The Wolf Of SUTD (TWOS): A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition

Athul Harilal[1]*, Flavio Toffalini[1], Ivan Homoliak[1],
John Castellanos[1], Juan Guarnizo[1], Soumik Mondal[1],and Martín Ochoa[2]
[1]*ST Electronics-SUTD Cyber Security Laboratory,*
Singapore University of Technology and Design, Singapore
{athul_harilal, ivan_homoliak, mondal_soumik}@sutd.edu.sg
{*flavio_toffalini, john_castellanos, juan_guarnizo*}*@mymail.sutd.edu.sg*
[2]*Department of Applied Mathematics and Computer Science,*
Universidad del Rosario, Bogotá, Colombia
martin.ochoa@urosario.edu.co

Abstract

In this paper we present open research questions and options for data analysis of our previously designed dataset called TWOS: The Wolf of SUTD. In specified research questions, we illustrate the potential use of the TWOS dataset in multiple areas of cyber security, which does not limit only to malicious insider threat detection but are also related to authorship verification and identification, continuous authentication, and sentiment analysis. For the purpose of investigating the research questions, we present several state-of-the-art features applicable to collected data sources, and thus we provide researchers with a guidance how to start with data analysis. The TWOS dataset was collected during a gamified competition that was devised in order to obtain realistic instances of malicious insider threat. 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 two types of malicious periods that was intended to capture the behavior of two types of insiders – 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. Their activities were monitored by several data collection agents and producing data for mouse, keyboard, process and file-system monitor, network traffic, emails, and login/logout data sources. In total, we obtained 320 hours of active participation that included 18 hours of masquerader data and at least two instances of traitor data. In addition to expected malicious behaviors, 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 was made publicly accessible for further research purposes. In this paper we present the TWOS dataset that contains realistic instances of insider threats based on a gamified competition. 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 total, we obtained 320 hours of active participation that included 18 hours of masquerader data and at least two instances of traitor data. In addition to expected malicious behaviors, 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. Furthermore, we illustrate the potential use of the TWOS dataset in multiple areas of cyber security, which does not limit to malicious insider threat detection, but also areas such as authorship verification and identification, continuous authentication,

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 9:1 (March 2018), pp. 54-85
*Corresponding author: ST Electronics-SUTD Cyber Security Laboratory, 8 Somapah Road, Building 2 Level 3 S(487372), Tel: +65-6486-7033/44, Web: `http://cyberlab.sutd.edu.sg/`

and sentiment analysis. We also present several state-of-the-art features that can be extracted from different data sources in order to guide researchers in the analysis of the dataset. The TWOS dataset is publicly accessible for further research purposes.

Keywords: malicious insider threat, masquerader, traitor, multiplayer game, user behavior monitoring, feature extraction, authorship verification, continuous authentication, sentiment analysis.

1 Introduction

In today's world, insiders have the potential to bring harm to the organization in which they work [1, 2, 3]. Considering knowledge of an insider, Salem et al. [4] 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 [5]. 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 [6, 7], biometric behavior by means of mouse [8], and keyboard usage patterns [9], 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. There are a number of challenges with obtaining high-quality datasets. Usually information about incidents of real insider attacks are kept confidential, since revealing the details could harm the reputation of the organizations involved [1]. 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 [1], 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. [10], 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 [11] and Purdue University [12] datasets of Unix commands, MITRE OWL [13] 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 [14]) or by interactive (e.g., RUU dataset [15]) simulation of malicious intent. The details of these datasets can be found in Section 8.

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 real one from Enron [16] company and the CERT datasets [17] that contain simulated traitors.

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 [4] (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). Among 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. [4], 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.

Contributions. We designed a multi player game called as **T**he **W**olf **o**f **S**UTD (TWOS). It encouraged user interactions in a simulated corporate environment and it's purpose was to collect a comprehensive dataset containing interactive malicious insider threat instances 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, SMTP logs (email bodies and other meta-information), login/logout of users, and psychological questionnaires. 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. The instructions for getting the dataset are available at `http://cyberlab.sutd.edu.sg/twos-dataset`.

The current paper is an extension of our previous work [18]. We extend the paper as follows: 1) We extend the list of notable events observed in order to enrich the understanding of the raw dataset. 2) In order to aid researchers in data analysis, we present possible options for state-of-the-art feature extraction for different data sources contained in the dataset. 3) We illustrate the potential use of TWOS dataset in multiple areas of cyber security, and we specify open research questions. 4) Based on preliminary analysis of dataset, we provide statistics about missing data.

2 Attacker's Model

According to [19], we can classify an "insider attack" as one that is initiated by an entity already inside the security perimeter of the system. It may refer to an entity that was previously authorized to access system resources but uses them in an inappropriate way. The insider attacker's model that we consider in this paper is based on categorization made in [4], which defines masquerader and traitor.

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 to perform malicious actions using the available privileges and information resources. However, such an attacker might have less knowledge about the system that is under attack. So, he may need to search through the system to identify valuable assets [4]. Note that masquerade attackers is a subclass of the identity theft problem. The identity theft problem may be either internal or external according to the source of the attack. External cases of the identity theft problem involve obtaining victim's credentials in different ways, such as phishing or social engineering. On the other hand, internal cases (performed by insiders), involve escalating the privileges through exploitation of some vulnerability or weak/broken/buggy access control mechanisms. In our designed game, we incentivized situations belonging to the internal identity threat problem whereby masqueraders try to perpetrate data exfiltration or unauthorized modification of information (sabotage).

Traitor. A traitor represents an attacker who knows the targeted system and has inherent access to information resources that are subject to an insider attack. This attacker uses his own legitimate credentials to perform malicious actions [4]. Attacks performed by traitors are much more complicated to detect due to fine-grained deviations from their normal behaviors. In our designed game, we incentivized traitor behavior involving data exfiltration upon leaving a company.

3 The Game

The purpose of our game was to obtain labeled data containing both normal user behavior and malicious insider behavior. One important characteristic of such data is that it should be as *realistic* as possible: our intention was to devise an experiment that stimulates participants into solving a task collaboratively (yielding normal or non-malicious behavior), and at the same time it will provide scenarios where participants have a certain incentive to *cheat*, and thus behaving as malicious insiders. We designed our game in accordance to the setting of similar companies that try to win over customers who buy companies' products. For simplicity, the companies offered similar products and they offered 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 by an automated script that was developed for the experiment. More specifically, we created a bucket of synthetic customers (10.000 entities), which was shared with all teams at the beginning of the competition. Each customer had a few points (similar to amount of money) that were given to students upon fulfilling a predefined set of operations (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 setting of 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 restrict a number of outgoing channels, such as USB drives, in order to constrain the communication channels through which students could transfer and share information. Note that the game was designed to preserve the anonymity of the participants, and therefore no real student IDs or names were stored. Moreover further anonymization of the data was performed (described in Section 5) in order to comply with the privacy policies of Institutional Review Board (IRB) at SUTD.

Overall, the competition lasted for five days (from Monday to Friday), and the top 3 teams (based

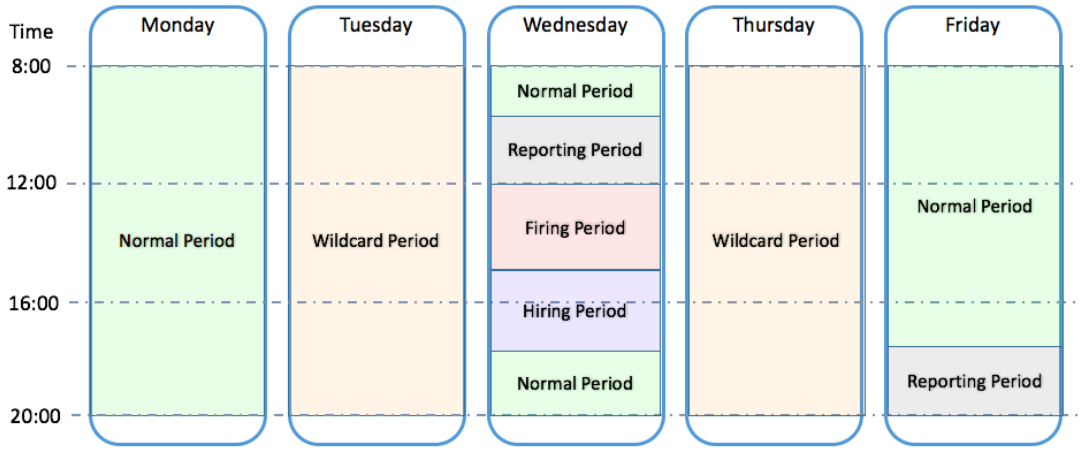


Figure 1: Competition schedule with stages, every period is drawn in a different color

on collected points from customers) were rewarded. The top 3 teams earned $800 *SGD*, $400 *SGD*, and $200 *SGD*, respectively. Each student had also earned $15 for his/her active participation. During the competition, students were expected to play for at least 10 hours. If they failed to do so, their prize money was proportionately reduced to the amount of time they played, which was computed based on the mouse and keyboard activities (see Section 5). Hence, the students were financially motivated to actively participate and win 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 a team was able to access the victim's machine. In particular, we describe the periods as follows:

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.

