Analysis and Evaluation of Keystroke Dynamics as a Feature of Contextual Authentication

Kemal Bicakci
Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey
bicakci@etu.edu.tr

Oguzhan Salman
Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey
osalman@etu.edu.tr

Yusuf Uzunay
Securify Information Tech. and Security Training Consulting Ltd.
Ankara, Turkey
yusuf.uzunay@securify.com.tr

Mehmet Tan
Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey
mtan@etu.edu.tr

***Abstract*—The current best practice dictates that even when the correct username and password are entered, the system should look for login anomalies that might indicate malicious attempts. Most anomaly detection approaches examine static properties of user's contextual data such as IP address, screen size and browser type. Keystroke Dynamics bring additional security measure and enable us to use individuals' keystroke behaviour to decide legitimacy of the user. In this paper, we first analyze different anomaly detection approaches separately and then show accuracy improvements when we combine these solutions with various methods. Our results show that including keystroke dynamics scores in session context anomaly component as a new feature performs better than ensemble methods with different weights for session context and keystroke dynamics components. We argue that this is due to the opportunity to capture the behavioral deviations of the individuals in our augmented model.**

***Index Terms*—User Authentication, Keystroke Dynamics, Contextual Authentication, Behavioural Biometrics, Machine Learning, Anomaly Detection.**

I. Introduction

User authentication used to be considered as a binary problem: if the username and password are correct, authentication is successful, otherwise it fails. We argue that this short-sighted viewpoint is the main reason of security problems including password cracking, phishing and many other attacks we all know about [20]. Today, more web sites consider user authentication as a more elaborate classification problem [10]. In this setting, correct entry of username and password is required but usually not sufficient. You collect as much relevant data as you can that might help you to differentiate between genuine logins and fraudulent logins with stolen credentials, then feed them into a classification algorithm. The algorithm provides not a single binary value but an authentication score between 0 and 1. A value closer to 1 gives more confidence that user is indeed the person who claims to be.

The authentication score could be used in various ways. For instance, a score smaller than a specified threshold may trigger additional authentication methods to boost the confidence. In other settings, users authenticated with a less than ideal score are only authorized with a limited subset of access rights. There are other use cases where we may want to tag the login attempts looking suspicious as anomalies for further investigation.

Many other policy options could be envisioned. No matter what kind of a policy is enforced based on the computed authentication score, we need to ensure that the accuracy of the classification is convincingly high. More specifically, as in a generic setting of a classification problem, two types of errors are possible:

- A type-I error also known as false positive means that a legitimate login attempt is considered as invalid.

- A type-II error also known as false negative means that an unauthorized login attempt is accepted as valid.

To reduce these errors, the rule of thumb is to diversify the data classification algorithms are trained on and to collect as much authentication data as possible. Two sources of data are especially prominent:

- Contextual information regarding the machine and the session: Operating system, screen-size, browser type of the machine used in the login as well as location (city and ISP) and time of the login.

- Keystroke dynamics: Timestamps of the key-up and key-down events while the user types in username and password.

Both of these data could be collected passively while users login as usual i.e., the user experience is not changed.

While the intuition that a classifier which uses the combination of these data performs better is strong, the actual state of the evidence is still mixed, at best. Filling this research gap and advancing theoretical and practical understanding in

This research is funded by TUBITAK (The Scientific and Technological Research Council of Turkey) under the grant No: 118E399.

978-1-6654-1863-8/20/$31.00 ©2020 IEEE

such a crucial area is the main motivation of our work. More specifically, we contribute to the literature in several ways:

- We collect all of the user data and simulate the attacks on the same web application we have developed. Thus, performance results could be shown on a coherent and realistic dataset (previous work has evaluated the performance using data collected from various sources pertaining to different users and combined them rather artificially [14]).

- We systematically review options for combining keystroke dynamics and contextual anomaly components. We evaluate and compare the performance results of these options.

