models by taking inspiration from ones already discussed in the literature: masquerader and traitor [41].

Masquerader. A masquerader is an attacker who can steal credentials or sessions of legitimate users, and once he gains access to the system, he impersonates the victim user to perform malicious actions using the available privileges and information resources. However, such attacker might have less knowledge about the system that is under attack. So, he may need to seek through the system to identify valuable assets [41]. Masqueraders can get the victim's credentials in different ways, such as phishing, rootkits, social engineering, etc. For example, BlackEnergy trojan that was utilized in the Ukraine power plant attack [10] misused user's privilege through macros in malicious documents.

Traitor. A traitor represents an attacker who knows the targeted system and has been granted access to information resources. This attacker uses his legitimate credentials to perform malicious actions [41]. Also, attacks performed by traitors are much more complicated to detect due to fine-grained deviations from their normal behavior. A well-known case of these type of threats is the Bradley Manning's case [7].

2.2 Types of Data Collected

In order to identify insider attacks, we realized the importance of gathering data from heterogeneous data sources, which can provide us with a holistic representation of what is going on in the context of an attack. Therefore, we intend to study the it's cumulative effect in the for the detection of malicious insiders. Most of the previous works do not provide a global context of attacks from heterogeneous data sources; instead they focus primarily on one data source:

- mouse activity [19],
- keystrokes [33],
- host activities [12, 16, 44] (such as system calls or file system interaction),
- network traffic [16],
- email activity [11, 16],
- login and Logout [16, 22],
- UNIX commands [39, 46].

3 THE GAME

We designed the competition with an intention of capturing user interactions in/among companies, which contain normal and malicious behaviors, and moreover is inspired from a real sales group.

We emulated a scenario that mimicked the setting of similar companies that try to win over customers by pitching their products. For simplicity, the companies offered similar products and they aimed to sell their products to a common set of customers. The employees of companies were mimicked by students, who participated in the game. For the experiment, we randomly grouped 24 students into 6 teams of 4 members each. Each team emulated the sales department of a company that was entrusted with the task of contacting and dealing with a set of customers.

We simulated customers through an automatic script that was developed for the experiment. More specifically, we set a bucket of synthetic customers (10.000 entities), which we shared with all teams at the beginning of the competition. Each customer has a few points (similar to amount of money) that were given to students upon fulfilling a predefined set of operations; it will be described later. The goal of the teams was to obtain as many points as possible by contacting and dealing with the customers.

To obtain malicious data, we created a scenario whereby intelligent but not necessarily technically skilled users were motivated to behave maliciously, while at the same time they could face consequences if caught. Students that participated, came from different technical backgrounds which is similar to the composition a real company. All students had access to a virtual workstation where all activities were monitored and logged for further analysis. The choice of virtual workstations allowed us to limit a number of outgoing channels, such as USB drives that are often restricted in many companies. On the other hand, the contest was designed to keep the anonymity of the participants, and therefore no real student ID's were stored.

Overall, the competition lasted for five days (from Monday to Friday), and in the end, the top 3 teams in the order of accumulated points were awarded. The top 3 teams were rewarded $800, $400, and $200 respectively. Every student was also given $15 for their participation. During the competition, students were expected to play for at least 10 hours. If they failed to do so, their prize was proportionately reduced to the amount of the time they played, which was computed based on the mouse and keyboard activities (see Section 4). Hence, the students were financially motivated to actively participate in the game.

3.1 Game Stages

We split the competition into various stages that allowed us to mimic a more realistic corporate scenario (see Figure 1). During *the normal periods*, participants had to perform some tasks in order to collect points; this enabled us to obtain normal user data. For the purpose of obtaining malicious user data, we designed: 1) *the firing and hiring period*, where team leaders were forced to fire an employee in order to incentivize traitor behavior ; and 2) *the wildcard periods*, where each team obtained credentials of another team's member, and hence were able to access the victim's machine. In particular, we describe the periods as follows:

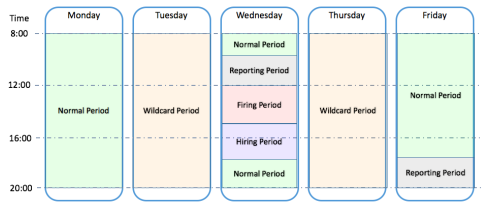


Figure 1: Competition schedule with stages, every period is drawn in a different colour

Normal Period. During this phase, participants were contacting customers and striving to understand the dynamics of obtaining points from the customers. During this period, we recorded normal user behavior.

The task of obtaining points from a customer is depicted in Figure 2. The first step is to construct a meaningful message addressing the customer. The message needs to satisfy a few conditions regarding the length of the body, grammatical correctness, and respectful salutation of a customer. Upon it's satisfaction, the customer would present a captcha engraved with either 3 or 6 words. The motivation behind this two fold interaction was to make the game more interesting and realistic, and to encourage participants in writing a different text to every customer. This enabled us to obtain a richer keystroke dataset. After obtaining the captcha, students had to construct sentences with all the included words and also fulfill the previously mentioned checks. Upon the satisfaction of all checks, the students were rewarded with points and a secret token from a customer. If a customer was previously contacted, then all further replies from that customer would contain the anonymous ID of the student that claimed points. This gave students a rough estimation of active players in the game.

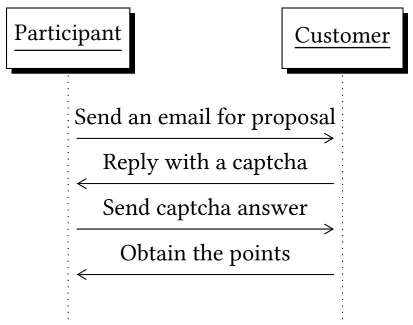


Figure 2: Interaction between a participant and a customer

Wild Card Period. During this phase, a student from each team was randomly chosen to be an attack target. We also randomized the pairing of masquerader team and attacked victim. The wild card period was scheduled on Tuesday and Thursday (see Figure 1), and each team was given a fixed time slot of 90 minutes to attack the victim's account. The masquerader team was notified about the victim's credentials three hours before the actual attack period. The attack period and notification period were chosen due to following reasons: (i) While notifying a masquerader, it is necessary to ensure that the credentials do not fall into the hands of a team that is supposed to attack the masquerader team. It might happen when a notification is sent to a team that is under masquerade attack. (ii) The notification period prior to attack was chosen to ensure the availability of atleast one student from the masquerader team.

Score Reporting Period. During this period, each team had to submit a list with names, secrete tokens and points of all customers that were acquired. This list could also include information that was stolen from another team during the masquerade period or through other means. In our infrastructure, we logged the student's ID when he successfully claimed points from a customer. This can be utilized to track customers that did not belong to a certain team. Our intention was to encourage students to behave maliciously and therefore we liberalized the checks by checking only 10% of the customers from the list. If stolen customers were detected in the submitted list, 10% of the total team's score was deducted. In the case when a team failed to submit the list, we computed the score from our logs and deducted 20% from it.