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 an attack was chosen to ensure the availability of at least 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 random 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). 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.

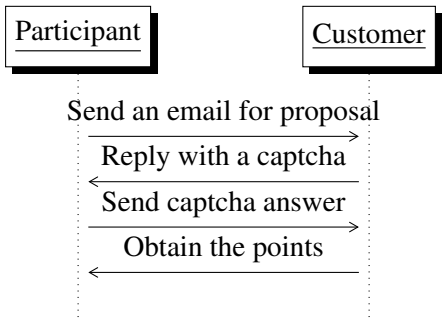


Figure 2: Interaction between a participant and a customer

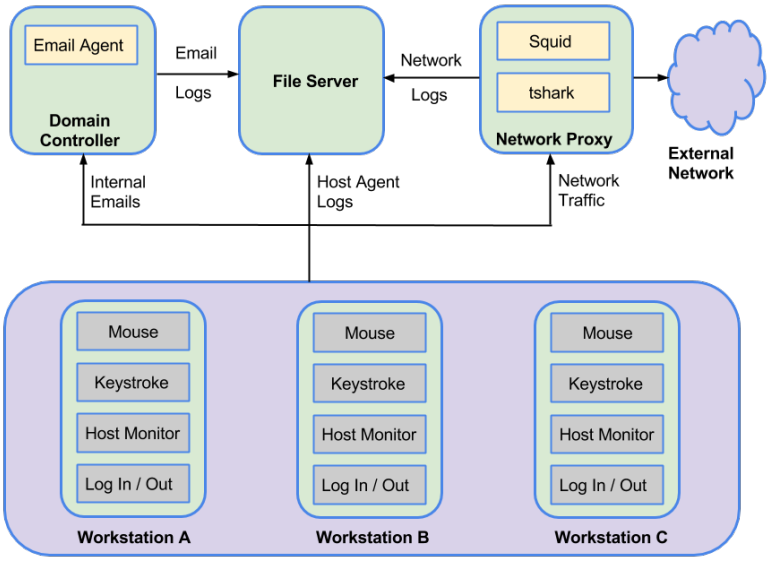


Figure 3: Architecture of infrastructure

4 Implementation of Data Collection

We set up the infrastructure for the competition on the cloud using Amazon web services [20] (see Figure 3). The infrastructure included 3 Amazon EC2 servers [21] and 24 Amazon WorkSpace instances [22]. 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.

The workstations were set up with a standard software suite (*i.e.,* MS Office, Mozilla Firefox and Microsoft Outlook). We also provided a private email address to every participant for internal communications. All participants' 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 [23] that 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 [24]. The host monitor agent was responsible for logging system calls generated by each Amazon WorkSpace. We chose Process Monitor [25] as a host monitor agent, since it is a standard Windows tool for forensic and system analysis [26].

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 [27] 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* [28] 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 [29]. For contacting the customers, participants used Microsoft Outlook [30]. 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* [31] 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 enabled an external communication were HTTP and HTTPS protocols. Other communication channels of the workstations, such as shared clipboard, were explicitly disabled.

5 TWOS: The Wolf of SUTD Dataset

In this section we describe the data collected from the competition and its 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.

5.1 Description of Data

We collected activities performed by students from 7 different data sources that were logged by our architecture. The structure of each data source and its anonymization mechanism are described as follows.

Keystrokes. 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 (e.g., extracting passwords, telephone numbers), we employed an anonymization process that allowed us to preserve as much information as possible and at the same time, it made reassembling of the original text challenging. We accomplished it by taking inspiration from typewriting [32], 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. The mapping of the keyboard layout is depicted in Figure 4.

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 a monitor's resolution. Because of the nature of these data, we did not employ any anonymization mechanism. This data source primarily serves for the purpose of identifying the normal user based on behavioral biometrics.

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

Figure 4: Mapping for anonymization of characters

Host monitor Logs (Process and File-System Monitor). The main information provided by the Host Monitor was related to file accesses (*e.g.,* open/read/write/rename/delete/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 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 Traces. We decided to parse only HTTP protocol because it was the only protocol used by students for interaction with the 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 the 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).

SMTP Logs – Email Bodies. In this data source, 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. The anonymization of email body was particularly important. Although students were thoroughly explained about not revealing their true identities or any personally identifiable or sensitive information while playing, there were a few such instances when the students revealed some personally identifiable information (e.g., revealing their name at the end of the email, trying to send emails from the game account to personal accounts). Since the email body could contain sensitive in-

formation of students, we transformed the email body using existing Linguistic Inquiry and Word Count (LIWC) tool [33], which generated 94 features expressing membership ratio to each of 94 word groups (e.g., anxious, angry, negative). Note that only recently written text by an originator of an email was included for the LIWC feature extraction, and we omitted the history of a conversation.

SMTP Logs – Meta-Information (*Excluding Email Bodies*). We decided to create dedicated data source for meta-information extracted from SMTP logs, beside the data source for email body. Meta-information extracted from SMTP logs represent information from headers, attachments, senders, recipients, and subject of an email. Subjects of emails were anonymized using table-based mapping for words not present in standard English dictionary and white-list of paths shared across data sources.

Logon/logout Activities. We monitored users' login/logout activities using Windows event log [26]. 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 Questionnaires. Taking inspiration from the previous works [34, 35, 36, 37, 1, 38], we asked the participants to fill up a psychological questionnaire. We used the questionnaire inspired by dark triad theory [39], which contained 50 questions. This questionnaire may enable researchers to correlate participants' behavior with psychological indicators.

5.2 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.

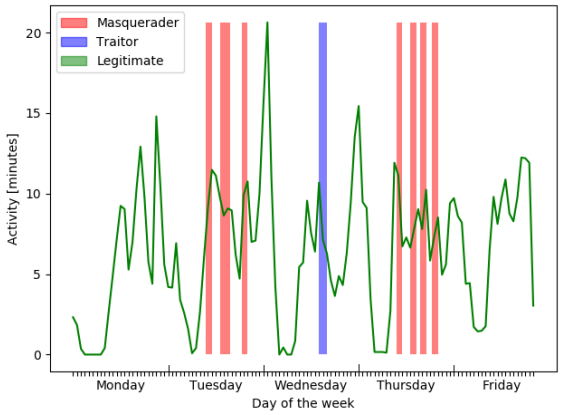
Statistics of Users' Activity. Average activity of all students is illustrated in Figure 5, which depicts active participation from three points of view (*i.e.,* keyboard/mouse, network traffic, and emails). All plots represent the activities of the whole week and they further show the phases of the 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 usually dropped after midnight and rose again in the morning. There is also a significant peak on Wednesday, just before the first scoring period. This can be explained by the fact that teams tried to gain as many points as possible before the first scoring period. Other important spikes were observed during the wild-card periods, which indicates active participation by students during these periods.

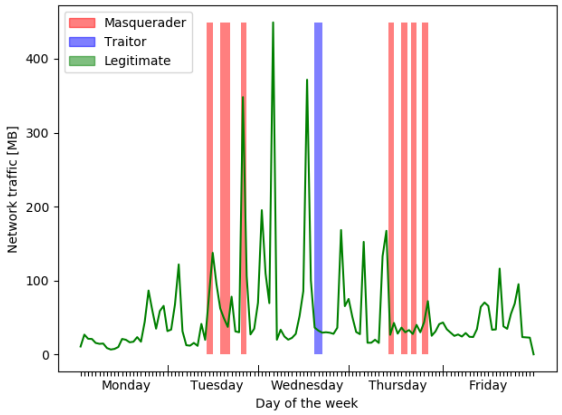
Average per user network traffic transferred through Network Proxy is depicted in Figure 5b. Amount of network traffic is an important measure because this was the only channel for exfiltration of data outside of user machines according to our architecture. We remind that participants were restricted 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 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 defensive techniques employed by students. The last important observation can be seen before the first 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.