- We show that the best accuracy result is obtained when keystroke dynamics score is included in session context anomaly component as a new feature. We argue this is due to the ability to take into account the deviations of these scores in this model.

The rest of the paper is organized as follows: The data collection procedure is presented in Section 2. All machine learning models for anomaly detection we build, analyze and compare are explained in Section 3. We discuss the performance results and promising future directions in Section 4. Limitations of our work are given in Section 5. The related work is summarized in Section 6. Concluding remarks are presented in Section 7.

II. Data Collection

We collected the data through a course website which require students to register with a username and password. The course website is used throughout one semester to reach course related information and announcements. For privacy reasons, we stored only the hash values of usernames. We provided users system-assigned passwords which start with the block "operatingsystem" followed by four numeric digits. This choice was made for security reasons. It prevents students from choosing their passwords which are likely to be used for their other websites [6].

In each login, we collected two types of data: machine and session properties and keystroke dynamics of individuals. Keystroke dynamics data are collected by JavaScript that we implemented, which was bond to only login forms.

As a total, we collected 4748 login data from 102 different users.

A. Data Properties

As mentioned, in each login we collected data pertaining to Keystroke Dynamics, Machine Properties and Session Properties. The summary description of data fields in each category is shown in Table 1.

Data Field	Description
Keystroke Dynamics	*Timestamps of Key-up and Key-down Events*
Machine Properties	*Operating System, Screen-size, Browser Type*
Session Properties	*City, ISP, Time of Login*

TABLE I
Summary of Collected Data.

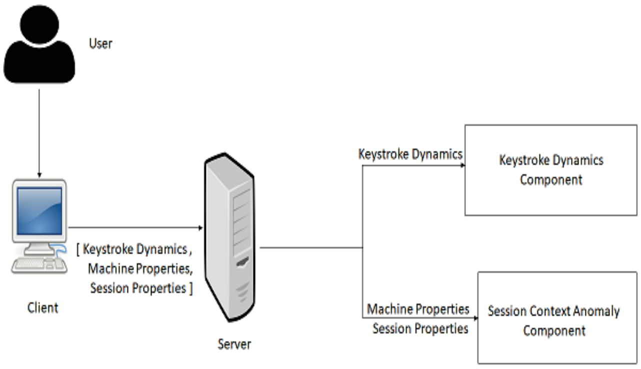


Fig. 1. Data flow during user logins.

B. Attack Simulation

We randomly selected 29 users from the pool of students who logged in to our system not less than 20 times. One of the authors then logged in at least 10 times as the selected students from different locations with different ISPs in order to simulate unauthorized access. For half of our attack simulations, we logged in from the same city but different ISP and for the other half, we used VPN to simulate access from abroad. We used randomly-selected operating-system, browser type, screen-size and login time for all of attack simulations.

III. Authentication Anomaly Detection

We test and compare an extensive list of machine learning algorithms and observe that tree-based methods outperform others in our setting. For the sake of brevity, we only report the results of these best-performing algorithms.

In total, we build four anomaly detection machine learning models. Two of these correspond to separate keystroke dynamics and session context models. Then, we build ensemble models from these machine learning components using weighted averaging based on voting convention. In our fourth and last model, we feed keystroke scores as a new additional feature to the session context model.

The best performing algorithms are isolation forest for keystroke dynamics and random forest for session context anomaly components, respectively. In total, we use 2870 login data for 29 students. We use the same training (70%) and test (30%) data in all four models. Below, we provide further details of our anomaly detection models.

A. Keystroke Dynamics

As mentioned, Keystroke Dynamics data is collected only during login from two form fields, namely username and password. In each field, the timing of keyboard key-up and key-down events are captured and from these timings, we construct two features for every pressed key by the following formula:

$$F_1 = T_{KeyUp}(n) - T_{KeyDown}(n)$$

$$F_2 = T_{KeyUp}(n) - T_{KeyUp}(n+1)$$

where F represents feature, T represents timestamp for the specified keyboard up or down events and n represents the pressed key order. By using the feature set for twenty logins, we use Isolation Forest algorithm to learn (train for) each user's keyboard dynamics pattern.

