Internet Fraud: The Case of Account Takeover in Online Marketplace

Ricardo Kawase
rkawase@team.mobile.de
mobile.de GmbH
14532, Marktplatz 1, Dreilinden
Germany

Francesca Diana
francesca.diana@codecentric.de
codecentric AG
10179, Köpenicker Str. 31, Berlin
Germany

Mateusz Czeladka
mszczap@team.mobile.de
mobile.de GmbH
14532, Marktplatz 1, Dreilinden
Germany

Markus Schüler
maschueler@team.mobile.de
mobile.de GmbH
14532, Marktplatz 1, Dreilinden
Germany

Manuela Faust
mfaust@team.mobile.de
mobile.de GmbH
14532, Marktplatz 1, Dreilinden
Germany

ABSTRACT

Account takeover is a form of online identity theft where a fraudster gains unauthorized access to an individual's account in a given system. Depending on the system, this unauthorized access can lead to severe consequences of privacy breach and financial loss to the victims, to the companies that maintain the system and to other users. In this paper, we present the work done in order to *prevent* and *detect* account takeovers at *mobile.de*, an online vehicle marketplace. To tackle the *prevention* problem, we first present a behavioral analysis of how fraudsters operate, and implemented a mutual two-factor authentication that achieved a reduction of 43% of account takeovers. To tackle the *detection* problem, we introduce a concept drift sensitive machine learning training approach that was able to improve our baseline methods by 18% in detection rates. The automatic detection reduced the exposure of fraudulent listings by 69%, resulting in a safer marketplace for buyers and sellers.

CCS CONCEPTS

• **Information systems → Electronic commerce**.

KEYWORDS

Online marketplaces; internet fraud; account takeover; fraud prevention; fraud detection; mutual two-factor authentication; compromised accounts

ACM Reference Format:
Ricardo Kawase, Francesca Diana, Mateusz Czeladka, Markus Schüler, and Manuela Faust. 2019. Internet Fraud: The Case of Account Takeover in Online Marketplace. In *30th ACM Conference on Hypertext and Social Media (HT '19), September 17–20, 2019, Hof, Germany.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3342220.3343651

1 INTRODUCTION

Mobile.de is Germany's largest vehicle marketplace online. Every day, thousands of online listings are posted by users seeking to sell their vehicles. *Mobile.de*'s folksonomy can be summarized in 2 main entities, namely users and listings (a vehicle ad), wherein users are classified in buyers and sellers. Further, there are essentially two types of sellers: private sellers and professional sellers (car dealers). A private seller is the regular user that owns a car and would like to advertise in the platform for potential buyers, while professional sellers are those users that usually represent a physical vehicle dealership. For simplicity, in the rest of this paper we will identify the latter as "dealer".

It is very important to mention that dealers are paying customers of *mobile.de* - each dealer has a contract with *mobile.de.* As of March 2019, *mobile.de* has over 40,000 active dealer accounts. They are one of the main contributors for *mobile.de*'s revenue as well as listing inventory (around 70% of the listings are dealer's listings). With that in mind, it is very important that *mobile.de* does its best to protect these customers against malicious attacks that try to take advantage of the business model and the system in place.

Fraud fighting is a segment of risk management. In online marketplaces such as *mobile.de*, fraud can occur in different forms, implying different risks. For example, persons who: abuse the system by creating several accounts; use the system to publish listings that are explicitly forbidden in the *General Terms and Conditions*; enter false information in their listings; are debtors; and the problem tackled in this paper, a person who impersonates another user with stolen credentials namely *Account Takeover* (in the rest of this paper we identify it as ATO).

There are several layers of fraud fighting depending on the scenario. In the ATO scenario, we have basically three layers: *Prevention*, *Detection* and *Enforcement.* As a general rule, a system must try to prevent fraud, in case it happens it should be able to detect it, and once it is detected it must enforce certain actions such as exiting the malicious actor from the system. *Prevention* is the end-goal, however as counter-intuitive as it may sound, you cannot implement *prevention* before having *detection* in place. The reason is simple, you cannot build prevention until you know what you

HT '19, September 17–20, 2019, Hof, Germany
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6885-8/19/09…$15.00
https://doi.org/10.1145/3342220.3343651

are trying to prevent, and the only way to find out is by detecting fraudulent events.