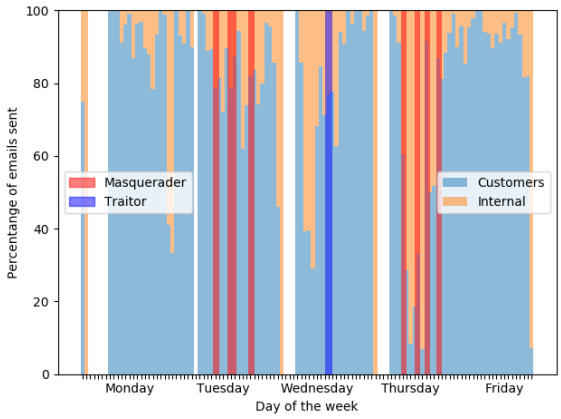
We illustrate in Figure 5c, the percentage of emails sent by participants toward the customers against emails sent for internal communication among the participants or technical support. If no emails were sent in a certain period of time, then we set both bars to *zero*. Although for the majority of time most of the emails were sent toward 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 first scoring period and during the second wild-card period. Moreover, we also observed an increasing number of internal emails at the midnight of Tuesday and also at the final



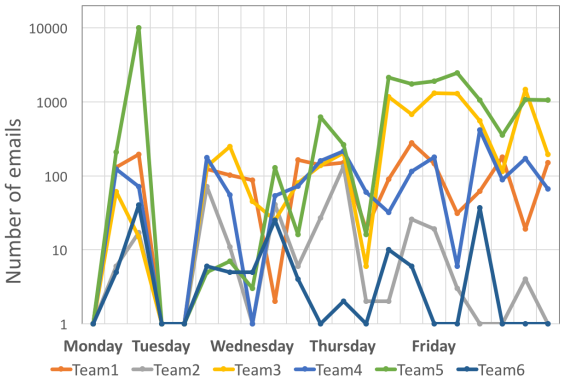
(a) Average keyboard/mouse activity per hour



(b) Average network traffic activity per hour



(c) Percentage of emails sent to customers or internally



(d) Number of emails sent to customers per team

Figure 5: Competition statistics

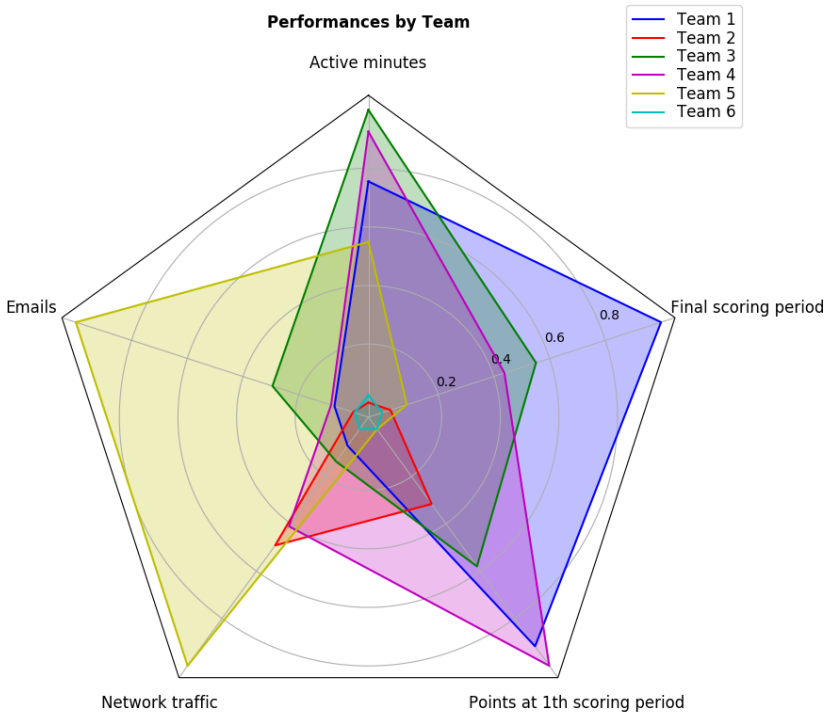


Figure 6: Comparison of all teams

stage of the competition because of the final scoring period.

In Figure 5d, we depict the number of emails sent to customers per team. This graph shows two particular behaviors. First, Team 5 sent a significant number of emails at the beginning of the game (over 10 thousand on Monday evening). It was a part of the team's strategy for obtaining points from as many customers as possible. It caused a Denial of service attack in the email server, which is closely described in Section 5.3. Second, teams became more active by the end of the game (Thursday evening and Friday), reaching levels of thousands of emails sent per team. In particular, teams that were leading in the competition exhibited such type of behavior.

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 figure, we emphasize the 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 ranked the 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; the team members managed to automate the process of sending emails to customers in order to receive the captchas (see Section 5.3). 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 a 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, most of the teams showed a similar type of behavior, where one user played significantly more than others.

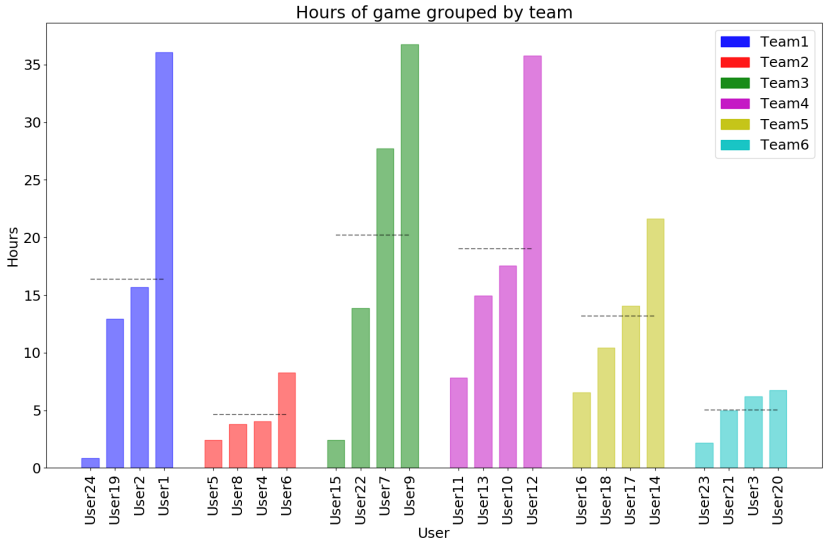


Figure 7: Total user activity

Analysis of Cheating. We analyzed the score files sent 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 teams that did not submit anything.

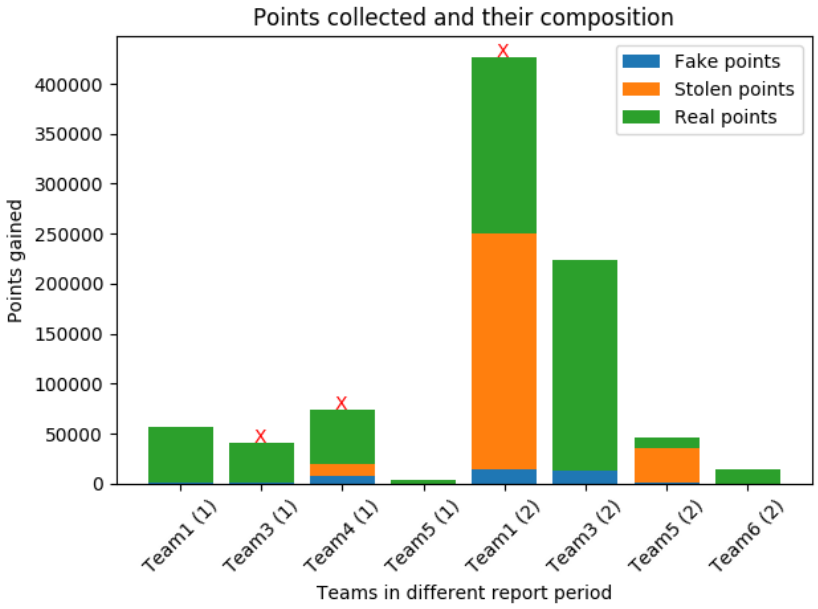


Figure 8: Composition of score per period

Vertical bars represent the amount of points gained in total, where green colored bars represent points collected by contacting customers, orange colored bars represent stolen points, and blue colored bars represent 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 (i.e.,Team 1) managed to steal more points than others.