After the training, the system compares the current features of key-up and key-down events with the previously computed keystroke dynamics model and generate a score between 0 and 1. When we analyze the keystroke scores for all normal user logins and attack simulation logins, we see that our model successfully differentiates real users from attack simulations with EER (Equal Error Rate) [1] of 18% with a threshold of 0.86. The keystroke scores for both cases are shown in Figure 2 and ROC curve is presented in Figure 3 .

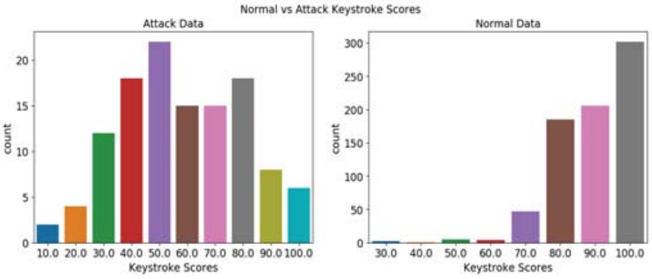


Fig. 2. Keystroke Dynamics Scores.

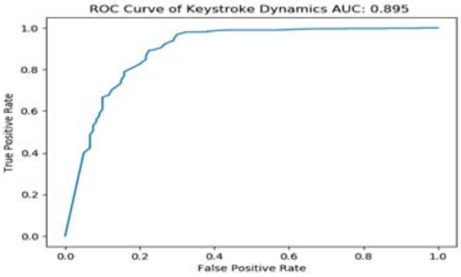


Fig. 3. ROC Curve for Keystroke Dynamics.

B. Session Context

Before feeding the data to our session context model, we first perform the necessary preprocessing. For instance, for the IP addresses collected from login data, we query each individual IP address and extract the corresponding city and ISP information. As a total, we obtain 33 different cities and 19 different ISPs.

Similarly, we convert login-time properties for each entry into six different time intervals in 24-hour format such that we treat each login time as morning, lunch-time, afternoon, etc. For browser version, we use abbreviated version of these entries such that Chrome v.x is converted to Chrome and Firefox v.x is converted to Firefox, etc. After the feature extraction, we use one-hot encoding technique to transform our categorical features into mathematical equivalents to feed all of these data into our machine learning model. Categorical feature in our context is the text data such as city, browser etc. which cannot be interpreted by machine learning algorithms by themselves. As a concrete example, because mathematical equivalents of categorical features are 1 or 0 (true or false), if an entry comes from a city X, the feature for the city X becomes 1 and 0 for all other cities. A similar procedure is applied for all other categorical features.

When we evaluate our session context model, our algorithm successfully differentiates real users from attack simulations with EER (Equal Error Rate) of 9% using threshold 0.95. The obtained scores and ROC curve for session context are shown in Figure 4 and Figure 5.

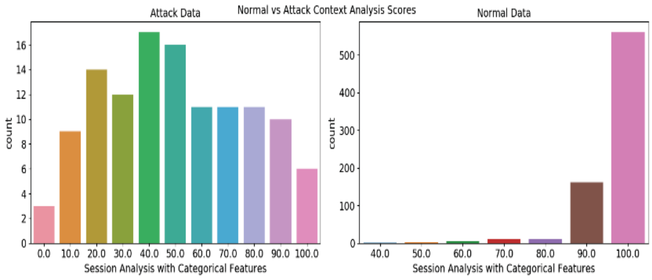


Fig. 4. Session Context Anomaly Scores.

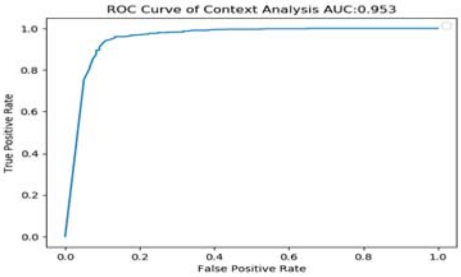


Fig. 5. ROC Curve for Session Context.