The score reporting period was scheduled on Wednesday and Friday (see Figure 1), while the latter marked the end of the competition. The main purpose of conducting it on Wednesday was to reflect the performance of each team in comparison to others and to motivate teams lower in the rank to play more.

Firing and Hiring Periods. After the end of the score reporting period on Wednesday (see Figure 1), every team, excluding the team ranked first, had to fire a member. This period was designed to create room for traitor behavior similar to scenarios when an employee is about to leave a company. The participants who were fired from their respective teams were notified about it a couple of hours before their machines were reset. Since the students were financially motivated, expulsion from their original team could arouse emotions leading to malicious actions for personal gain. For example, a fired student could exfiltrate sensitive information or establish new alliances with other teams. After firing a student, he was randomly assigned to a new team. During the hiring phase, accounts of fired members were reset and from then onwards, they became part of a new team.

3.2 Architecture

We set up the infrastructure for the competition on the cloud using Amazon web services [3] (see Figure 3). The infrastructure included 3 Amazon EC2 servers [2] and 24 Amazon WorkSpace instances [4]. These EC2 servers were used as Domain Controller, File Server and Network Proxy server respectively. A virtual Windows machine was assigned to each participant (AWS WorkSpace instance) in order to participate in the competition.

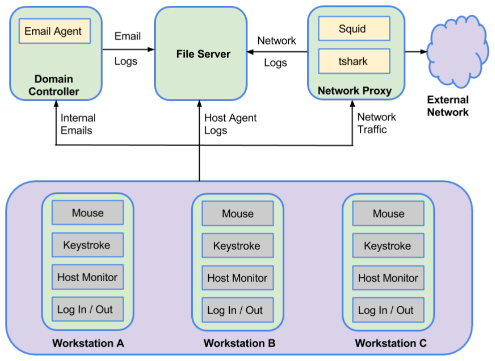


Figure 3: Architecture of infrastructure

The workstations were set up with a standard software suite (*i.e.,* MS Office, Mozilla Firefox and Microsoft Outlook [31]). We also provided a private email address to every participant for internal communications. All participant accounts were configured to not have administrator privileges (*i.e.,* they were not allowed to install any new program or change configurations). In the architecture, all machines were managed by a Windows Domain Controller server [49], which was selected due to the following reasons:

(i) It is a widespread solution in corporate environments.
(ii) Windows Domain Controller is based on Active Directory and it enables one to easily manage the infrastructure.

Each user workstation was configured to run 3 agents that logged system calls, mouse, and keyboard activity. Mouse and keystroke agents were programmed in python and leveraged *pyinput* library [36]. The host monitor agent was responsible for logging system calls generated by each Amazon WorkSpace. We chose Process Monitor [54] as a host monitor agent, since it is a standard Windows tool for forensic and system analysis [40].

The File Server served as a repository for accumulating log files from the host machines. The logs created by mouse and keystroke activities were small and they were updated slowly. Hence, a direct network path to the File Server was opened for them. On the other hand, logs generated by the Host Monitor were massive and updated very quickly. Hence, in order to prevent the creation of large network buffers and save bandwidth, the logs were compressed and sent to the File Server on an hourly basis.

In order to intercept network traffic from workstations, a man in the middle squid proxy server [50] was configured (see Network Proxy in Figure 3). Trustworthiness of Network Proxy was instilled by the installation of a certificate into the workstation's trusted list of certificates. Due to above technique, HTTPS traffic could be intercepted and decrypted. All network logs were captured by tcpdump [52] in PCAP format and they were transferred directly via a shared network path.

Simulated Customers. Synthetic customers were simulated with the help of Microsoft Exchange service [32]. For contacting the customers, participants used Microsoft Outlook [31]. Microsoft Exchange comes with an option to copy all incoming messages to a specific mailbox. Using this technique, we captured all emails sent and received by each participant. However, emails sent by the customers were not saved as they were redundant in nature. All emails were directly logged into MYSQL [35] database within the File Server. For the sake of simplicity and in order to keep our email service private from external email services, we configured the email server to not deliver emails outside. The only channels that allowed for external communication were HTTP and HTTPS protocols. Other communication channels of the workstations such as shared clipboard were explicitly disabled.

4 DATASET

In this section we describe the data collected from the competition and their structure. Then, we illustrate trends observed across the phases described in Section 3.1. Finally, we report a list of interesting events that happened and lessons learned during the experiment.

4.1 Description of Data

The activities performed during the competition were collected by our architecture into 7 different datasets. Each dataset and its anonymization mechanism is described as follows.

Mouse Traces. This dataset contains all actions generated by mouse movements and clicks. More specifically, these data refer to the position of the cursor on the screen, which was sampled every 16*ms* and it was measured in pixels. At each position, we indicate which mouse action was involved: mouse movement, button pressed/released or scroll. Moreover, we also provide the monitor's resolution. Because of the nature of these data, we did not employ any anonymization mechanism. This data source primarily serves the purpose of identifying the normal user based on biometric behavior.

Key Strokes. This dataset contains all keys pressed by the users. It logs all characters that include alphanumeric and special symbols. Furthermore, we indicate whether the actual key was pressed or released – the latter information can be used for measurement of how long a specific key had been pressed.

Since it is possible to infer a lot of sensitive information by rebuilding the text – *i.e.,* passwords, telephone numbers – we employed an anonymization process that allowed us to preserve as much information as possible but at the same time, it made reassembling of the original text challenging. We accomplished it by taking inspiration from typewriting [8], where the keyboard is split into zones. More precisely, we split the qwerty *en-US* keyboard layout into three zones – left, center and right. Then, we substituted each letter with its relative zone. In a similar vein, we grouped all digits into a single symbol, while we left all other keys such as `ctrl`, `alt`, punctuation symbols in their original form. Hence we have tried to keep as much information as possible, while also addressing privacy issues; this renders the dataset useful to the research community. The mapping of the keyboard layout is depicted in Figure 4.