5.3 List of 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 network monitoring; the scope of these events include both malicious and normal periods. Since some of those events 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 which event and when the event happened. In contrast to our previous work [18], we extend the list of notable events list by additional two events: abnormal communication among teams and collective effort of masquerading teams.

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 a body of the first 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 interaction 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 (see the spike over 10 thousands emails sent on Monday evening in Figure 5d), 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 [40]. 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.

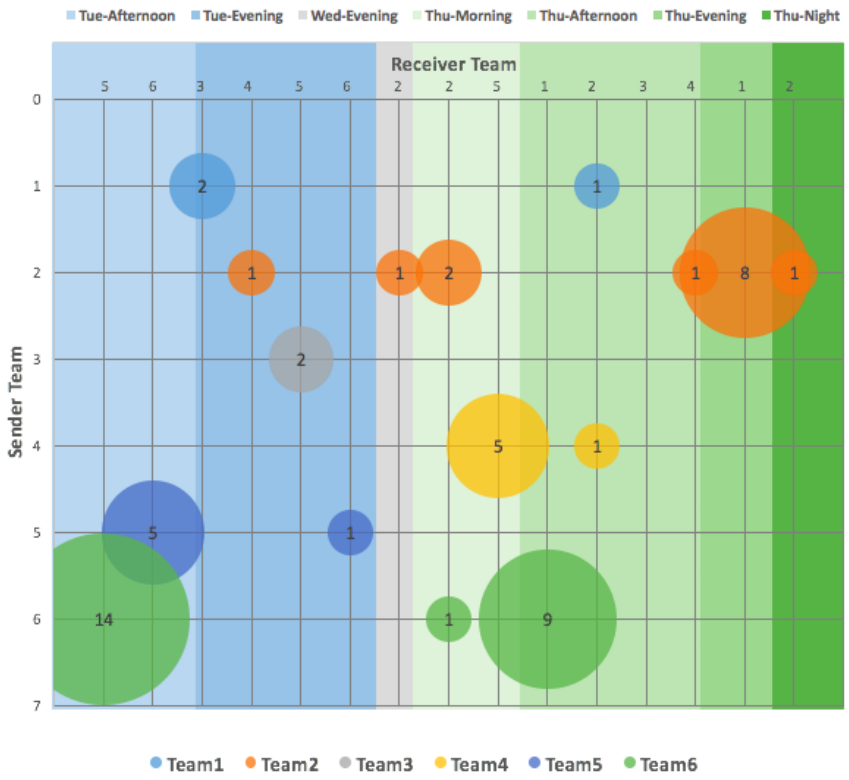


Figure 9: The amount of exchanged emails among teams

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.

Abnormal Communication among Teams. Between Tuesday afternoon and Thursday night, the analysis shows that teams Two and Six were actively trying to contact others. The analysis highlights a very active interaction between teams Five and Six on Tuesday afternoon. Team 6 send 14 messages and in return Team 5 replied to Team 6 five times during the same period (see bottom left on Figure 9). The analysis also shows interactions between attacker and victim teams, during and after attack periods (i.e. Tuesday evening team One contacts team Three members, top left in the graph).

These abnormal behaviors enrich the dataset, adding features that were not part of the initial design of the game, similar to communication among rival companies. All these diverse communication patterns will open new opportunities for further research focused on the development and testing of more complex detection mechanisms such as *traitor monitors*. This diversity makes the dataset more complete and realistic in comparison with other datasets in the literature.

Collective Effort from Masquerading Team. During the masquerade period, a masquerading team was enabled to perform malicious activities by attacking a victim account of another team. The game allowed only a single masquerader from the masquerading team to attack the victim machine. Hence other members from the masquerading team had to either wait or coordinate with the masquerader that takes over the victim account in order to expedite the process of getting valuable information. We found a few such instances where the masquerading team members coordinated during the masquerade period. For instance, a few members of Team 5 (first masquerade period) and Team 3 (second masquerade period) coordinated between themselves through sharing of information based on what was seen in the victim's machine. It resulted in exfiltration of important files from the victim's machine.

5.4 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.

5.5 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 for the future design of such experiments. When we observed the Denial of Service attack mentioned in Section 5.3, 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.

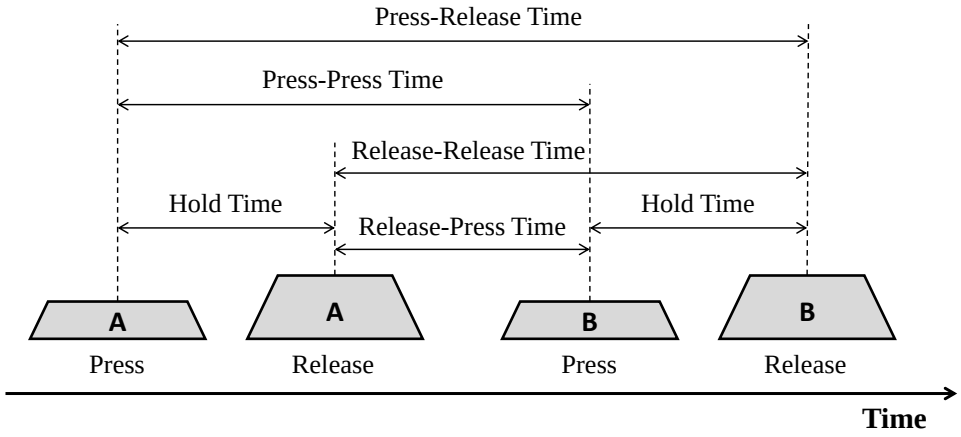


Figure 10: Low-level keystroke dynamics features

6 Preliminary Data Analysis and Feature Extraction

In this section, we move our focus on a description of the state-of-the-art features that can be extracted over particular data sources of the dataset. Moreover, when dissecting options for feature extraction per particular data source, we also outline which other data representations/fields we plan to publish in the future, which are additional to the ones introduced in [18].

Keystrokes. Raw data representing different keystroke events has been collected and enable to extract almost all the state-of-the-art features ingesting respective events. For keystrokes, we captured key press and release time for individual keys. Therefore, we can obtain the single-key feature (*i.e.,* Hold Time) and all of the digraph features (*i.e.,* Press-Release Time, Press-Press Time, Release-Release Time and Release-Press Time) proposed in [41, 42]. Pictorial representation of such low level features is depicted in Figure 10. From this figure we can understand that the keystroke Hold-Time feature is the time difference between a key press and a key release time for a given keystroke (in the Figure 10 it is for key 'A' and 'B'). We can also understand the keystroke digraph features from this figure, where we have shown an example for a key digraph *AB*. These features are mostly used for keystroke dynamics based behavioral biometrics. Also, these features can be used for extraction of various higher-level statistical features associated with a user's behavior. However, in our dataset, we have email data that are written in a proper English language, therefore we can use some stylometry features for authorship verification as well, which are explained later in this section.

Mouse Traces. We have captured the mouse trajectory and mouse click data while users interacted with their system. Therefore, all the state-of-the-art mouse features can be extracted from our dataset. According to the literature, the mouse features can be divided into two parts [43, 44]:

- ***Schematic Features:*** These features characterize the constituents of mouse actions during GUI interactions such as the statistical distribution of mouse action types or mouse pointer positions. There are four different schematic features we can generate from the raw data that are used in the literature:

  1. Mouse action histogram: statistics of occurrences of various mouse action types.
  2. Percentage of silent periods: statistics of idle time of mouse.
  3. Distribution of cursor positions on the screen.

4. Distribution of movement distances per direction.

- ***Motor-Skill Features:*** This group of features characterize the efficiency, agility, and motion habits of individual mouse actions such as the acceleration pattern or the speed of a double-click. There are five different motor-skill features we can generate from the raw data that are used in the literature:

    1. The elapsed time of single click: time interval between mouse button down and button up of a click.
    2. Elapsed time of double-click: overall time and three internal intervals between the button downs and ups of a double-click.
    3. Average movement speed compared to the directions: average movement speed calculated for different directions.
    4. Average movement speed and acceleration compared to travelled distance: average speed/accelerations calculated for different distance travelled.
    5. The transition time of actions: transition time between following mouse actions.