C. Ensemble Model with Weighted Average

After we analyzed two models above, we construct the combination of these models and test whether the performances of the ensemble models are better than separately implemented components. To do that, we use an approach where weighted averaging based on voting convention is implemented.

More specifically, ensemble modeling is a machine learning method which relies on the probabilistic scores of multiple

[1] We note that Type-I and Type-II error types are not independent. Usually, a decrease in one of them means an increase in the other. One way to summarize the operating characteristics of the system is to look at the Crossover Error Rate, also known as the Equal Error Rate (EER). The system has parameters that can be tuned to adjust the two types of error to the point where they are equal. When the two are equal, their common value is called EER. EER is usually preferred as the combined measure, which provides an approximate representation of overall system accuracy.

machine learning models rather than one model to decide the final outcome. For instance, our ensemble model which consists of two machine learning components can be shown as the following:

$$y = ax_1 + bx_2$$

where x1 and x2 represents the probabilistic scores of two different machine learning components that ranges between 0 and 1. Also, a and b represents the coefficients of these models which satisfies the equation

$$a + b = 1$$

to ensure that the final outcome of ensemble model will also be between 0 and 1.

In our experiment, we use session context anomaly and keystroke dynamics components (x_1 and x_2) to construct our ensemble model and we experiment with different weights (a and b) for these models. The results we obtained are shown in Figure 6 and as expected when we give the weight of 1 for one model and 0 for another, we re-observe the performances of single implementations; however, when we change the weights of these machine learning models in a way that both machine learning models contribute to the final outcome, we observe that the performances of the ensemble model always outperform single implementations and the best parameters of weights are 0.6 for session context component and 0.4 for keystroke dynamics component. With these weights, we obtain EER of 6% with a threshold of 0.89. ROC Curves for different weights are shown in Figure 7.

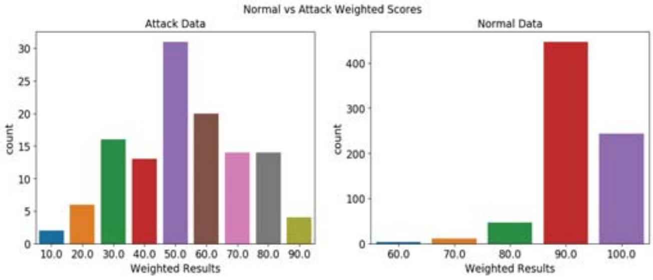


Fig. 6. Scores for the best performing ensemble model.

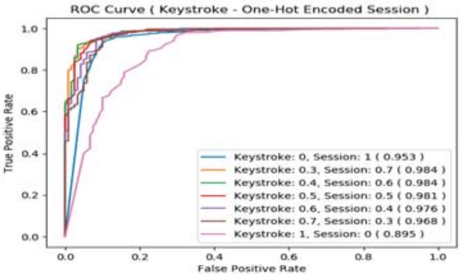
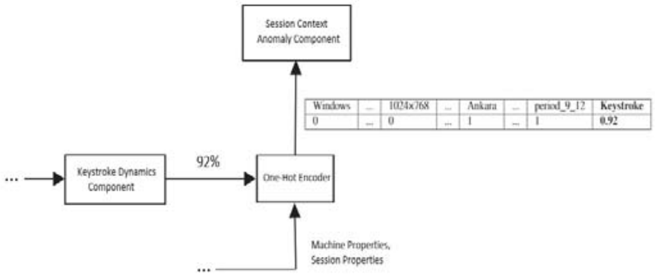


Fig. 7. ROC Curves for Ensemble Models with different weights.

