Insider Threat Detection - A Machine Learning Perspective

Introduction

Computer networks and telecommunications play a significant role in information exchange. An increase in valuable information, along with enabling technology expansions, have led to increases in threats. The sources of these threats are not only from outside but, also, from within the organization. Such threats possess a large security risk and are seemingly difficult to detect. According to Wikipedia, insider threat is a malicious threat to an institution that arrives from people within the enterprise itself such as employees, former employees, contractors or business associates, who have insider information concerning the organization's security practices, data and computer systems. Insiders are highly dangerous in almost all industries/domains ranging from manufacturing, healthcare, and national security because unlike outsiders, working for penetration in a company, they typically have legitimate access to computer systems and the network which they need to perform their daily jobs. (Awake Security). According to CISA, well publicized insiders have caused chronic and irredeemable harm to national security.

Trusted insiders commit intentional or unintentional disruptive or harmful acts across all infrastructure sectors and in virtually every organizational setting. Once an insider threat is caught and revealed to the public, trust in an enterprise is diminished. The public that relies on an agency or business becomes wary of who has access to their information. Additionally, negative media reports further tarnish a company's reputation.

Once an internal threat crisis becomes full-blown, it's no longer just an inside problem - it becomes an external one eventually. Making matters worse, when dealing with an external threat, high-level leaders usually lose focus on further internal threat assessment measures until the external crisis is dealt with. The longer an insider threat incident lingers, the costlier it gets for businesses. Incidents that took more than 90 days to contain cost organizations $13.71 million on an annualized basis, while incidents that lasted less than 30 days cost roughly half, at $7.12 million (observeIT).

According to TechJury, more than 34% businesses around the globe are affected by insider threats yearly. The number of insider-caused cybersecurity incidents increased by a whopping 47% since 2018. According to observeIT, the average annual cost of insider threats has also skyrocketed in 2020, rising 31% to $11.45 million.

Traditional cybersecurity tools fail at helping organizations to properly implement the least privilege principle, which means that no worker is allowed to gain access to more data than required to perform their tasks at any point. Many companies still hold on to signature-based (or rule-based) cybersecurity tools in an age where cyber-attacks have transformed. Cyber-criminals now use artificial intelligence and machine learning to scale their attacks and work with greater preciseness, deftness, and sophistication. Unfortunately, legacy security tools are weak in the face of such advanced threats (Infosecurity). Clues towards an employee's malicious intentions may be spread across multiple datasets, hidden among tens or hundreds of thousands of other data points, or separated by weeks or months of inactivity. Machines, on the other hand, excel at these types of subtle pattern detection across large datasets. Specialized algorithms can be designed to look for anomalies, such as deviations from normal computing behavior or violations of defined policies and procedures, or even just activities that don't match

the behavior of other employees. When enough of these indicators co-occur in a single employee, an organization has significant reason for concern.

This project has been designed for providing a machine learning oriented solution for detecting insider threats. This project utilizes the Isolation Forest Model, an unsupervised anomaly detection algorithm which helps in identifying insiders responsible for performing activities of such nature. Our proposed approach is based on using this machine learning algorithm over systems logs. Log analysis is the most effective insider threat detection method when a security breach is discovered. Organizations that do not use activity monitoring and analysis of systems' logs have fewer chances of uncovering insider attacks.

Problem Statement

As organizations depend on cyber systems to support critical missions, a malicious insider who is trying to harm an organization can do so through, for instance, wrecking a critical IT system by various potential tools and techniques or by stealing intellectual property (IP) to benefit a new employer or a competitor. Government and industry organizations are responding to this change in the threat landscape and are increasingly aware of the escalating risks. CERT has been a widely acknowledged leader in insider threat since it began investigating this problem in 2001. The *CERT Guide to Insider Threat* was inducted in 2016 into the Palo Alto Networks Cybersecurity Canon, illustrating its value in helping organizations understand the risks that their own employees pose to critical assets. Since 2001, the CERT Insider Threat Center has collected and analyzed information about hundreds of insider cybercrimes, ranging from national security espionage to theft of trade secrets.

The CERT Division, in partnership with ExactData, LLC, and under sponsorship from DARPA I2O, generated a collection of synthetic insider threat test datasets. These datasets provide both synthetic background data and data from synthetic malicious actors.