Host monitor (Process and File-System Monitor). The main information provided by the Host Monitor were related to file accesses (*e.g.,* open/read/close), registries (*e.g.,* query/set/get value) and processes (*e.g.,* spawn/destroy). The logs were anonymized by replacing all file paths, registry paths and user names with random

$$\begin{aligned}\{Q, W, E, A, S, D, Z, X, C\} &\Rightarrow \text{LEFT}\\ \{R, T, Y, U, F, G, H, V, B\} &\Rightarrow \text{CENTER}\\ \{I, O, P, J, K, L, N, M\} &\Rightarrow \text{RIGHT}\\ \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} &\Rightarrow \text{DIGIT}\end{aligned}$$

Figure 4: Mapping for anonymization of characters

tokens. The mapping of such strings to random tokens is kept in our database for consistency with other data sources. We further created a white list of paths that were not anonymized. This list contained paths related to the Windows structure and it was established in order to help future analysis without compromising users' privacy. The white list of these paths is as follows:

- C:
- C:\Program Files
- C:\Program Files (x86)
- C:\Windows
- D:
- D:\Users
- HKEY_CLASSES_ROOT
- HKEY_CURRENT_CONFIG
- HKEY_CURRENT_USER
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKLM

Network. We decided to parse only HTTP protocol because it was the only protocol used by students for interaction with external world. The network traffic was captured by our HTTP proxy server. This allowed us to monitor only HTTP communications and easily extract specific features from them. For the anonymization phase, we employed the following. We substituted the private IP addresses with new ones, while we kept all public IPs unchanged. For the transport layer, we preserved the headers of *TCP* and *UDP* packets in their original form. However, we substituted the original payload of packets with *JSON* string that contained features extracted from the original payload. It also includes the length of the original payload in Bytes. For a *TCP* packet containing an *HTTP* request, we also added the *method* (*e.g.,* GET/POST) and the *host*. If a *TCP* packet contained an *HTTP* response, we also added its *status code*, *content length* and *content type* (*e.g.,* text/html, image/jpeg).

Emails. We saved all email messages originating from students and technical support, however we discarded emails originating from synthetic customers. In the preprocessing stage, all email addresses, usernames, and paths were anonymized using mapping from our database. If a URL occurred in an email message, we preserved only it's domain name, while query string was removed. In order to perform sentimental analysis only on the recently written text by an originator of an email, we omitted history of the conversation. Such input was passed to existing Linguistic Inquiry and Word Count (LIWC) tool [38], which generated 94 features expressing membership ratio to each of 94 word groups (e.g., anxious, angry, negative).

Logon/logout Activities. We monitored users' login/logout activities using Windows event log [40]. We opted for this technique because it is a standard tool for gathering these information in Windows environment, and it is useful for host-based analysis.

Psychological Questionnaire. Taking inspiration from the previous works [5, 15, 17, 24, 29, 47], we asked the participants to fill up a psychological questionnaire. We used the questionnaire inspired by dark triad theory [37], which contained 50 questions. This questionnaire may enable researchers to correlate participants' behavior with psychological indicators.

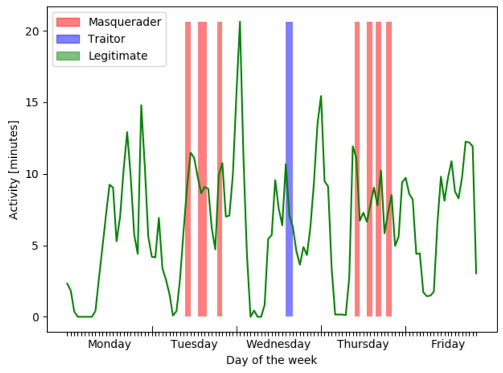
4.2 Preliminary Analysis And Statistics

In this section, we describe statistics of the dataset and other important information regarding particular teams and their members, respectively. We believe that the following information will help interested researches in interpreting the data.

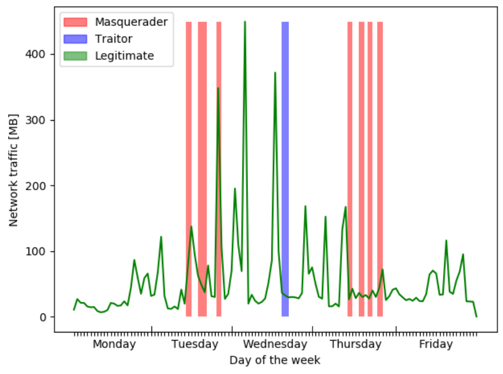
Statistics of Users' Activity. Average activity of all students are illustrated in Figure 5, which depicts active participation from three points of view (*i.e.,* keyboard/mouse, network traffic, emails sent). All plots represent the activities of the whole week and they further show the phases of the game: red bars represent masquerade sessions and dark blue bars represent traitor sessions.

More specifically, an average keyboard/mouse activity per hour is depicted in Figure 5a. We consider a minute as active, when a participant was logged into a machine and we recorded at least one entry of mouse or keyboard activity in that minute. We employed this heuristic because only login/logout actions were not enough to determine whether a user was physically working on a machine. He could have just opened the window without any human intervention. Looking at the graph, we can observe that average user activities dropped after midnight and rose again in the morning. There is also a significant peak on Wednesday, just before the 1st scoring period. This can be explained by the fact that teams tried to gain as many points as possible before the 1st scoring period. Other important spikes were observed during the wild-card periods, which indicates active participation by students during these periods.

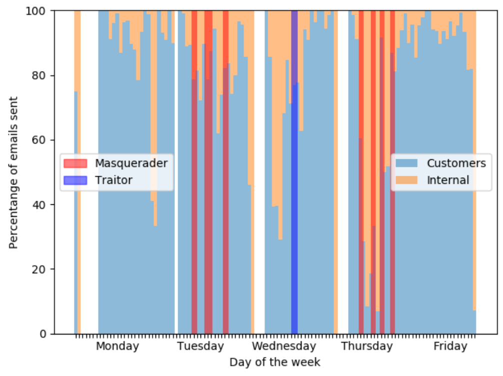
Average per user network traffic transferred through Network Proxy is depicted in Figure 5b. Amount of network traffic is an important measure because this was the only channel for exfiltration of data outside of user machines according to our architecture. We remind that participants were restricted from directly copying and pasting information between a VM and their physical machine. It is interesting to note that the amount of network traffic spiked during the first round of masquerade attacks on Tuesday, while the network traffic was low during the second round of masquerade attack on Thursday. These observations can be explained by looking at different contexts. During the first wild-card period, masqueraders were able to exfiltrate a lot of data. Therefore some users were more prudent in the next wild-card period by improving their defense techniques. In particular, participants took inspiration from the consequences of the first wild-card period and decided to protect their data by storing them outside of their machines, before the beginning of the second wild-card period. This can be observed



(a) Average keyboard/mouse activity per hour



(b) Average network traffic activity per hour



(c) Percentage of emails sent to customers or internally

Figure 5: Competition statistics

from the fact that the network traffic dropped just before the second wild-card period on Thursday. Second wild-card period can be useful to study the defensive techniques employed by students. The last important observation can be seen before the 1st scoring period. During this time, participants knew that they might loose their data if they would be fired from the team. Therefore they did a backup of their data. We can also see that after the hiring phase other smaller spikes appeared and this might indicate attempts to recover the backed up information.