D. Keystroke Scores as an Additional Feature in Session Context Model

As mentioned in session context anomaly component, we transformed the categorical dataset into a dataset that consists of only binary values to feed our session context anomaly component. For this part, additionally we also decide to retrain our session anomaly component with slightly modified dataset that consists of keystroke scores as an additional feature. For instance, when a user logs in, first keystroke dynamics component outputs the keystroke score of that individual login as previously and then one-hot encoder takes the keystroke score output from keystroke dynamics component and appends it to its dataset before training session context anomaly component. Figure 8 shows an example of a data-flow for this mode where keystroke score of an individual is 92% similar to previous keystrokes and the normalized score of this output, which is 0.92 in this case, is appended to the previously constructed one-hot encoded matrix. This modified session-context dataset has both previously constructed session context data and keystroke scores of each individual logins to be fed for this model.



Windows	...	1024x768	...	Ankara	...	period_9_12	**Keystroke**
0	...	0	...	1	...	1	**0.92**

Fig. 8. Keystroke score appended to the feature set of session context.

With this approach, we observe that it even outperforms the best performing weighted average model with the performance of EER 5% using threshold value of 0.91. Figure 9 and Figure 10 shows scores and ROC curve when keystroke score is added as an additional feature of the session context model.

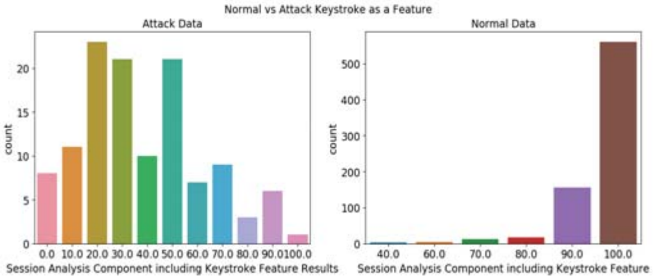


Fig. 9. Scores for Keystroke as an additional feature.

IV. Discussion and Future Work

Considering all of the presented models with their results (summarized in Table II), we see that the best performance is

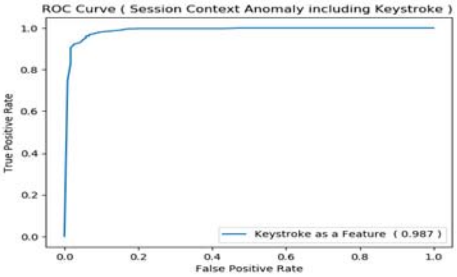


Fig. 10. ROC Curve for Keystroke as an additional feature.

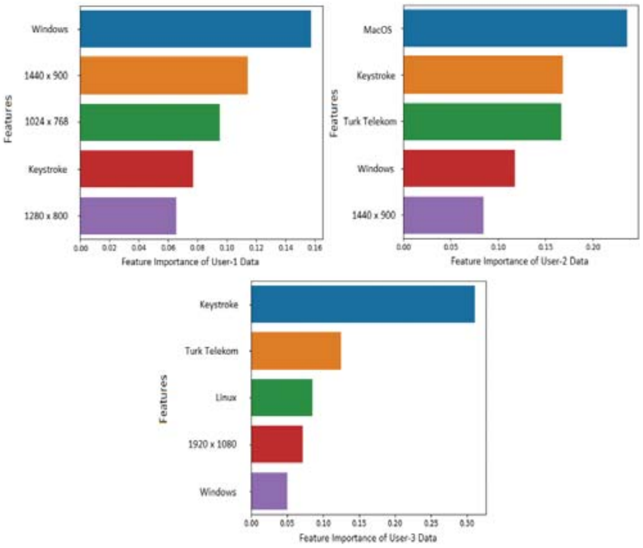


Fig. 11. Standard deviation (std) values for keystroke scores: std (User-1): 15.04%, std(User-2): 10.49%, std(User-3): 5.16%.