For more background on this data, please see the paper, Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data.

Version 3.2 of the data was chosen for this project. The following datasets were extracted and utilized out of this release:

a. Logon.csv
    i. Fields: id, date, user, pc, activity (Logon/Logoff).
    ii. Weekends and holidays are included as days when fewer people work.
    iii. No user may log onto a machine where another user is already logged on, unless the first user has locked the screen.
    iv. Logoff requires preceding logon.
    v. A small number of daily logons are intentionally not recorded to simulate dirty data.
    vi. Some logins occur after-hours

vii. After-hours logins and after-hours thumb drive usage are intended to be significant.
viii. Screen unlocks are recorded as logons.
ix. Some users log into another user's dedicated machine from time to time.

b. Device.csv
   i. Fields: id, date, user, pc, activity (connect/disconnect).
   ii. Some users use a thumb drive.
   iii. Some “connect” events may be missing disconnect events, because users can power-down the machine before removing the drive.
   iv. Users are assigned a normal/average number of thumb drive uses per day. Deviations from a user's normal usage can be considered significant.

c. File.csv
   i. Fields: id, date, user, pc, filename, content
   ii. Each entry represents a file copy/transfer to a removable media device.
   iii. Content consists of a hexadecimal encoded file header followed by a space-separated list of content keywords.

d. Users.csv
   i. Fields: employee name, userID, email, functional unit, department, team, supervisor name
   ii. Extracted employee data for the following Departments for functional unit - 2 i.e. Research and Engineering:
      1. Engineering
      2. Software Management

e. Psychometric.csv
   i. Fields: employee_name, user_id, O, C, E, A, N
   ii. Big 5 psychometric score
   iii. See Big 5 Personality Traits for the definitions of O, C, E, A, N.

Full Metadata information can be found from the Insider Threat Test Dataset.

The purpose of this research project is to identify potential insiders, accessing confidential information and not following the least privilege principle via various activities, using anomaly detection algorithms.

Methodology

In this work, we apply our proposed approach on a well known synthetic dataset for insider threat studies (r3.2) generated by CERT. The dataset is over 20GB of various system log files recording all activities of 1,000 users in a duration of 500 days including weekends. The activity log files are related to logons, devices, files, users, and users' psychometric scores. The date and time of each activity are also recorded in the dataset. Each log file includes various information about activities. For example, the logon file contains the date, time, user ID, PC number and the type of activity (e.g., login or logoff). There are some missing activities such as missed logoff after login or devices disconnecting after a successful connection. This happens when a user powers off a machine before logging off or discards a removable device.

To be able to apply our proposed model over large datasets such as log files; first, we need to process the raw data and extract the essential attributes to build the features since it is important to identify anomalies in behavioral patterns recorded in log files and the users performing them.

In feature engineering, not all data are usable or beneficial for detecting insider threats. Hence, we extract the most descriptive attributes from each log file and combine them in one table.

Log_on_off_stats:

In this log file, there are two types of activities: logons and logoffs. There are users who logged in after working hours or on weekends indicating anomalous behavior. We extract each user's activities based on the time of the activity. Summary statistics with respect to login and log off activity of each user has been recorded in this dataset. The extracted features are user, on_min_ts, on_max_ts, on_mode_ts, on_mean_ts, off_min_ts, off_max_ts, off_mode_ts, off_mean_ts. Here 'ts' means time in seconds, 'on' means log on and 'off' means log off.

Device_disconn_stats and Device_conn_stats:

All activities in these files indicate usage of removable drives on PCs including the connection or the disconnection of a device. Similar to the logon file, we process the data related to devices based on time resulting in the following features: user, min_ts, max_ts, mode_ts, mean_ts.

File_stats_new:

Each row in the file log represents each users' mode and maximum number of files transferred onto a removable device. Although file names and content are included, we do not consider them in our feature set. The final feature set includes the following features: user, mode_transfers_per_user, and max_transfers_per_user.