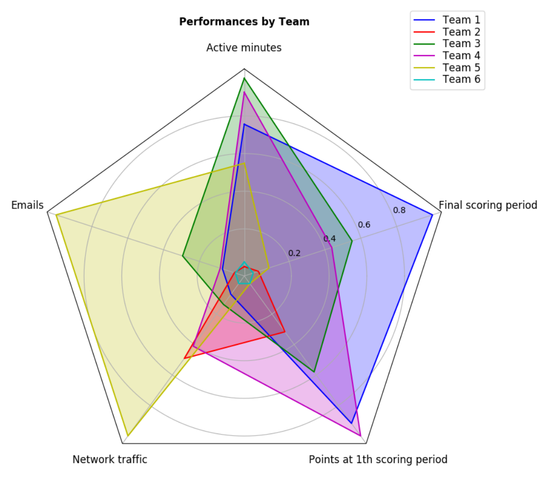


Figure 6: Comparison of all teams

We illustrate in Figure 5c, the percentage of emails sent by participants towards the customers against emails sent for internal communication among the participants or technical support. If no emails were sent in a certain period of time, then we set both bars to *zero*. Although for the majority of time most of the emails were sent towards customers, there were some moments where this trend shifted in the favor of internal communication. At the beginning of the game, there was a spike in the emails sent among the participants, which occurred because each team was planning its future steps. The percentage of internal emails also grew before the 1st scoring period and during the 2nd wild-card period. Moreover, we also observed an increasing number of internal emails at the midnight of Tuesday and also at the final stage of the competition because of the final scoring period.

Teams' Performances. Qualitative evaluation of every team's performance from five different aspects is shown in Figure 6. All metrics have been normalized to fit the range of radar plot into the interval [0.0, 1.0]. Looking at the image, we emphasize correlation of the final scores and the activity, which is evident for the best two teams – Team 3 and Team 4. Although statistics of Team 1 show less activity, it was able to achieve the first position in the final scoring period. Team 1 came first due to the high amount of stolen points during wild-card period (see Figure 8). Another interesting situation can be seen regarding Team 5. This team invested a lot of effort into trying to cheat the system; they managed to automate the process of sending emails to customers in order to receive the captchas (see Section 4.3). They also tried to automatically extract the captchas through an external service, which was noticed in their email communications. However, they were unsuccessful in completing the attack. Due to the above mentioned reasons, this team generated maximum amount of network traffic and emails, but on the other hand earned only few points.

Amount of activity spent by each team and its members is depicted in Figure 7, where dashed lines represents average hours of activity by each team's member. In sum, four teams generated more than 15 hours of activity on average. Moreover, majority of the teams showed a similar type of behavior, where one user played significantly in comparison others.

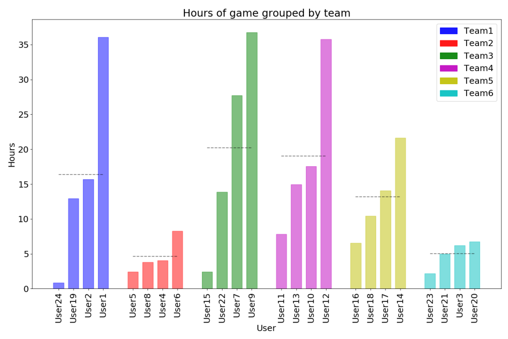


Figure 7: Total user activity

Analysis of Cheating. We analyzed the score files send by teams during the two scoring periods in order to infer cheating, which is depicted in Figure 8. In the plot, we show only the teams that submitted their score reports to us and we excluded the ones that did not submit.

Vertical bars represents the amount of points gained in total, where green colored bars represents points collected by contacting customers, orange colored bars represents stolen points and blue colored bars represents points from corrupted entries (incorrect information) in submitted files. Note that score from corrupted entries was not considered in overall sum. Additionally the red X represents random checks for stolen points. We can see that the winning team (Team 1) managed to steal more points than others.

4.3 Notable Events Observed

In this section, we describe a few interesting events that were observed during the competition. Information about such events was obtained from email logs, and the scope of such events include both malicious and normal periods. Since some of those events

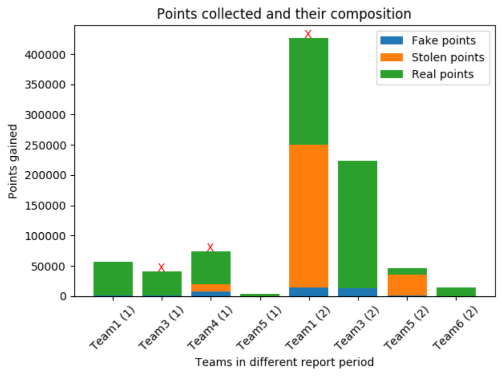


Figure 8: Composition of score per period

occurred outside the malicious periods, they can be considered as added value due to their spontaneous nature. The dataset contains an index that describes which participants were involved in each event and when the event happened.

Automation. Since every team was entitled to contact as many customers as possible, participants tried to understand the principle of the agent that was responsible for replying to their emails. A few students from Team 1 and Team 3 managed to semi-automate the process of sending emails to the customers. By understanding the length requirement of the body of the email, they automated the process of sending the initial email to the customer. For the second email (captcha answer), they created a response template that only needed the words from the captcha to be put in the same order as they appeared in the captcha, which had to be done manually. This resulted in minimal human intervention for carrying out the task.

Denial of Service Attacks. A few students misused the infrastructure by initiating transactions with a huge number of customers by spamming and this caused the email service to be unavailable to other students. Such transactions were performed by members of Team 5, who created a script that automatically obtained captcha images of all customers. After that they tried to identify the text embedded in a picture using an OCR (Optical Character Recognition) software [34]. However, they were unsuccessful in their endeavor. After the detection of such attacks, we limited the number of email messages sent per minute to 5. Later, similar behavior was observed by Team 3. These attacks were detected by the spikes in emails traffic on Monday evening after 10PM and on Wednesday at midnight.

Masquerade Period Countermeasures. Since the teams did not know when and who amongst them would be attacked, they deployed various countermeasures to prevent the attacking team from getting the list of the earned points.

A few members from Team 3 deployed multiple score files with password protection in order to confound and slow down the masquerader attacking their machines. This was similar to the concept of honeytokens, which are intriguing to attackers and who may fall into their trap. However, the files created by students did not have all the properties of honeytokens. Team 4 introduced an obfuscation technique in order to make their earned points unavailable for other teams attacking their machines. Members of this team added a fixed amount of points to each customer's actual points. This resulted in the points being useless to the attacker, but allowed Team 4 to easily reconstruct the original points. Other protection countermeasure was observed in Team 1 and Team 3, which used web services for storing important files.