obtained with the model which uses keystroke dynamics scores as an additional feature in session context model. The reason behind it is that as some individuals' behaviour may oscillate more than others, either they change their posture while typing or use different keyboards more than others, using keystroke score as a feature gives the opportunity to the overall model (augmented session context model) to take this information into consideration as well. To observe how important keystroke feature is, we use feature importance algorithms provided by open source scikit-learn machine learning library. In scikit-learn, for the random forest algorithm there are two implementations to evaluate how important a feature is: permutation and gini importance [21]. In permutation importance, the feature importance is determined by permuting the features in the dataset and comparing how the performance is affected by removing or adding a specific feature; however, the algorithm assumes that the features and labels are highly uncorrelated but this hypothesis does not hold for highly-correlated anomaly detection schemes. On the other hand, in gini importance algorithm, the feature importance is determined by how much a feature decreases the entropy (impurity) in the decision trees.

Using gini importance algorithm, we extract the most important features of the models. In the bar plots of Figure 11, three samples (User-1, User-2 and User-3) plots represent the users whose standard deviation σ of keystroke dynamics are the highest, lowest and mid-range. In these plots, keystroke scores are also arranged in order (from high to low standard deviations). As seen in this figure, the importance of keystroke feature decreases as σ of keystroke scores increases and we correctly incorporate this information into our models. On the other hand, when we treat keystroke dynamics and session context anomaly models' scores separately, we lose the ability to consider behavioral deviations of the individuals, which results in a worse-performing model.

One of the drawbacks of this approach is that because session anomaly component uses the score of keystroke dynamics component as a feature, these components cannot function in parallel and they have to be implemented sequentially which creates additional latency. However, we do not expect this latency to exceed more than a few milliseconds so it is safe to say that it would not harm user experience.

In our work, we mainly focus on comparing performances of different anomaly detection approaches and finding better ways to combine them. We ignore some important details that may be useful for obtaining better performance. For instance, one of the most useful data as session context might be browser version which is abbreviated in our experiment as just browser type. During our experimentation we realize that browser version is a stair-like data which means when users update their browser, they no longer login from a previous browser version from the same device. Considering that browser version is ordinal categorical data, we believe there could be a better implementation than label encoding (such as the subtraction of current and last logged-in browser versions from the same device must be greater than or equal to zero because browser version always increases as it gets updated). Another important contextual information that might be incorporated into the model in a future work is the number of failed login attempts before users successfully logged in. This information could be used for instance simply by increasing threshold value as the number of failed logins increase.

Metrics	Keystroke	Session	Weighted	KaaF
Equal Error Rate (EER)	18.67 %	9.1 %	6.3 %	5.29 %
EER Threshold	86 %	95 %	89 %	91 %
Precision (Legitimate)	96.51 %	98.42 %	99.01 %	99.16 %
Recall (Legitimate)	80.98 %	90.96 %	93.22 %	94.41 %
F1-Score (Legitimate)	88.07 %	94.54 %	96.03 %	96.73 %
Precision (Attack)	40.66 %	61.58 %	68.90 %	73.08 %
Recall (Attack)	81.67 %	90.83 %	94.17 %	95 %
F1-Score (Attack)	54.29 %	73.40 %	79.58 %	82.61 %
Accuracy	81.08 %	90.94 %	93.35 %	94.5 %

TABLE II
SUMMARY OF PERFORMANCE RESULTS - LEGITIMATE LOGINS (752 TEST SAMPLES), ATTACK LOGINS (120 TEST SAMPLES) (KAAF IS SHORT OF KEYSTROKE AS A FEATURE).

V. Limitations

We think that one of the reasons why our keystroke dynamics model has performed relatively poorly is that for privacy concerns we assigned users the passwords instead of allowing them choose themselves. This choice might make some of the keystroke scores of legitimate users lower than expected i.e., users would be more comfortable and could show a more distinguishable pattern while typing their already familiar and memorized passwords.

For the session context data, since most of our data were collected while students login from the university campus, the contextual variations were not as high as the data that would be observed in some other settings. We conjecture that for instance in case of e-banking website, the variations in legitimate data would be much higher and therefore we could expect lower session context performance results.