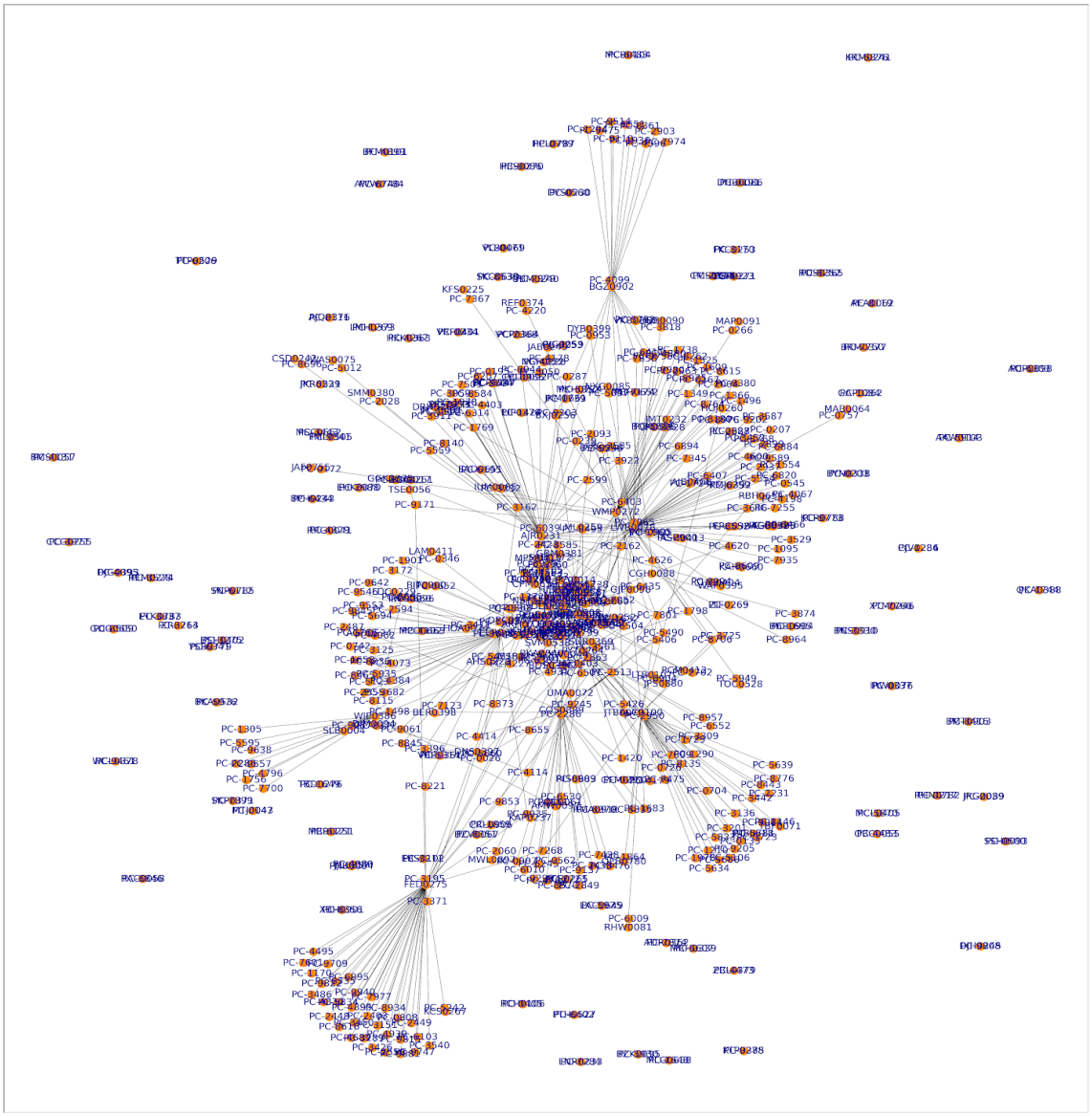
Note: File_stats_new, Device_disconn_stats and Device_conn_stats have been combined to form device_full for fitting Isolated Forests model.

Psychometric_users_tidy:

Psychometric attributes describe the users' five-factor model of personality dimensions and is used for employee evaluation reports. A higher employee psychometric score requires higher scores in *Extraversion*, *Conscientiousness*, *Agreeableness*, *Openness* and a lower score in *Neuroticism*. We use these attributes to calculate the psychometric score for each user and include it as a feature in our model. The features in this dataset are: employee_name, user_id, O, C, E, A, N.