Attacks in Masquerade Period. During the wild-card period, a number of miscellaneous attacks were observed. A few instances of sabotage were observed according to email sources. For instance, a member of Team 2 deleted all emails and important folders of the attacked victim. Similar behavior was observed during the attack of Team 4 on Team 5. Important files found by the attacker were leaked from the victim's machine, which is indicated by a high number of emails with attachments. Another interesting breach was implemented by a few members of Team 3. They were able to create a rule within MS Outlook, which enabled one to forward some emails of the victim to the attacker. Later, the victim found and rectified the matter.

Traitor Attacks. During the firing period, one student from each team (excluding the winning team) was fired and randomly put into another team. Out of the 5 students who were fired, only two students were active during this period. This resulted in having only few instances of traitor data.

Data exfiltration of important files was observed by the member who was fired from Team 1. This member exfiltrated data through emails to the new group; this is similar to the malicious insider scenario, where an employee leaving the company steals some proprietary data for benefit of his further job.

The second case of firing an active member happened in Team 2. However the firing was not initiated by the team leader and therefore we randomly fired one member. A few emails were exchanged between the fired member and the leader of Team 2 without any attached files. From these emails, we observed conversation regarding the possibility of data exfiltration. But the intent of fired member did not propagate into malicious actions such as exfiltration of sensitive data. This was a preliminary analysis of our traitor cases. However deeper forensics are needed to capture more such instances.

4.4 Summary of Collected Data

Summary of the collected data are described in Table 2 of Appendix. In the same fashion, we show a summary of data collected during the malicious sessions (both *masquerader* and *traitor*) in Table 1 of Appendix. The *User* column indicates the user account from which the data was recorded. Columns *mouse* and *keyboard* indicate the number of their respective entries that were logged, while *Network* column shows the amount of network data sent and received by a user machine. Finally, *Mail* column informs about the number of emails that originated from an email account.

4.5 Lessons Learned

During the course of the competition, occurrence of certain events have enabled us to understand a few areas that can be improved. Some of them received immediate attention as they were affecting the performance of the infrastructure, while others that were not severe can be taken as the lessons learned from the competition. When we observed the Denial of Service attack mentioned in Section 4.3, the Email Agent was significantly slowed down and it affected other students while playing. This was immediately resolved by employment of a rate filter that allowed only 5 outgoing emails per minute, per user. Also, the domain controller was configured to forbid installation of any software into the workstations. In spite of it, members of Team 5 were able to bypass it in order to initiate the Denial of Service attack. Therefore, in the future we will consider to use other optional security mechanisms such as Mandatory Access Control.

Although we obtained traitor instances during the firing and hiring period, we observed that many fired participants were inactive. Hence, we plan to take countermeasures in the future round of the experiment in order to avoid such situations by *e.g.,* making more firing and hiring periods. Also, we plan to incorporate a concept drift that represents changing behavior of normal users with time such as performing different tasks or moving to other projects.

5 RELATED WORK

We divided related work into game-based approaches and datasets, both dealing with insider threat problem only.

5.1 Game-Based Approaches

Although we found several game-based studies dealing with the insider threat problem, none of them provided collected dataset to research community in comparison to our work. The following contains identified examples of such studies.

Brdiczka et al. [9] utilized data from World of Warcraft game for insider threat detection, where malicious data were represented by players who decided to quit a guild. Therefore, profiles of such malicious users were mostly similar to that of traitors. The author aimed at analysis of social network data, psychological profiling data and behavioral data.

Taylor et al. [51] conducted four-stage multi player game dealing with organized crime investigation that involved 54 participants and it lasted for 6 hours. Participants were working in teams of 4 members, each having access to different database of information. During the stages of the game, participants were asked to perform data exfiltration tasks that required data from other users' databases for certain payoff – this mimicked traitor behavior. Although participants were able to leak or obtain some information through shared printer or unlocked workstations, only email communication among the participants was captured.

In the similar vein, Ho et al. [25] organized a multi player game called Collabo that involved 27 participants who formed 6 teams. The game lasted for 5 days and consisted of solving assigned tasks that must be completed within a given timeframe. The list of assigned tasks consisted of 7 logical problems per day; these tasks were the same for each team, but were assigned to them in various order. The goal was to collect as many points as possible while solving the tasks. The competition rewarded the top 3 teams, while an additional financial award (a "bait") was introduced secretly to team leaders of 3 teams and made them to face an ethical dilemma: a) collaborate with their teammates to achieve the best outcome and, if they win, distribute the additional prizes evenly with the teammates, or b) undermine the team's collaborative efforts and keep additional prize for themselves. Authors of the game aimed at language action cues in chat messages of insiders with an intention to detect changes in traitor's behavior. While in our case, malicious actions of the users can be analyzed using various data sources in an effort to look for more indicators of malicious activities.

Azaria et al. [6] designed a single player game, called BAIT, in which players had to select tasks that they would perform in a high security facility, while working with classified information. A player might either be an honest worker or a malicious insider. Both types of players received a list of classified topics and should gather information on each of these topics, edit this information and send it to the topic's requester. Additionally, all the players were given another topic covering their personal interests. Malicious insiders were given a special topic (e.g., design plans for a new missile) and they were told to exfiltrate data related to the topic, while minimizing the likelihood of detection by the surveillance system. The authors used host-based monitoring approach that recorded fetch, transfer (i.e., to USB, printer, CD/DVD) and send (i.e., by email, Internet, unencrypted) actions of users. The game was performed on Amazon Mechanical Turk (AMT) and involved 654 benign players and 45 malicious ones.

5.2 Datasets

We divided commonly used datasets in the insider threat detection into five categories: (1) *Masquerader-Based*, (2) *Traitor-Based*, (3) *Miscellaneous Malicious*, (4) *Substituted Masqueraders*, and (5) *Identification/Authentication-Based*. These categories are depicted in Figure 9 and can be obtained by consecutive application of the following three criteria:

(a) discerning the user's intent in nonuser's data (i.e., data considered as "malicious"), which yields *malicious* and *benign* branches,

(b_1) for the malicious intent branch, by the way in which policy violation was executed – using legitimate user's access *(Traitor-Based*); or by obtaining unauthorized access (*Masquerader-based*), or both of the cases are independently included in a dataset (*Miscellaneous Malicious*), and

(b_2) for the benign intent branch, by discerning whether explicit formation of substituted malicious class is present. Here *Substituted Masqueraders* category contains such malicious class and *Identification/Authentication-Based* category does not.

The most valuable datasets for research community lie in the malicious intent branch, and are collected from real companies in the optimal case. As such optimal cases are rare (e.g., [11]), the second best option is to use datasets aimed at simulation of real environments (e.g., [16, 44]), which we followed in our paper.