Host Monitor Logs. From this data source, it is possible to extract features related to processes running on the host machine and the file system activity of the corresponding user. It can be used to detect changes from the normal behavior of the user interaction with the host machine. From the literature we observe that *(Number of New Processes, Number of running processes on the system, Number of Document editing applications)* are some of the features that have been used to profile process behavior due to user interaction [45]. Regarding the file system features, a number of them exists in the literature *(Number of File system accesses [46, 47], Number of PC's with file accesses, Number of distinct file accesses, Number of after hour file accesses [45])*. These file system features are used to create a normal user profile based on the file access patterns of a normal user without storing information regarding the specific files that were accessed. However more advanced file system features capture the set of related files accessed within a user directory [6]. By linking a group of files that are accessed together, it is possible to detect malicious insider behavior through changes in the group of files accessed together. From our host monitor data source, it is also possible to construct such features to evaluate user behavior at a certain instance.

Network Traces. Using collected tcdpdump traces, it is possible to extract several state-of-the-art network-level features (their subsets, respectively), originally intended for network traffic classification or intrusion detection, such as KDD Cup '99 features [48], network discriminators for flow-based classification [49], Kyoto 2006+ features [50], or Advanced Security Network Metrics (ASNM) features [51].[1] These features are extracted over network sessions (represented by TCP connections) having the same pairs of IP addresses and ports in packet headers. From mentioned feature sets, it make sense to consider only particular features that can be used for profiling of normal user behavior in terms of accessing different (or the same) hosts in different (or close) times – primarily taking into consideration session durations, and various payload lengths. Examples of useful features for profiling of usual user session are or can be derived from: ratio of connections to different hosts, the number of data bytes from source to destination, median of packet sizes in outbound traffic of a connection, approximation of inbound/outbound flows of communication by polynomial of n-th order in the index domain of packet occurrences, Fast Fourier Transformation (FFT) of inbound/outbound packet sizes, etc. Note that application level

[1] Note that the full list of all these feature sets is present in appendix of [52].

features of the ISO/OSI model are constrained only to the ones that we provide as the replacement of the packet payload – for example *content-based* features of KDD Cup '99 [48] or *additional features* of Kyoto 2006+ [50] feature sets cannot be derived; however, there is no need to extract them as they are intended to intrusion detection and not user profiling. Proposal of simple HTTP-based features monitoring based on similarity of group of users behavior is present in [45].

Also note that we plan to extend the JSON data present in the payload of the packets in order to include query string of a URL, which will be anonymized in the similar vein as in the case of host monitor logs or email bodies – using shared table-based mapping. Then, even graph-based HTTP feature extraction, such as the one used in [53], will be possible to employ.

SMTP Logs – Email Bodies. As we have already mentioned, in [18] we extracted state-of-the-art LIWC features over email bodies, as due to potential privacy issues we did not intend to publish the whole content of the email bodies, even in anonymized form. To some extent, the LIWC features are useful for example in sentiment analysis of the text written by the author of an email, estimation of personality characteristics, or mental disorders; all of which may be the precursors of insider threat behavior. All such features belong to *motive* category of MOC features (motive, opportunity, capability) from the categorization presented by Gheyas and Abdallah in [54]; the paper presents also other features that can be derived from LIWC features. Since the style of writing can be quite unique to an individual, this measure can also be used to detect malicious insider anomalies. Additionally to LIWC features, we plan to extend the dataset of features by *(Number of Exclamations **(!)**, Number of Questions **(?)**, Number of Dashes **(-)**, Number of Double Dashes **(–)**, Number of Semi Colons **(;))*** [55], which can be used to profile normal user behavior based on the writing style considering mentioned diacritics. The email logs (metadata) along with LIWC features extracted from body can also be used for detection of colluding traitors, based on some indicators hinting a group of participants, such as a communication of a fired member with the previous team, but we did not confirm such cases yet. Such group of participants may result to th exchange of emails that could contain sensitive information or they could exchange emails expressing emotional aspects. Features such as *(Number of Words expressing achievements or rewards, Number of companies involved into the communication)* can be used for this purpose. In addition to [18], we plan to extend the dataset of emails by publishing the anonymized text of bodies using replacement of words not present in the standard English dictionary and using our white list, which is based on mapping that is shared across all data sources. In the similar vein, we also plan to anonymize and include query string of URLs occurred in the text of emails, which is currently discarded. This is important for a potential matching of URL strings contained in the email bodies and the HTTP request issued by users.

SMTP Logs – Metadata *(Excluding Email Bodies).* From the collected email logs representing metadata about the communications, it is possible to profile user behavior in order to detect both masqueraders and traitor instances. For example, features such as *(Number of Daily Sent / Read Messages, Number of Sent / Read Messages at Night, Number of Sent / Read Messages in the Morning)* [55] can be used to detect anomalies based on statistical differences deviated from a normal user behavior.

6.1 Drawbacks of the TWOS Dataset

Here, we describe several drawbacks that are related to missing data.

Keystrokes. From the preliminary analysis of the keystroke dataset, we found some missing data entries related to key press and release events. Previously, we have mentioned that dataset has been collected from 24 participants for five days/sessions and therefore we have $24 \times 5 = 120$ sessions in total. However, we found out that 28 sessions have less than 500 keystrokes, which is less than the average

number of keystrokes computed over all users per day (session). We also found out that the keystroke data from two participants (User15, User24) are unusable as all of their sessions have inadequate data. These users were active only for a few time instances on certain days of the competition. Hence, these participants did not play for all the days and had lesser than 500 keystrokes per a day. Furthermore, we found out that 14 participants have at least one session with less than 500 keystrokes. We have also found that each session has on an average 11.7% of missing events (keystroke actions that were performed by a user but not logged by the logger). In some instances, a key-press event was recorded, but there was no corresponding key-release event that was captured. The opposite case was also observed. Out of 11.7% missing events, only 1.2% of the data corresponded to alphanumeric characters and the remaining 98.8% of missing data were *Modifier keys* or *Cursor keys* etc. Hence, the keystroke agent that intercepts keystroke events of users need to be revised in order to resolve these issues.

Host Monitor Logs. During the competition, the logging agent of host monitor experienced several outages across particular users. This resulted into some missing data of normal period but also in wild-card periods; firing period did not suffer from any missing data. Considering normal periods of users who actively played, the most common amount of missing data is equal to 5% of all data collected from a particular user – associated users are User9, User12, User13, and User17. Then other active users (User1, User6, User14) had around 3% of missing normal data, while 1% and lesser of missing data occurred in data collection of users User2, User7, and User21. Note that in the case of User15, who was almost all the time of the competition inactive, there were 23% of missing normal data. Regarding wild-card periods, there were two users (User2 and User13) who have missing data thanks to 1 hour outage of the process monitor logger in each case. This resulted into the lost of around 30 minutes of masquerade data for each user. In the future work, the host monitor agent has to be revised and made more stable.

7 Envisioned Use of the TWOS Dataset

This dataset is primarily designed for, but not limited to, the malicious insider threat research. It can be used to detect malicious insider activities by looking into one source of data (e.g., mouse logs, keystroke logs, etc.) or a combination of such data sources. The advantage of using combination of more data sources as compared to a single data source is that the former can capture time correlations among anomalies/behaviors spread across multiple data sources. For example when a malicious insider (traitor) sends an email to an outsider, his mouse activity and keystroke activity might look similar to his normal profile, and therefore data from such sources might not yield any anomaly. However, malicious insider activity could be caught due to abnormal email activities arising from anomalous stylometry features, due to the receiver's email address belonging to the competing company, or due to anomalous number of files attached, etc.

Figure 11 shows a basic block diagram representation of our intended risk assessment system for insider threat. After getting access to the device, a given user's activity may generate different events, either keystrokes, mouse, host monitor, network traffic, emails, or a combination of one or more of them. Events generated from one or more data sources can be used to analyze user activity in order to decide if the events were generated by a malicious insider or a genuine user. Furthermore, events can be analyzed based on a time window-based approach (analyzing user events regularly at fixed time intervals) or event-based approach (where the detector waits for the fixed number of events before analyzing them). These events are then input into their respective modules (*i.e., Keystrokes (KS)*, *Mouse Traces (MT)*, etc.) for *Feature Extraction Module* and then to the *Comparator Module* (*i.e., KS Comparator*, *MT Comparator* etc.) for comparison with the stored genuine user's profile and generation of the risk scores. After getting a risk score from the respective comparator, we will combine all these scores to produce a final risk score and based on a threshold, the *Decision Module* will decide whether the present user is genuine or not.