In terms of implementation difficulties, we observe that some users have password manager or auto-fill feature enabled in their browser and because our keystroke dynamics component requires the entries of username and password fields by hand, we had to delete the required fields even if auto-fill feature is enabled and asked the users to re-enter their credentials. We believe that this requirement is one of the usability drawbacks of keystroke dynamics in general. Secondly, we realized that some smartphone keyboards do not support key-up and key-down events and these events are fired at the same time which cancels out one of the two keystroke dynamics features mentioned in Section 3.A and reduces the performance of keystroke dynamics significantly. We decide to revisit smartphone keyboard issue in a future study because it requires either complete smartphone keyboard overhaul or smartphone specific redesign of the experiment.

VI. Related Work

The literature on password-based authentication is enlarging with contributions from different disciplines. It is well understood that there is no authentication mechanism which performs better in terms of usability and deployability than password-based schemes and it is no longer expected that they could be totally replaced in the near future [1].

It is also well known that users are inclined to pick weak and guessable passwords [17]. To deal with this predicament, there are approaches to detect legitimacy of logins by location-based [2], [9], device fingerprinting [3] and behavioural-biometrics [4], [8], [12], [15], [16], [18] based approaches. The main incentive of these methods is to keep the authentication user-friendly and even if the chosen password is weak, they can detect whether the user is really the one that he/she claims to be. Another advantage of these complementary techniques is that the collected data may not be easy to replicate because they cannot be obtained as easily as passwords themselves [19].

One of the biggest concerns of these implicit authentication methods which minimize the change in user experience is the error rates observed as either false-reject or false-accept rates. It is also shown that contrary to the naive expectation, there are occasions where replication attacks against behavioural biometrics schemes are feasible [7]. This result indicates that even when behavioral biometrics has acceptable error rates, it is still necessary to implement supplementary methods such as contextual authentication.

The term risk-based authentication has gained popularity in recent years and many corporations started adapting these methods in their applications. Wiefling et al. [10] investigated how major companies adopted risk-based authentication techniques and showed that even though there is no general consensus for the implementation of risk-based authentication, they concluded that IP address is the most important identifier to decide legitimacy of the current login. When the IP address of current login changes, all the companies they have analyzed trigger multi-factor authentication.

In an internal network, detecting anomalies in case of a network breach is somewhat a different problem. For instance, in Siadati et al. [11], the authors proposed to divide the network architecture into multiple domains and correlated the network traffic to detect malicious login activity not consistent with the previous traffic.

To address scalability issues of sparse contextual data, Freeman et al. [5] proposed a statistical approach for contextual data where they computed probability estimation of each feature and applied smoothing for unseen features, which basically decreases the weights of the probability for observed features and increases the weights for unobserved events.

Most of the risk-based authentication work were concentrated on contextual data but recently, in Solano et al. [14] the authors analyzed the performance improvement of anomaly detection by combining multiple machine learning components. In their study, they retrieved the behavioral data which contains both mouse and keystroke dynamics from a public dataset called The Wolf Of SUTD and the session context data from an in-house application. In contrast to their work, both keystroke and session context information for both legitimate users and attack simulations are collected from the same application in our study. We also tested for the first time the model in which we added keystroke scores to our session context model as an additional feature, which result in decreased error rates.

VII. Conclusion

One of the widely used methods to mitigate the security problems with respect to weak or stolen passwords is two-factor authentication. Even though two-factor authentication is effective, it is often not an appreciated approach by general public because of its usability drawbacks. Therefore we believe improving the security of password-based authentication while keeping its usability advantages is crucial for an effective and usable cyber security policy. With such a policy in place, two factor authentication could be activated only when an anomaly in password-based authentication is detected.