In this paper, we describe our work to mitigate account takeovers in two different layers. First, the prevention case, where the challenge in hand is to stop a fraudster before he is able to take over the account of a customer. Second, the detection case, where the challenge is to detect and suspend a compromised customer account as soon as possible (*enforcement*), ideally, before any damage is done.

In this paper, we will thoroughly describe all the aspects of fraud events in the account takeover use-case. First, we present a summary of related work and compare them to our work. Next, in Section 3, we describe how fraudsters operate. After that, in Section 4, we describe our initiatives in order to tackle the *prevention* aspect of account takeovers, together with its outcomes. in Section 5, we tackle the *detection* aspect, including data analysis, feature engineering, machine learning modeling strategy and results. In Section 6, we list the most important learnings and implications of our work, providing guidelines for other systems facing the same risks. We finalize our paper with an outlook, drawing our next steps to improve our fraud fighting mechanisms in Section 7 and our conclusions in Section 8.

2 RELATED WORK

In recent attempts to mitigate account takeovers, Marforio et al. [16] evaluate the effectiveness of personalized security indicators. Basically, they evaluate the effectiveness of personalized login pages in supporting users to identify spoof ones. The idea behind is that users are able to configure how their login pages should look like. Unfortunately, we see two problems in this approach. First, the personalized page can be replicated by the spoof website. Fraudsters at *mobile.de* aim at specific dealers, and due to the nature of the business, they are able to get contact details of each dealer and the dealers' system identification number. By attempting a login with these information, a personalized login page would be displayed to the fraudster. Personalized login solutions only work when the attacker does not have information on the victim. Second, as the authors mentioned in their paper, users are inclined to trust when the personalization announces a failure or maintenance. This is a workaround that could be easily reproduced in spoof websites. Our solution also requires diligence from the user, however the randomization of the security indicator cannot be replicated on online middleman attacks, and provides a more robust mutual two-factor authentication.

In a similar effort to address the pre-payment scams, Edwards et al. demonstrate that a semantic analysis of the content of messages exchanged between fraudsters and victims, can be automatically classified with a very high confidence [9]. There are a few differences that make our scenario more complex. On the scams described by Edwards et al., most of the persuasion stages applied by fraudsters require specific approaches (and specific choices of words) to build trust. At *mobile.de*, trust already exists. The buyer believes that he is negotiating with a legitimate dealer, and the nature of the messages exchanged on fraud-deals and legitimate-deals are very similar. Second, most of the negotiation happens offline, and the pre-payment persuasion happens over the phone or by email, channels which are out of the control of our fraud detection systems.

Other existing solutions, which virtually eliminates account takeovers, is the use of external hardware such as Yubikeys[1] or smart-cards. The effectiveness of Yubikeys have been studied and proved [15]. Also, smart-cards have been extensively studied [24, 27], evaluated [23, 24] and improved [25], however in our scenario, the adoption of these solutions is very limited when not enforced. At *mobile.de*, dealers are strongly against any solution that involves an additional piece of hardware or device. They use *mobile.de* as a working tool and it is unacceptable to be locked out of the system if they do not have the two-factor authentication device in hand. Further, in many cases, sales agents who interact with *mobile.de*'s website, do not have the access or permission to plug external devices in their working machine. Thus, our work focuses on providing a solution that not only tackles the problem, but it is also accepted and adopted by users.

To the best of our knowledge, Abdallah et al. [2] provide the most complete overview on tackling the fraud detection challenge. In this paper, we applied several of the learnings and introduced a time shifting training approach to integrate concept drifting. Several works have proposed techniques to incorporate concept drift in fraud detection scenarios: Cross-validation decision tree ensemble methods that handles concept drift [11], mining concept drifting data streams using weighted ensemble classifiers [26] or adaptive model that addresses fluctuation and evolution on users behaviors [19]. Our solution is closely related to the work proposed by Fan [11], as we incrementally introduce past data to improve the model detection. The main difference is that we do not need to perform cross-validation to mine the concept drift. We assume that the concept drift, if there is any, is already represented in the most recent part of the data which is used for testing, resulting in a simplified, yet effective solution.