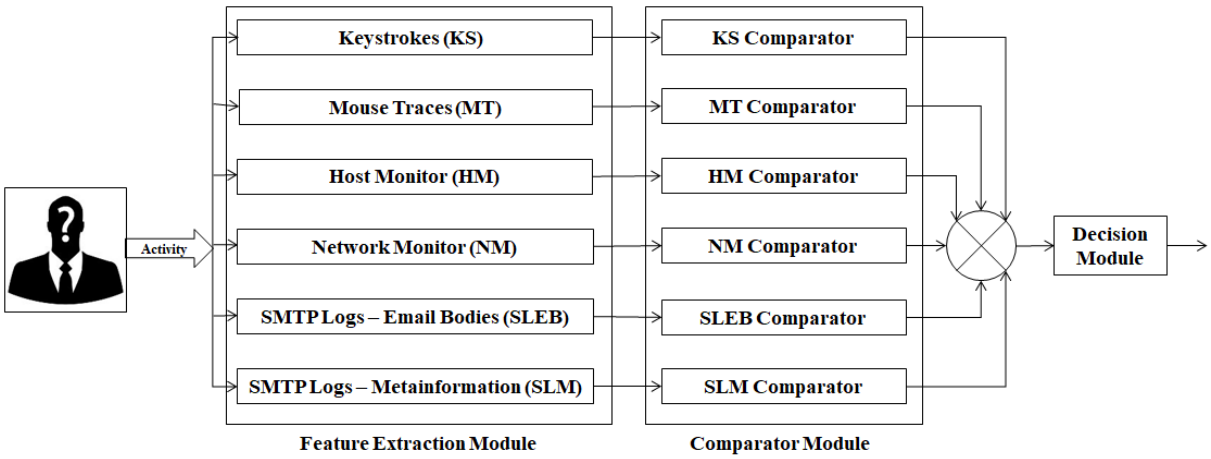


Figure 11: Block diagram representation of our intended system

7.1 Research Questions Related to Malicious Insider Threat

The above envisioned system produces several open research questions for the future research using the TWOS dataset. Some of these research questions are listed below:

- What is the optimized feature set that can be used for different events to achieve the best performance?
- What will be the best comparator for the respective events?
- What are the possibilities to combine all the risk scores?
- Can the decision-making threshold be predefined, or it has to be dynamically changed based on the scenario?

7.2 Collateral Research Questions

As we have already outlined, the TWOS dataset can also be used for exploring other research problems than malicious insider threat. In this section, we provide examples of other security-related areas with some research questions specified.

Authorship Verification and Identification. Considering the text of SMTP logs containing anonymized email bodies, authorship verification and identification problems can be addressed. Although these problems may look related to the masquerader problem, we did not discover instances of masquerader sessions containing some communication on behalf of the victim – also time duration of masquerade sessions were know to the all participants, and therefore such emails could be easily recognized. For the authorship verification and identification problems, we state a few research questions:

- What authorship verification/anomaly detection techniques are the best performing in this one-class problem?
- To what extent can anonymized email messages be replaced by LIWC features for the authorship verification problem?

- What authorship identification techniques are the best performing in this multi-class problem?
- To what extent can anonymized email messages be replaced by LIWC features for the authorship identification problem?

Continuous Authentication. People use access control mechanisms to protect against unauthorized access by another person. This means that a user needs to give proof of his/her identity when starting or unlocking a device. However, in many cases people leave the computer physically unattended for shorter or longer periods when it is unlocked and this is prone to session hijacking. To protect the devices from session hijacking, we need to monitor the user's activity continuously and establish the authenticity of the activity, which is termed as *Continuous Authentication (CA)* [42].

Considering only legitimate data of our dataset, in particular mouse and keystroke data sources, it is possible to address the CA problem. For this problem, we formulate several research questions:

- What are the best performing CA methods working over individual data sources (mouse and keystrokes) and their combination respectively?
- What are the most discriminative features derived from particular data sources for addressing the CA problem?
- How does the performance of CA methods change when normal users behave in a malicious manner (i.e., considering normal data of user versus the data from masquerader period when he/she acted as a masquerader)?

Sentiment Analysis & Psychology. In our competition, we had asked the students to fill up a psychological questionnaire containing 50 questions. Our main intention behind this was to understand certain characteristics of each student that can help to correlate their behavior when they have an opportunity to become traitors. However, we got only a few instances of traitors, which is not enough for doing statistically significant experiments, but researchers are free to try it.

Nevertheless, the data from psychological questionnaires and emails can be used for investigation of research questions related to psychology or sentiment analysis, which may further help in a traitor-level risk assessment of individuals. The examples of such research questions are as follows:

- Is it possible to correlate the groups of people with similar sentiment characteristics (based on text in emails) and groups of people with similar personality traits?
- If there exist any correlations mentioned in the previous question, is it possible to utilize them in other research questions related to continuous authentication, authorship verification or identification? And if so, what is their impact?

8 Related Work

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

8.1 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. [56] 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. [57] 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. [58] 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 reward (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. [59] 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.

8.2 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 12 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

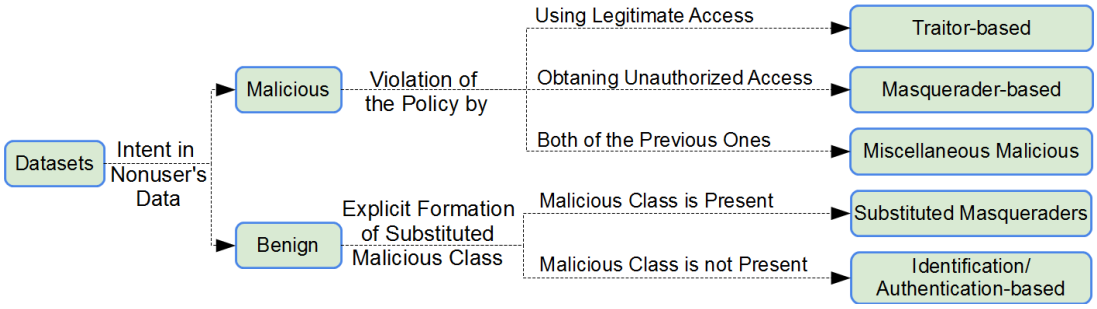


Figure 12: Categorization of datasets

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., [16]), the second best option is to use datasets aimed at simulation of real environments (e.g., [15, 17]), 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 [15] is a masquerader dataset that was introduced by Salem and Stolfo in [46, 5]. 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 [14] has been designed and implemented by Camiña et al. [6] 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 [7]) 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 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. [4] 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 [16] 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 [17] and described generation approach of the datasets in [60]. 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 [4].

Schonlau dataset (a.k.a. SEA) [61] was introduced by Schonlau et al. [10] 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 [62].

Balabit Corp. has created a dataset intended for performance evaluation of behavioral biometrics based on mouse dynamics [63]. 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 [11] 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* [64]. 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 [12] was introduced by Lane and Brodley [65] 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 [13] was designed for continuous knowledge acquisition and individualized tutoring of application software across an organization [66]. However it was also used for analysis of human interactions with GUI-based applications for the purpose of user authentication [67]. 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.

9 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. Furthemore, we illustrated the potential use of the TWOS dataset in multiple areas of cyber security, which does not limit to malicious insider threat only. Also, we presented several state-of-the-art features that can be extracted from different data sources in order to guide researchers in the analysis of the dataset.

In future work, we will do deeper analysis of the dataset and do comparative study of supervised classification algorithms applied to the insider threat detection problem. We also plan to collect the second version of TWOS dataset, where we will aim to remove drawbacks of the current version, and moreover we will extend the duration of the experiment with regards to incorporated concept drift.

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] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley, January 2012.

[2] Verizon, "2016 Data Breach Investigations Report," http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/, [Online; Accessed on July 1, 2017], 2016.