Masquerader-Based Datasets. Although there is a lot of research dealing with the masquerader detection problem, only few are using datasets specifically built for such purposes. The following contains datasets that include malicious intent in data with malicious labels, while they are aimed at violation of policy by obtaining unauthorized access.

RUU (*Are You You*) dataset [44] is a masquerader dataset that was introduced by Salem and Stolfo in [42, 43]. The dataset was generated by 34 normal users and 14 masqueraders and unlike our dataset consists only host-based events derived from file system access, processes, windows registry, dynamic library loading, and window events. The dataset contains masquerade sessions performed by humans according to a specific task of finding information that could be used for financial gain.

WUIL (*Windows-Users and Windows-Intruder simulations Logs*) dataset [12] has been designed and implemented by Camiña et al. [13] and includes generic file system interactions regardless of their type (i.e., open, write, read). WUIL dataset contains records from 20 users (updated to 76 in [14]) who were monitored at different periods of time during their daily routine activities. The data were collected using an internal tool for file system audit of Windows machines of various versions (i.e., XP, 7, 8, and 8.1). While the legitimate users' data had been collected from real users, the

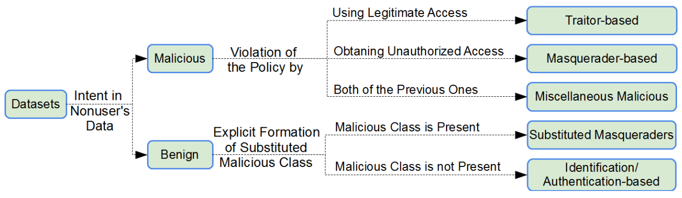

Datasets
Intent in Nonuser's Data
Malicious
Benign
Violation of the Policy by
Explicit Formation of Substituted Malicious Class
Using Legitimate Access
Obtaning Unauthorized Access
Both of the Previous Ones
Malicious Class is Present
Malicious Class is not Present
Traitor-based
Masquerader-based
Miscellaneous Malicious
Substituted Masqueraders
Identification/ Authentication-based

Figure 9: Categorization of datasets

masquerade sessions were simulated using batch scripts considering three skill levels of users: *basic*, *intermediate*, and *advanced*. Hence their dataset contain synthetic malicious instances unlike our dataset.

Traitor-Based Datasets. For malicious intent branch, datasets for dedicated traitor detection is not as widespread as masquerader case. This can be explained by the assumption that masquerader detection is simpler and more straightforward than traitor detection, as argued by Salem et al. [41] who mention that masquerader is likely to perform actions inconsistent with the victim's typical behavior and behavior is something that cannot be stolen. The following contains datasets that include malicious intent in data considered as malicious, and thus they are aimed at violation of policy using legitimate access. **Enron** dataset [11] consists of a collection of 500,000 real-world emails (from 1998 to 2002) associated with 150 users, mostly senior management of the Enron company; some of the users represents collaborating traitors. Although some of the emails were deleted as they contained attachments or confidential information, this dataset contains interesting information that can be used for text analysis, social network analysis, or link analysis aimed at detection of insider threat. However, the dataset lacks other sources that could be used to analyze user activities in more detail.

Miscellaneous Malicious Datasets. The datasets composed of both malicious insider subtypes (masqueraders and traitors) belong to this category.

CERT with other partners generated a collection of synthetic insider threat datasets [16] and described generation approach of the datasets in [21]. CERT datasets were generated using scenarios containing traitor instances as well as other scenarios involving masquerade activities. The collected logs contain logon data, browsing history, file access logs, emails, device usage, psychometric information, and LDAP data.

Substituted Masqueraders from Benign Users. In this category of datasets, data considered as malicious are explicitly substituted by other legitimate users' data. Unlike TWOS, these datasets contain data labeled as malicious that do not represent malicious intent. Previous research has indicated that such datasets are less suitable for testing of masquerader detection solutions in contrast to *Masquerader-Based* datasets [41].

Schonlau dataset (a.k.a. SEA) [45] was introduced by Schonlau et al. [46] and consists of sequences of 15,000 Unix commands per user that were produced from 50 individuals with different job roles. In this dataset, masquerader data are obtained by randomly mixing normal data from other users and thus the data does not contain any malicious intent. Maxion showed that the Schonlau dataset is not appropriate for the masquerader detection task [30].

Balabit Corp. has created a dataset intended for performance evaluation of behavioral biometrics based on mouse dynamics [19]. In this dataset, mouse activities from 10 users were extracted from Remote Desktop Protocol (RDP) connections and the dataset contains 1,612 hours of logged mouse activities. During the data collection, users did not have to follow any specified tasks, however they usually performed administrative tasks on remote desktops. The dataset contains masquerader data which are again obtained from legitimate data of other users.

Authentication/Identification-Based Datasets. This category of datasets can be used for the purpose of identification or authentication of users, regardless of their intent, although benign intent is assumed implicitly. The following contains examples of this category.

Greenberg's dataset [22] is the first known collection of authentication-based data. The author collected a dataset comprised of full command-line entries (including arguments and timestamps) from 168 users of the Unix shell *csh* [23]. The original data were split into four groups comprising of 55 novice users, 36 experienced users, 52 computer-scientist users, and 25 non-programmer users.

Purdue University (PU) dataset [39] was introduced by Lane and Brodley [27] and contains 9 sets of sanitized UNIX command user data. This was drawn from *tcsh* shell histories of 8 computer users at Purdue over the course of 2 years.

MITRE OWL (Organization-Wide Learning) dataset [33] was designed for continuous knowledge acquisition and individualized tutoring of application software across an organization [28]. However it was was also used for analysis of human interactions with GUI-based applications for the purpose of user authentication [18]. During a period of two years (from 1997 to 1998), the data were collected from 24 employees using Microsoft Word on Macintosh operating system. The dataset contains a total of 74,783 commands corresponding to 11,334 sessions.

Hence our dataset differs from the above mentioned datasets in one or more ways. Unlike other datasets, TWOS contains malicious intent in data labeled as malicious, and they are logged as a result of spontaneous user interactions with the workstation.

6 CONCLUSION

We have collected a dataset of 24 users from several host-based heterogeneous data sources (such as mouse, keyboard, processes and file-system) by means of a carefully designed gamified competition. In accordance to the proposed scenarios of the game, the dataset contains a mixture of normal and malicious activities. Unlike other datasets, the malicious masquerader and traitor activities were performed by users and not injected into the dataset or substituted from other legitimate users. Overall, we obtained 320 hours of data that included 18 hours of masquerader data and at least 2 instances of traitor data, with an average of 13 hours of per user participation. Moreover, during the competition some groups engaged in malicious activities different from the intended ones (masquerader and traitor). Although preliminary analysis have revealed a number of interesting events (denial of service attacks, masquerade period countermeasures, masquerader and traitor attacks), a deeper analysis is required to extract the hidden malicious events that lie within the dataset. In this sense, we plan to do a detailed analysis of the dataset, which will aim to show the distinction between the severity of user activities performed during different time periods.