In this work, using a dataset collected in a coherent and realistic setting we compared different authentication anomaly detection approaches and presented different methods to combine keystroke dynamics and contextual information components. When we included the authentication scores of keystroke dynamics as an additional feature in our session context model, our findings showed that improved results could be obtained. This improvement comes from the fact that the proposed combination allows us to incorporate the deviations of keystroke scores of individuals into our augmented machine learning model.

References

[1] C. Herley and P. van Oorschot, "A research agenda acknowledging the persistence of passwords," IEEE Security & Privacy, vol. 10, no. 1, pp. 28–36, 2012.

[2] D. E. Denning and P. F. MacDoran, "Location-based authentication: Grounding cyberspace for better security," Computer Fraud & Security, vol. 1996, no. 2, pp. 12–16, 1996.

[3] P. Eckersley, "How unique is your web browser?" in Privacy Enhancing Technologies Symposium (PETS). Springer, 2010, pp. 1–18

[4] H. Khan, A. Atwater, and U. Hengartner, "A comparative evaluation of implicit authentication schemes," in RAID. Springer, 2014, pp. 255–275.

[5] Freeman, D. & Dürmuth, M. & Biggio, Battista. (2016). Who are you? A statistical approach to measuring user authenticity. Proc of NDSS 2016. 1-15.

[6] Florencio, Dinei & Herley, Cormac. (2007). A large-scale study of web password habits. 16th International World Wide Web Conference, WWW2007. 657-666. 10.1145/1242572.1242661.

[7] Khan, Hassan & Hengartner, Urs & Vogel, Daniel. (2016). Targeted Mimicry Attacks on Touch Input Based Implicit Authentication Schemes. 387-398. 10.1145/2906388.2906404.

[8] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang. Silentsense: silent user identification via touch and movement behavioral biometrics. In 19th Annual International Conference on Mobile Computing & Networking. ACM, 2013.

[9] Z. Dong, R. D. Perera, R. Chandramouli, and K. Subbalakshmi, "Network measurement based modeling and optimization for IP geolocation," Computer Networks, vol. 56, no. 1, pp. 85–98, 2012

[10] Wiefling, Stephan & Lo Iacono, Luigi & Dürmuth, Markus. (2019). Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild. 10.1007/978-3-030-22312-0-10 .

[11] Siadati, H., & Memon, N. (2017). Detecting structurally anomalous logins within enterprise networks. In CCS 2017 - Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (Vol. Part F131467, pp. 1273-1284). Association for Computing Machinery. https://doi.org/10.1145/3133956.3134003

[12] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In Information Security. Springer, 2011.

[13] F. Breitinger and C. Nickel, "User survey on phone security and usage," in BIOSIG, 2010, pp. 139–144.

[14] Solano J., Camacho L., Correa A., Deiro C., Vargas J., Ochoa M. (2019) Risk-Based Static Authentication in Web Applications with Behavioral Biometrics and Session Context Analytics. In: Zhou J. et al. (eds) Applied Cryptography and Network Security Workshops. ACNS 2019. Lecture Notes in Computer Science, vol 11605. Springer, Cham

[15] M. Jakobsson, E. Shi, and R. Chow, "Implicit authentication for mobile devices," in 4th USENIX Workshop on Hot Topics in Security (HotSec '09), Montreal, Canada, August 2009.

[16] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and I know it's you!: Implicit authentication based on touch screen patterns. In Annual Conference on Human Factors in Computing Systems. ACM, 2012.

[17] A. Adams and M. A. Sasse, "Users are not the enemy," Communications of the ACM, vol. 42, no. 12, pp. 40–46, 1999

[18] B. Draffin, J. Zhu, and J. Zhang. Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In Mobile Computing, Applications, and Services. Springer, 2014.

[19] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In 19th Annual International Conference on Mobile Computing & Networking. ACM, 2013.

[20] Joseph Bonneau,Authentication is machine learning. https://www.lightbluetouchpaper.org/2012/12/14/authentication-is-machine-learning/. Last accessed on 03/19/2020.

[21] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html/. Last accessed on 6/30/2020.