[3] Accenture, "Accenture," https://www.accenture.com/t20160704T014005Z__w__/us-en/_acnmedia/PDF-23/Accenture-State-Cybersecurity-and-Digital-Trust-2016-Report-June.pdf#zoom=50, [Online; Accessed on July 1, 2017], 2016.

[4] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A Survey of Insider Attack Detection Research," in *Insider Attack and Cyber Security*. Springer, 2008, pp. 69–90.

[5] M. B. Salem and S. J. Stolfo, "Modeling User Search Behavior for Masquerade Detection," in *Proc. of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID'11), Menlo Park, California, USA*, ser. Lecture Notes on Computer Science, vol. 6961. Springer, Berlin, Heidelberg, September 2011, pp. 181–200.

[6] B. Camiña, R. Monroy, L. A. Trejo, and E. Sánchez, "Towards Building a Masquerade Detection Method Based on User File System Navigation," in *Proc. of the 10th Mexican International Conference on Artificial Intelligence (MICAI'11), Puebla, Mexico*, ser. Lecture Notes on Computer Science, vol. 7094. Springer, Berlin, Heidelberg, November-December 2011, pp. 174–186.

[7] J. B. Camiña, R. Monroy, L. A. Trejo, and M. A. Medina-Pérez, "Temporal and Spatial Locality: An Abstraction for Masquerade Detection," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2036–2051, September 2016.

[8] A. Garg, R. Rahalkar, S. Upadhyaya, and K. Kwiat, "Profiling Users in GUI Based Systems for Masquerade Detection," in *Proc. of the 4th IEEE Workshop on Information Assurance (IWIA'06), Egham, Surrey, UK*. IEEE, June 2006, pp. 48–54.

[9] K. S. Killourhy and R. A. Maxion, "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics," in *Proc. of the 39th IEEE/IFIP International Conference on Dependable Systems & Networks (DSN'09), Lisbon, Portugal*. IEEE, July-June 2009, pp. 125–134.

[10] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, "Computer Intrusion: Detecting Masquerades," *Statistical science*, pp. 58–74, 2001.

[11] S. Greenberg, "Saul Greenberg's Homepage," http://saul.cpsc.ucalgary.ca/, [Online; Accessed on May 1, 2017], 1988.

[12] Purdue Millenium Lab, "UNIX User Data," 1998.

[13] MITRE Corporation, "OWL Dataset: Usage of Microsoft Word Commands," http://research.cs.rutgers.edu/~sofmac/ml4um/data.html, [Online; Accessed on May 1, 2017], 2000.

[14] B. Camiña, "WUIL Dataset," http://homepage.cem.itesm.mx/raulm/wuil-ds/, [Online; Accessed on May 1, 2017], 2014.

[15] Salem, Ben, "RUU dataset," http://sneakers.cs.columbia.edu/ids/RUU/data/, [Online; Accessed on May 1, 2017], 2009.

[16] CALO Project, "Enron Email Dataset," http://www.cs.cmu.edu/~enron/, [Online; Accessed on May 1, 2017], 2015.

[17] CERT, "Insider Threat Tools - Dataset," https://www.cert.org/insider-threat/tools/, [Online; Accessed on May 1, 2017], 2013.

[18] A. Harilal, F. Toffalini, J. Castellanos, J. Guarnizo, I. Homoliak, and M. Ochoa, "TWOS: A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition," in *Proc. of the 9th ACM CCS International Workshop on Managing Insider Security Threats (MIST'17), Dallas, USA*. ACM, October-November 2017, pp. 45–56.

[19] R. W. Shirey, "Internet Security Glossary, Version 2," https://tools.ietf.org/html/rfc4949, [Online; Accessed on March 1, 2018], 2007, IETF RFC 4949.

[20] Amazon Web Services Inc, "Amazon Web Servicies (AWS)," https://aws.amazon.com, [Online; Accessed on July 1, 2017], 2017.

[21] Amazon Web Services Inc, "Amazon Elastic Compute Cloud (EC2)," https://aws.amazon.com/ec2/, [Online; Accessed on July 1, 2017], 2017.

[22] Amazon Web Services Inc, "Amazon WorkSpaces," https://aws.amazon.com/workspaces/, [Online; Accessed on July 1, 2017], 2017.

[23] D. A. Solomon and H. Custer, *Inside Windows NT*. Microsoft Press Redmond, 1998, vol. 2.

[24] M. Palmér, "Pynput," https://pypi.python.org/pypi/pynput, [Online; Accessed on June 1, 2017], 2017.

[25] Windows, "Process Monitor," 2017.

[26] M. E. Russinovich and A. Margosis, *Troubleshooting with the Windows Sysinternals Tools*. Microsoft Press, 2016.

[27] Squid-Cache, "Squid Proxy," http://www.squid-cache.org/, [Online; Accessed on July 1, 2017], 2017.

[28] Tcpdump, "Tcpdump," http://www.tcpdump.org/, [Online; Accessed on July 1, 2017], 2017.

[29] Microsoft, "Microsotft Exchange," 2017.

[30] Microsoft, "Microsoft Outlook," https://www.microsoft.com/en-xm/outlook-com/, [Online; Accessed on July 1, 2017], 2017.

[31] Oracle, "MySQL," https://www.mysql.com/, [Online; Accessed on July 1, 2017], 2017.

[32] R. Ben'Ary, *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, 1989.

[33] Pennebaker Conglomerates, Inc., "LIWC 2015," 2017, accessed on July/2017. [Online]. Available: http://liwc.wpengine.com

[34] M. Maasberg, J. Warren, and N. L. Beebe, "The Dark Side of the Insider: Detecting the Insider Threat

Through Examination of Dark Triad Personality Traits," in *Proc. of the 48th Hawaii International Conference on System Sciences (HICSS'15), Washington DC, USA*. IEEE, January 2015, pp. 3518–3526.

[35] F. L. Greitzer and T. A. Ferryman, "Methods and Metrics for Evaluating Analytic Insider Threat Tools," in *Proc. of the 2013 Security and Privacy Workshops (SPW'13), San Francisco, California, USA*. IEEE, May 2013, pp. 90–97.

[36] E. D. Shaw, "The Role of Behavioral Research and Profiling in Malicious Cyber Insider Investigations," *Digital Investigation*, vol. 3, no. 1, pp. 20–31, 2006.

[37] E. Cole and S. Ring, *Insider Threat: Protecting the Enterprise from Sabotage, Spying, and Theft*. Syngress, 2005.

[38] M. L. Ambrose, M. A. Seabright, and M. Schminke, "Sabotage in the Workplace: The Role of Organizational Injustice," *Organizational Behavior and Human Decision Processes*, vol. 89, no. 1, pp. 947–965, 2002.

[39] D. L. Paulhus and K. M. Williams, "The Dark Triad of Personality: Narcissism, Machiavellianism, and Psychopathy," *Journal of Research in Personality*, vol. 36, no. 6, pp. 556–563, 2002.

[40] S. Mori, H. Nishida, and H. Yamada, *Optical Character Recognition*. John Wiley & Sons, Inc., 1999.

[41] S. P. Banerjee and D. Woodard, "Biometric Authentication and Identification Using Keystroke Dynamics: A Survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012.

[42] S. Mondal and P. Bours, "A Study on Continuous Authentication Using a Combination of Keystroke and Mouse Biometrics," *Neurocomputing*, vol. 230, pp. 1–22, 2017.

[43] H. Gamboa and A. Fred, "A Behavioural Biometric System Based on Human Computer Interaction," in *Proc. of the 1st SPIE on Biometric Technology for Human Identification*, ser. 5404, 2004, pp. 381–392.

[44] C. Shen, Z. Cai, X. Guan, H. Sha, and J. Du, "Feature Analysis of Mouse Dynamics in Identity Authentication and Monitoring," in *Proc. of the 19th IEEE International Conference on Communications (ICC'09), Dresden, Germany*. IEEE, June 2009, pp. 1–5.

[45] H. Eldardiry, E. Bart, J. Liu, J. Hanley, B. Price, and O. Brdiczka, "Multi-Domain Information Fusion for Insider Threat Detection," in *Proc. of the 2013 Security and Privacy Workshops (SPW'13), San Francisco, California, USA*. IEEE, May 2013, pp. 45–51.

[46] M. B. Salem and S. J. Stolfo, "Masquerade Attack Detection Using a Search-Behavior Modeling Approach," Columbia University, Computer Science Department, Tech. Rep. CUCS-027-09, 2009.