ACKNOWLEDGEMENTS

This research was supported by ST Electronics and National Research Foundation (NRF), Prime Minister's Office Singapore, under Corporate Laboratory @ University Scheme (Programme Title: STEE Infosec - SUTD Corporate Laboratory).

REFERENCES

[1] Accenture. 2016. Accenture. (2016). https://www.accenture.com/t20160704T014005Z__w__/us-en/_acnmedia/PDF-23/Accenture-State-Cybersecurity-and-Digital-Trust-2016-Report-June.pdf#zoom=50 Accessed on Jul/2017.

[2] Amazon Web Services Inc. 2017. Amazon Elastic Compute Cloud (EC2). (2017). https://aws.amazon.com/ec2/ Accessed on Jul/2017.

[3] Amazon Web Services Inc. 2017. Amazon Web Servicies (AWS). (2017). https://aws.amazon.com Accessed on Jul/2017.

[4] Amazon Web Services Inc. 2017. Amazon WorkSpaces. (2017). https://aws.amazon.com/workspaces/ Accessed on Jul/2017.

[5] Maureen L Ambrose, Mark A Seabright, and Marshall Schminke. 2002. Sabotage in the workplace: The role of organizational injustice. *Organizational behavior and human decision processes* 89, 1 (2002), 947–965.

[6] Amos Azaria, Ariella Richardson, Sarit Kraus, and V. S. Subrahmanian. 2014. Behavioral Analysis of Insider Threat: A Survey and Bootstrapped Prediction in Imbalanced Data. *IEEE Transactions on Computational Social Systems* 1, 2 (jun 2014), 135–155. https://doi.org/10.1109/TCSS.2014.2377811

[7] BBC News. 2013. Bradley Manning sentenced to 35 years in Wikileaks case. (2013). http://www.bbc.com/news/world-us-canada-23784288#FBM276124 Accessed on July/2017.

[8] Ruth Ben'Ary. 1989. *Touch Typing in Ten Lessons: A Home-study Course with Complete Instructions in the Fundamentals of Touch Typewriting and Introducing the Basic Combinations Method.* Penguin.

[9] Oliver Brdiczka, Juan Liu, Bob Price, Jianqiang Shen, Akshay Patil, Richard Chow, Eugene Bart, and Nicolas Ducheneaut. 2012. Proactive insider threat detection through graph learning and psychological context. In *SPW - Proceedings - IEEE CS Security and Privacy Workshops.* 142–149.

[10] Chris Brook. 2016. Nuclear Power Plant Disrupted By Cyber Attack. (2016). https://threatpost.com/nuclear-power-plant-disrupted-by-cyber-attack/121216/ Accessed on July/2017.

[11] CALO Project. 2015. Enron Email Dataset. (2015). http://www.cs.cmu.edu/~enron/ Accessed on May/2017.

[12] Benito Camiña. 2014. WUIL dataset. (2014). http://homepage.cem.itesm.mx/raulm/wuil-ds/ Accessed on May/2017.

[13] Benito Camiña, Raúl Monroy, Luis A Trejo, and Erika Sánchez. 2011. Towards building a masquerade detection method based on user file system navigation. In *Mexican International Conference on Artificial Intelligence.* Springer, 174–186.

[14] J. B. Camiña, R. Monroy, L. A. Trejo, and M. A. Medina-PÃĭrez. 2016. Temporal and Spatial Locality: An Abstraction for Masquerade Detection. *IEEE Transactions on Information Forensics and Security* 11, 9 (Sept 2016), 2036–2051. https://doi.org/10.1109/TIFS.2016.2571679

[15] Dawn M Cappelli, Andrew P Moore, and Randall F Trzeciak. 2012. *The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes (Theft, Sabotage, Fraud).* Addison-Wesley.

[16] CERT. 2013. Insider Threat Tools - Dataset. (2013). https://www.cert.org/insider-threat/tools/ Accessed on May/2017.

[17] Eric Cole and Sandra Ring. 2005. *Insider threat: Protecting the enterprise from sabotage, spying, and theft.* Syngress.

[18] Alaa El Masri, Harry Wechsler, Peter Likarish, and Brent ByungHoon Kang. 2014. Identifying users with application-specific command streams. In *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on.* IEEE, 232–238.

[19] Á. Fülöp, L. Kovács, T. Kurics, and E Windhager-Pokol. 2016. Balabit Mouse Dynamics Challenge Dataset. (2016). https://github.com/balabit/Mouse-Dynamics-Challenge Accessed on May/2017.

[20] Ashish Garg, Ragini Rahalkar, Shambhu Upadhyaya, and Kevin Kwiat. 2006. Profiling users in GUI based systems for masquerade detection. In *Proceedings of the 2006 IEEE Workshop on Information Assurance*, Vol. 2006. 48–54.

[21] Joshua Glasser and Brian Lindauer. 2013. Bridging the gap: A pragmatic approach to generating insider threat data. In *Security and Privacy Workshops (SPW), 2013 IEEE.* IEEE, 98–104.

[22] Saul Greenberg. 1988. Saul Greenberg's homepage. (1988). http://saul.cpsc.ucalgary.ca/ Accessed on May/2017.

[23] Saul Greenberg. 1988. *Using unix: Collected traces of 168 users.* Technical Report.

[24] Frank L Greitzer and Thomas A Ferryman. 2013. Methods and metrics for evaluating analytic insider threat tools. In *Security and Privacy Workshops (SPW), 2013 IEEE.* IEEE, 90–97.

[25] Shuyuan Mary Ho, Jeffrey T Hancock, Cheryl Booth, Mike Burmester, Xiuwen Liu, and Shashank S Timmarajus. 2016. Demystifying insider threat: Language-action cues in group dynamics. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on.* IEEE, 2729–2738.

[26] Kevin S Killourhy and Roy A Maxion. 2009. Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on.* IEEE, 125–134.

[27] Terran Lane and Carla E Brodley. 1997. An application of machine learning to anomaly detection. In *Proceedings of the 20th National Information Systems Security Conference*, Vol. 377. Baltimore, USA, 366–380.

[28] Frank Linton, Deborah Joy, Hans-Peter Schaefer, and Andrew Charron. 2000. OWL: A recommender system for organization-wide learning. *Educational Technology & Society* 3, 1 (2000), 62–76.

[29] Michele Maasberg, John Warren, and Nicole L Beebe. 2015. The dark side of the insider: detecting the insider threat through examination of dark triad personality traits. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on.* IEEE, 3518–3526.

[30] Roy A Maxion and Tahlia N Townsend. 2002. Masquerade detection using truncated command lines. In *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on.* IEEE, 219–228.