User_pc relationship:

Graph analysis provides a detailed visualization for identifying each user-pc relationship. There are instances where multiple machines are used by a single user. An undirected bipartite network presents a holistic view of this arrangement where the edge weights represent the number of log off instances between a user and a pc. Log off instances have been used since logoffs require preceding logins. The visualization can be seen on the next page:



Model Results

Anomaly Score for malicious users identified by Isolated Forest Algorithm under User - PC relationship can be visualized below:

	user	ascore
11	AJR0231	-0.227213
12	ALC0100	-0.153243
16	ARH0777	-0.144462
24	BGZ0902	-0.242321
44	CGH0088	-0.144462
...	...	...
69	FED0275	-0.252276
132	LWB0078	-0.342761
157	PYT0264	-0.206112
204	WJP0386	-0.245450
205	WMP0272	-0.176549

13 rows × 2 columns

This table depicts malicious users, identified by the algorithm, where there is an extremely large amount of user- pc interactions (Appendix A). By large, we mean any amount of interaction greater than 40. This is a user defined parameter and can be tweaked accordingly. Under the field ‘ascore’, if the anomaly score is less than 0, they are considered outliers and thus get identified by the algorithm.

On a similar note, we have the following tables corresponding to logon / logoff activity, removable device/file transfers and psychometric data

Table 1

	user	ascore
0	AJQ0376	-0.027424
3	BCP0247	-0.010455
5	BMS0057	-0.084555
7	CAE0080	-0.053961
11	CSD0242	-0.096347
...	...	...
34	QLC0248	-0.107472
36	REM0274	-0.077860
37	SBM0063	-0.094124
42	WXW0044	-0.014448
43	ZBL0379	-0.115604

21 rows × 2 columns

Table 2

	user	ascore
0	AJQ0376	-0.027424
3	BCP0247	-0.010455
5	BMS0057	-0.084555
7	CAE0080	-0.053961
11	CSD0242	-0.096347
...	...	...
34	QLC0248	-0.107472
36	REM0274	-0.077860
37	SBM0063	-0.094124
42	WXW0044	-0.014448
43	ZBL0379	-0.115604

21 rows × 2 columns

Table 3

	user	ascore
8	LML0259	-0.030337
10	JAB0249	-0.050925
16	ACL0394	-0.018298
17	JTB0079	-0.026172
23	YSB0779	-0.008631
...	...	...
197	ETK0783	-0.052447
200	BZK0095	-0.070558
202	GFM0250	-0.038079
212	QKA0388	-0.042877
213	GGK0375	-0.057603

58 rows × 2 columns

<u>Device/File Transfer</u> <u>Logon/Logoff</u> <u>Psychometric</u>

Table1: Device/File transfer

This table identifies malicious users based on maximum and mode file transfers per user (Appendix B) combined with time frame for which device was connected and disconnected respectively.

Table 2: Logon/Logoff

This table identifies malicious users based on maximum, minimum, mode and average time frame for which a user was logged on and logged off on a machine.