[47] C. Gates, N. Li, Z. Xu, S. N. Chari, I. Molloy, and Y. Park, "Detecting Insider Information Theft using Features from File Access Logs," in *Proc. of the 19th European Symposium on Research in Computer Security (ESORICS'14), Wrowclaw, Poland*, ser. Lecture Notes in Computer Science, vol. 8713. Springer, Cham, September 2014, pp. 383–400.

[48] I. U. University of California, "KDD Cup 99," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, [Online; Accessed on March 1, 2018], 1999.

[49] A. W. Moore, D. Zuev, and M. Crogan, "Discriminators for Use in Flow-based Classification," Intel Research, Cambridge, Tech. Rep., 2005.

[50] "Kyoto 2006+ Dataset," http://www.takakura.com/Kyoto_data/, [Online; Accessed on March 1, 2018], 2006.

[51] I. Homoliak, M. Barabas, P. Chmelar, M. Drozd, and P. Hanacek, "ASNM: Advanced Security Network Metrics for Attack Vector Description," in *Proc. of the 12th International Conference on Security & Management (SAM'13), Las Vegas, USA*. Computer Science Research, Education, and Applications Press, July 2013, pp. 350–358.

[52] I. Homoliak, "Intrusion detection in network traffic," Ph.D. dissertation, Brno University of Technology, Czech Republic, 2016.

[53] W. B. Jaballah and N. Kheir, "A Grey-Box Approach for Detecting Malicious User Interactions in Web Applications," in *Proc. of the 8th ACM CCS International Workshop on Managing Insider Security Threats (MIST'16), Vienna, Austria*. ACM, October 2016, pp. 1–12.

[54] I. A. Gheyas and A. E. Abdallah, "Detection and Prediction of Insider Threats to Cyber Security: A Systematic Literature Review and Meta-Analysis," *Big Data Analytics*, vol. 1, no. 1, August 2016.

[55] G. Gavai, K. Sricharan, D. Gunning, R. Rolleston, J. Hanley, and M. Singhal, "Detecting Insider Threat from Enterprise Social and Online Activity Data," in *Proc. of the 7th ACM CCS international workshop on*

managing insider security threats (MIST'15), Denver, Colorado, USA. ACM, October 2015, pp. 13–20.

[56] O. Brdiczka, J. Liu, B. Price, J. Shen, A. Patil, R. Chow, E. Bart, and N. Ducheneaut, "Proactive Insider Threat Detection Through Graph Learning and Psychological Context," in *Proc. of the 2012 Security and Privacy Workshops (SPW'12), San Francisco, California, USA*. IEEE, May 2012, pp. 142–149.

[57] P. J. Taylor, C. J. Dando, T. C. Ormerod, L. J. Ball, M. C. Jenkins, A. Sandham, and T. Menacere, "Detecting Insider Threats Through Language Change." *Law and Human Behavior*, vol. 37, no. 4, p. 267, August 2013.

[58] S. M. Ho, J. T. Hancock, C. Booth, M. Burmester, X. Liu, and S. S. Timmarajus, "Demystifying Insider Threat: Language-action Cues in Group Dynamics," in *Proc. of the 49th Hawaii International Conference on System Sciences (HICSS'16), Washington DC, USA*. IEEE, January 2016, pp. 2729–2738.

[59] A. Azaria, A. Richardson, S. Kraus, and V. Subrahmanian, "Behavioral Analysis of Insider Threat: A Survey and Bootstrapped Prediction in Imbalanced Data," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 2, pp. 135–155, June 2014.

[60] J. Glasser and B. Lindauer, "Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data," in *Proc. of the 2013 Security and Privacy Workshops (SPW'13), San Francisco, California, USA*. IEEE, May 2013, pp. 98–104.

[61] M. Schonlau, "Masquerading User Data," http://www.schonlau.net/, [Online; Accessed on May 1, 2017], 2001.

[62] R. A. Maxion and T. N. Townsend, "Masquerade Detection using Truncated Command Lines," in *Proc. of the 32nd International Conference on Dependable Systems and Networks (DSN'02), Washington, D.C., USA*. IEEE, June 2002, pp. 219–228.

[63] A. Fülöp, L. Kovács, T. Kurics, and E. Windhager-Pokol, "Balabit Mouse Dynamics Challenge Dataset," https://github.com/balabit/Mouse-Dynamics-Challenge, [Online; Accessed on May 1, 2017], 2016.

[64] S. Greenberg, "Using Unix: Collected Traces of 168 Users," University of Calgary, Tech. Rep., 1988.

[65] T. Lane and C. E. Brodley, "An Application of Machine Learning to Anomaly Detection," in *Proc. of the 20th National Information Systems Security Conference (NIST'97), Baltimore, Maryland, USA*, vol. 377. NIST, October 1997, pp. 366–380.

[66] F. Linton, D. Joy, H.-P. Schaefer, and A. Charron, "OWL: A Recommender System for Organization-Wide Learning," *Educational Technology & Society*, vol. 3, no. 1, pp. 62–76, March 2000.

[67] A. El Masri, H. Wechsler, P. Likarish, and B. B. Kang, "Identifying Users with Application-Specific Command Streams," in *Proc. of the 12th Annual International Conference on Privacy, Security and Trust (PST'14), Toronto, Ontario, Canada*. IEEE, July 2014, pp. 232–238.

Author Biography

Athul Harilal is a Research Assistant at Singapore University of Technology and Design (SUTD). His research interests focuse on various aspects of malicious insiders such as detection of data exfiltration by insiders, forensic analysis of insider file system and process activities etc. Prior to working at SUTD, he worked on malware analysis that tried to identify the source of malware spread across the network as part of his Master's thesis at Nanyang Technological University (NTU), where he pursued Communications Engineering.

Flavio Toffalini is a Ph.D. student at Singapore University of Technology and Design (SUTD), Singapore. His research is mainly focused on designing of novel techniques for studying and monitoring insider threats by using trusted computing technologies. Flavio obtained his Master Degree in Computer Science at University of Verona with a thesis about Web-security. He also collaborated with other European research centers such as University of Padua (Italy), and Eurecom (France) on cyber-security and software engineering related topics.

Ivan Homoliak is a Postdoctoral Research Fellow at Singapore University of Technology and Design (SUTD) and currently works in insider threat detection project that focuses on application of machine learning for insider threat detection. Ivan has a Ph.D. in the area of adversarial intrusion detection in network traffic from Brno University of Technology, Faculty of Information Technology (BUT FIT), Czech Republic (2016). Ivan earned Master of Science degree from the BUT FIT in 2012 in the areas related to intrusion detection and supervised machine learning.

John Henry Castellanos is a PhD student at SUTD in 2015 John Henry received his MSc degree in Information Security at Universidad de Los Andes (Bogota, Colombia). Before he studied Electronics Engineering at Universidad Francisco de Paula Santander (Cúcuta, Colombia). He has over nine years of experience in several industries such as Automation, Telecommunications, Datacenter and Cybersecurity. Since 2016 he has been extending his work in research in different topics related to applied security.

Juan David Guarnizo Hernandez completed a Master degree in Information Security (MSc) at Universidad de los Andes (Bogotá, Colombia). He worked more than four years in banking under roles such as software developer and IT security analyst. Now, Juan is a Ph.D. student at Singapore University of Technology and Design (SUTD). As research interests, he pursues to elaborate new methods and techniques of malicious insider detection and prevention in business environments.

Soumik Mondal is a post-doctoral research fellow at Singapore University of Technology and Design (SUTD), and his research aims at the pattern recognition and machine learning challenges related to information security. Before joining SUTD, Soumik earned his Ph.D. in Information Security from Norwegian University of Science and Technology (NTNU) and worked as a post-doctoral research fellow at University of Twente (UT), Netherlands. He is the author of more than 20 publications in this area. His research interests are in cyber security, cyber forensics, and biometrics.

Martín Ochoa is assistant professor at the department of Applied Mathematics and Computer Science of the Universidad del Rosario, Bogotá, Colombia. He is interested in foundational and applied aspects of Software Security. He has a background in Mathematics and Systems Engineering and holds a PhD in Computer Science from the TU Dortmund (Germany). Prior to his current affiliation he has been assistant professor in the Singapore University of Technology and Design, post-doc in the Technical University of Munich and researcher in cybersecurity at Siemens CT.

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