[31] Microsoft. 2017. Microsoft Outlook. (2017). https://www.microsoft.com/en-xm/outlook-com/ Accessed on Jul/2017.

[32] Microsoft. 2017. Microsotft Exchange. (2017). https://products.office.com/en/exchange/microsoft-exchange-server-2016 Accessed on Jul/2017.

[33] MITRE Corporation. 2000. OWL Dataset: Usage of Microsoft Word Commands. (2000). http://research.cs.rutgers.edu/~sofmac/ml4um/data.html Accessed on May/2017.

[34] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. 1999. *Optical character recognition.* John Wiley & Sons, Inc.

[35] Oracle. 2017. MySQL. (2017). https://www.mysql.com/ Accessed on Jul/2017.

[36] Moses Palmér. 2017. Pynput. (2017). https://pypi.python.org/pypi/pynput Accessed on Jun/2017.

[37] Delroy L Paulhus and Kevin M Williams. 2002. The Dark Triad of personality: Narcissism, Machiavellianism, and psychopathy. *Journal of Research in Personality* 36, 6 (2002), 556 – 563. https://doi.org/10.1016/S0092-6566(02)00505-6

[38] Pennebaker Conglomerates, Inc. 2017. LIWC 2015. (2017). http://liwc.wpengine.com Accessed on July/2017.

[39] Purdue Millenium Lab. 1998. UNIX User Data. (1998). http://kdd.ics.uci.edu/databases/UNIX_user_data/UNIX_user_data.html Accessed on May/2017.

[40] Mark E Russinovich and Aaron Margosis. 2016. *Troubleshooting with the Windows Sysinternals Tools.* Microsoft Press.

[41] Malek Ben Salem, Shlomo Hershkop, and Salvatore J. Stolfo. 2008. A Survey of Insider Attack Detection Research. In *Insider Attack and Cyber Security*, Salvatore J. Stolfo, Steven M. Bellovin, Angelos D. Keromytis, Shlomo Hershkop, Sean W. Smith, and Sara Sinclair (Eds.). Springer US, Boston, MA, Chapter 4, 69–90. https://doi.org/10.1007/978-0-387-77322-3_5

[42] M Ben Salem and Salvatore J Stolfo. 2009. Masquerade attack detection using a search-behavior modeling approach. *Columbia University, Computer Science Department, Technical Report CUCS-027-09* (2009).

[43] Malek Ben Salem and Salvatore J Stolfo. 2011. Modeling user search behavior for masquerade detection. In *International Workshop on Recent Advances in Intrusion Detection.* Springer, 181–200.

[44] Salem, Ben. 2009. RUU dataset. (2009). http://sneakers.cs.columbia.edu/ids/RUU/data/ Accessed on May/2017.

[45] Matthias Schonlau. 2001. Masquerading User Data. (2001). http://www.schonlau.net/ Accessed on May/2017.

[46] Matthias Schonlau, William DuMouchel, Wen-Hua Ju, Alan F Karr, Martin Theus, and Yehuda Vardi. 2001. Computer intrusion: Detecting masquerades. *Statistical science* (2001), 58–74.
[47] Eric D Shaw. 2006. The role of behavioral research and profiling in malicious cyber insider investigations. *Digital investigation* 3, 1 (2006), 20–31.
[48] Robert W Shirey. 2007. *Internet security glossary, version 2.* RFC 4949.
[49] David A Solomon and Helen Custer. 1998. *Inside Windows NT.* Vol. 2. Microsoft Press Redmond.
[50] Squid-Cache. 2017. Squid Proxy. (2017). http://www.squid-cache.org/ Accessed on Jul/2017.
[51] Paul J Taylor, Coral J Dando, Thomas C Ormerod, Linden J Ball, Marisa C Jenkins, Alexandra Sandham, and Tarek Menacere. 2013. Detecting insider threats through language change. *Law and human behavior* 37, 4 (2013), 267.
[52] Tcpdump. 2017. Tcpdump. (2017). http://www.tcpdump.org/ Accessed on Jul/2017.
[53] Verizon. 2016. 2016 Data Breach Investigations Report. (2016). http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/ Accessed on Jul/2017.
[54] Windows. 2017. Process Monitor. (2017). https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx Accessed on Jul/2017.

A APPENDIX

1st Wild-Card Period

Attacking Team	Victim	Mouse	Keyboard	Network (MB)	Emails
Team3	User2	133946	3989	14.43	2
Team4	User4	12526	863	1.79	2
Team1	User9	140393	1175	1.14	0
Team2	User12	24998	4421	42.50	0
Team6	User17	54271	1421	1.76	3
Team5	User20	144190	3162	8.56	13

2nd Wild-Card Period

Attacking Team	Victim	Mouse	Keyboard	Network (MB)	Emails
Team2	User1	17722	535	2.26	2
Team1	User6	94336	2661	2.21	8
Team6	User7	30954	302	1.05	0
Team5	User13	95751	4247	1.81	10
Team4	User14	97086	1457	2.53	0
Team3	User21	211974	4816	2.33	12

Traitor Session

Fired Member	New Team	Mouse	Keyboard	Network (MB)	Emails
User3	Team6	26324	261	0.96	1
User8	Team2	0	0	1.04	0
User15	Team3	0	0	0.43	0
User18	Team5	12405	1995	2.46	3
User24	Team1	0	0	0.45	0

Table 1: Summary of malicious data entries

User	Mouse	Keyboard	Network (MB)	Email
User2	1064997	70210	159.36	428
User1	2219725	194326	186.93	1103
User24	21289	2005	15.27	1
User19	2684961	135483	121.89	524
Team1	**5990972**	**402024**	**483.45**	**2056**
User8	218633	21882	614.19	109
User4	106775	14042	91.16	32
User5	55809	21006	25.15	68
User6	524118	47928	105.03	168
Team2	**905335**	**104858**	**835.53**	**377**
User22	2868648	134903	98.55	847
User7	2028454	116065	143.38	1369
User15	313870	5581	55.45	3875
User9	3308780	165331	241.8	1578
Team3	**8519752**	**421880**	**539.18**	**7669**
User10	1306712	101172	119.02	576
User13	2620107	106633	69.22	366
User12	1819012	357330	360.96	1279
User11	268994	69672	219.03	144
Team4	**6014825**	**634807**	**768.23**	**2365**
User14	2537664	65236	870.33	14494
User18	704609	54856	79.4	214
User16	626525	25938	107.3	517
User17	636293	44449	204.08	10241
Team5	**4505091**	**190479**	**1261.11**	**25466**
User3	396023	10059	77.3	83
User20	372911	28611	229.95	82
User21	316442	27058	71.54	55
User23	90686	11533	45.48	27
Team6	**1176062**	**77261**	**424.27**	**247**

Table 2: Summary of all collected data entries