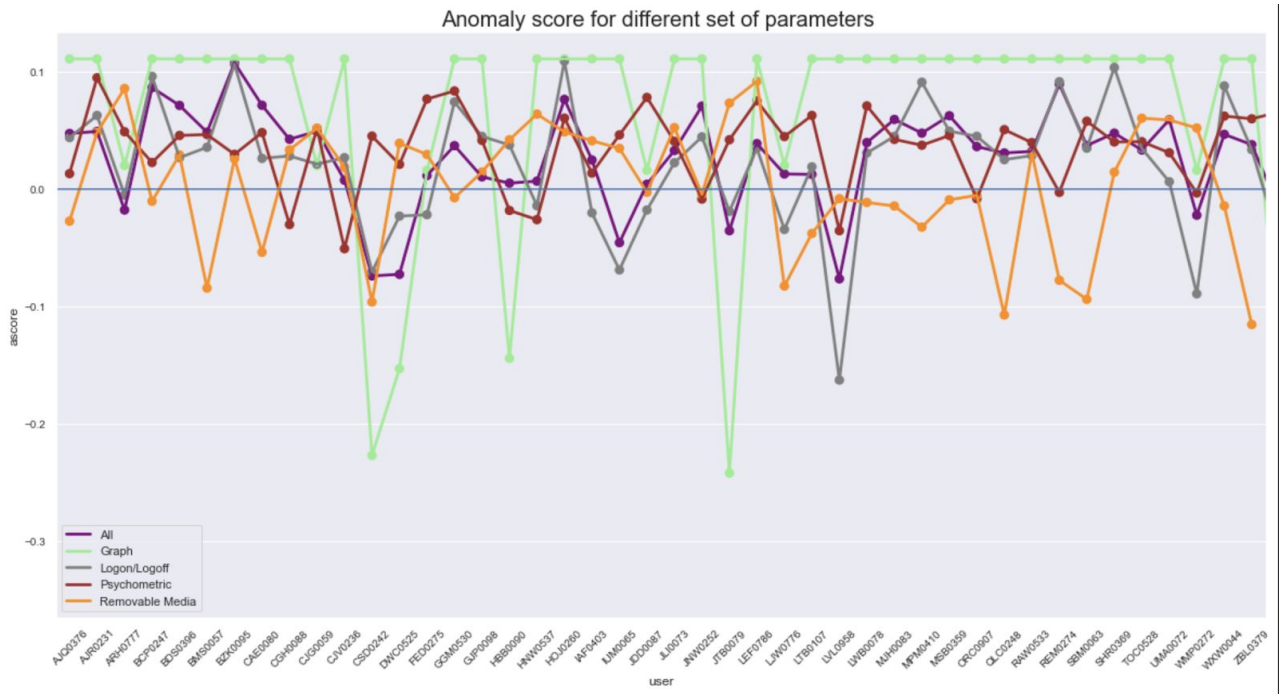
Table 3: Psychometric
This table identifies malicious users based on OCEAN psychology tests. Initial scores (out of 50) have been inputted in the model for identifying potential troublemakers.

The next table combines the 3 activities as model input to provide list of employees as potential threats:

	user	ascore
2	ACL0394	-0.017909
11	AJR0231	-0.074172
12	ALC0100	-0.072796
20	BCP0247	-0.045768
24	BGZ0902	-0.035601
...	...	...
204	WJP0386	-0.055455
205	WMP0272	-0.051767
206	WXW0044	-0.001165
210	YJT0368	-0.009480
212	ZBL0379	-0.025502

44 rows × 2 columns

Below is a line plot of anomaly scores for different set of parameters (separated and combined):



Benefits

Unsupervised learning techniques are attractive for researchers to explore complex problems such as insider threat detection. Due to the unpredictable nature of an insider's behavior, unsupervised learning techniques can help empower system administrators and security specialists to focus their efforts on suspected users.

Anomaly detection algorithms are based on two assumptions:

1. Most behavioral patterns of users in the system are normal with a small percentage of abnormal patterns; the second is that abnormal behavioral patterns are statistically different from normal ones.
2. Besides the fact that there is no need for labeling data for training a model, the main advantage of using unsupervised learning is its ability to adapt to the occurrence of new behavioral patterns of insiders. Among various unsupervised anomaly detection algorithms, the Isolation Forest algorithm is a promising one not only for their successful applications, but also for the mechanisms this algorithm is built on.

The Isolation Forest anomaly detection algorithm provides a ranking list that reflects the degree of the anomaly. This can be accomplished by setting the data points according to their anomaly score or path lengths. Thus, the points on top of the list will be the most anomalous and thus can identify the malicious user of the highest level of threat.

Costs

Implementing this algorithm has its downsides as well. The large amount of data produced by a system requires time and resources to produce a proper set of features/variables as training data. In other words, algorithms of such nature require a considerable amount of initial time and resources in performing feature engineering. This model is incapable of working with multivariate time series from a practical standpoint. It is also not possible to get a particular decision for each isolation tree (iTree). Moreover, it is not possible to visualize the tree under this algorithm. Another concern with respect to this algorithm is its sensitivity to a slight deviation in the data and its related effects. Lastly, this data did not provide any target feature or testing data for evaluating model performance using evaluation metrics. This led us to provide anomaly scores for various activities based on initial implementation of the model only.

Conclusion and Future Research

In this paper, we have studied the challenge of insider threat detection using Isolated Forest Algorithm. We have extracted meaningful features from a synthetic dataset including the psychometric scores of users. The dataset was vectorized in different ways including daily records as well as periodically aggregated records of user activities. The unsupervised algorithms will be of great help to system administrators or security auditors in an organization to focus their efforts on users who are detected as having anomalous behavior by the algorithms. Future work includes exploring other available datasets or considering the risk factor in the detection process. The risk factor will help in addressing the activities that pose more security risks to the organization’s resources. In addition, the user anomaly score produced in each cycle can be used as a trust score for that user in the next cycle. In future work, we can explore the idea of calculating the trust score based on other factors if provided, such as access control policy of the system.

References

CERT. (n.d.). Retrieved from https://wikis.ece.iastate.edu/insider-threat-detection/index.php/CERT

Big Five personality traits. (2020, December 23). Retrieved from http://en.wikipedia.org/wiki/Big_Five_personality_traits

CERT Guide to Insider Threats Named to Cybersecurity Canon. (n.d.). Retrieved from http://www.sei.cmu.edu/news/article.cfm?assetid=453778&article=097&year=2016

Cost of Insider Threats. (2020, November 17). Retrieved from https://www.observeit.com/cost-of-insider-threats/

Friday talks: The dark horse of Isolation Forest. (2020, August 06). Retrieved from https://thingsolver.com/friday-talks-the-dark-horse-of-isolation-forest/

Insider Threat Definition & Examples. (2019, August 19). Retrieved from https://awakesecurity.com/glossary/insider-threat/

Insider Threat Mitigation. (n.d.). Retrieved from https://www.cisa.gov/insider-threat-mitigation

Insider threat. (2020, July 06). Retrieved from https://en.wikipedia.org/wiki/Insider_threat

Isolation forest. (2020, November 26). Retrieved from https://en.wikipedia.org/wiki/Isolation_forest

Lewinson, E. (2019, September 26). Outlier Detection with Isolation Forest. Retrieved from https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e

Michael Usiagwu Digital Marketing ConsultantFollow @michaelusiagwu1Connect on LinkedIn. (2020, December 04). Insider Threat Mitigation: The Role of AI and ML. Retrieved from https://www.infosecurity-magazine.com/next-gen-infosec/insider-threat-mitigation-ai-ml/

Sklearn.ensemble.IsolationForest¶. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

(n.d.). Retrieved from https://ieeexplore.ieee.org/document/6565236

20 Insider Threat Statistics to Look Out For in 2020. (2020, August 17). Retrieved from https://techjury.net/blog/insider-threat-statistics/#gref

Appendix

Appendix A:

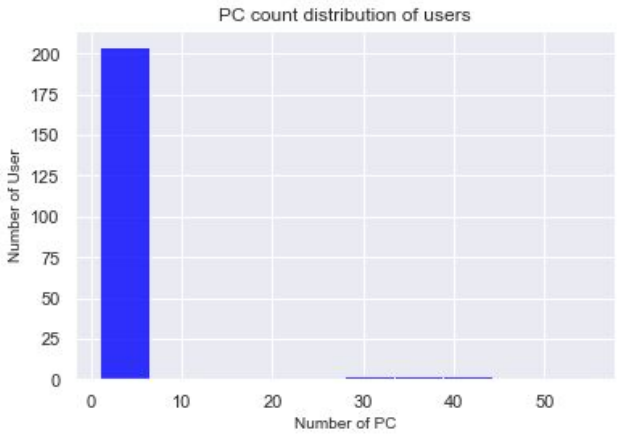
User-pc node degree relationship graph (SNA) - Bar chart Anomaly Visualization

In [ ]:
```
user_pc
# node degree from the graph analysis
plt.hist(user_pc.pc_count, alpha=0.8, color = 'blue');
plt.xlabel('Number of PC', size =10)
plt.ylabel('Number of User', size = 10)
plt.title('PC count distribution of users', size=12)
```

In [316]:
```
user_pc.loc[user_pc['pc_count'] > 40]
```

Out[316]:

	user	pc_count
69	FED0275	43
132	LWB0078	55



Appendix B:

Visualization - File transfers per user

In [102]:
```
file_stats_new
import seaborn as sns
```

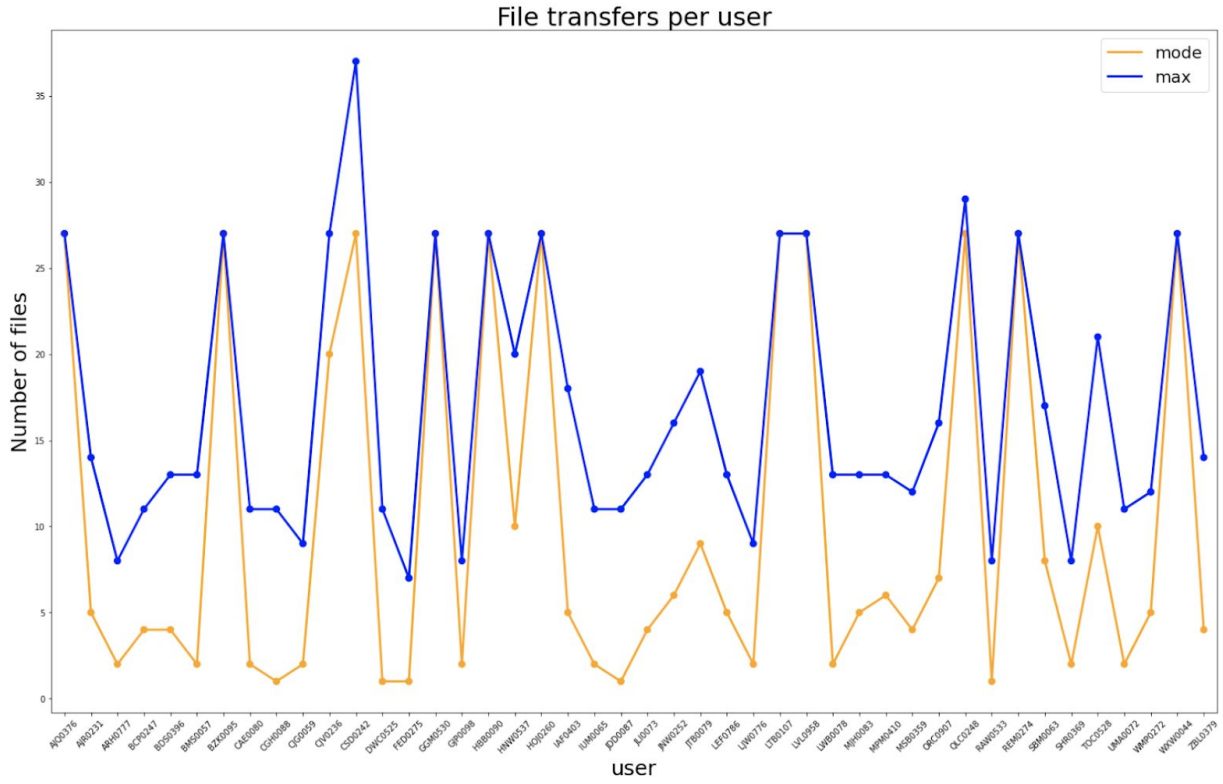
In [108]:
```
f, ax = plt.subplots(figsize = (25,15))
x_col='user'

sns.pointplot(ax=ax,x=x_col,y='mode_transfers_per_user',data=file_stats_new, color='orange')
sns.pointplot(ax=ax,x=x_col,y='max_transfers_per_user',data=file_stats_new,color='blue')

ax.legend(handles=ax.lines[::len(files_per_day_stats)+1], labels=["mode", "max"], fontsize = 20)

ax.set_title('File transfers per user', size = 30)
plt.rcParams["axes.labelsize"] = 25
plt.ylabel("Number of files")
plt.xticks(rotation = 45, fontsize = 10)
plt.yticks(fontsize = 10)
# plt.legend(fontsize=20)
plt.show()

```


File transfers per user
mode
max
Number of files
user
0
5
10
15
20
25
30
35
AYO376
APR0231
ARH0777
BCP0247
BDS0396
BMS0057
BZK0095
CAE0080
CGH0088
CQG0059
CVV0236
CSD0342
DWC0525
FED0275
GGM0530
GPO098
HBB0090
HNW0537
HOJ0260
IAF0403
IUM0085
JDD0087
JLL0073
JNW0252
JTB0079
LEF0786
LIW0776
LTB0107
LVL0958
LWB0078
MPH0083
MPM0410
MSB0359
ORC0907
QLC0248
RAW0533
REM0274
SBM0063
SHR0369
TOX0528
UMA0072
WMP0272
WXW0044
ZBL0379