3 ACCOUNT TAKEOVER - THE ANATOMY

Mobile.de does not control transactions between buyer and sellers. It is a "matchmaking" platform that bridges the gap between the two sets of entities. Once a buyer contacts a seller (in sales terminology a "lead"), the system's job is basically done. Further contacts between the parties, negotiations, product (vehicle) verification, paperwork and payment are all done offline. Given this setup, the first question that comes to mind is "How does a fraudster profit from taking over sellers' accounts?". A fair question that has a very simple answer: pre-payments (also known as advance-fee scam).

Once the fraudster has taken control of a dealer account, his goal is to get as many leads as possible, as fast as possible, before the owner of the account (or the system) identifies that a breach has happened. To achieve this, fraudsters take a series of *lead-boosting* steps. They upload listings of high-demand vehicles into the marketplace, they set a very low yet reasonable price for the vehicles and they book pay-per-use features of the system that increase the visibility of the listing in the platform (e.g. top of the page ad, or first page ad, among others). Last but not least, they make sure to add new contact information in the dealer account or in the listings. If all these steps are done undetected by the system, the bait is set, and for a few minutes or hours, potential buyers start to contact the fraudster.

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

Since every aspect of the listing looks legitimate (the website, the dealer and the vehicle), buyers lower they guard and contact the fraudster, who impersonates a legitimate seller. During the contact, via email or even phone call, the social engineering begins. Research into social engineering [18] showed that the attacks often follow a simple process: gather information about the target, develop and exploit a trust relationship, and utilize the gathered information.

In our case, the fraudster convinces the potential buyer (now a victim) that the vehicle is a once-in-a-lifetime deal, however he has already several other interested buyers, and to hold this vehicle for the victim, he requires a deposit of a certain amount of money as prepayment. Many times, the victim has already been in contact with the real dealer, maybe even knows the physical location, which increases his trust and confidence on the on-going negotiation. Convinced, the victim transfers the money to the fraudster and the damage is done.

Once the victims finally realize their mistake, they contact *mobile.de*'s customer support and report the case. The previous paragraphs are a summary of these unfortunate reports. Monthly, there are very few cases that reach this point, however, the total monthly loss can go up to thousands of Euros.

3.1 Phishing and Spoofing

Fraudsters steal dealers' credentials with spoof websites. These websites are indistinguishable from *mobile.de*'s login page, except by the URL address. If any dealer enters his credential in a spoof website, his account at *mobile.de* is compromised. To get dealers to do so, fraudsters contact the dealer themselves with phishing emails, and more effectively, with SMS messages containing a link to a spoof website. They send dealers a message, pretending to be from *mobile.de*, saying that they must log in as soon as possible to verify their account or there will be some consequence to their online inventory. Unadvised, the dealers access the spoof website and provide their credentials. As a rule, dealers are very easy to contact. They always have their emails and phone numbers (many times mobile numbers) available on their listings.

Customer support at *mobile.de* has reported several dealers, victims of SMS phishing, simply because on a mobile device it is much harder for the user to identify a spoof URL. Whenever a spoof website is reported, customer support triggers the legal department which contacts the spoof website host and is able to take it down once the case is verified.

3.2 Increasing Spoof Website Longevity

We have mentioned that *mobile.de*'s customer support and legal teams are able to contact the spoof website hosts, and in short time, take them offline. These websites are detected from the complaints coming from dealers that were victims of phishing. The exact processes and steps required for taking down a spoof website is out of the scope of this paper. To give an idea of the scale of the problem, in 2017 and 2018 we detected and reported respectively 133 and 109 spoof websites. Almost every month, 10 different websites trying to mimic *mobile.de* are reported.

Our investigations revealed a new trend that started in 2017, that extended the online time of certain spoof websites. Instead of creating a whole new page and domain, fraudsters were hacking existing pages of legitimate small business, blogs, etc, and adding the spoof *mobile.de* login page into it. We have found a few legitimate websites with WordPress[2] infrastructure that contained subfolders hosting these malicious pages. Additionally, several of these webpages had a HTTPS certificate, which theoretically increases their trustworthiness. In cases like this, the process to put the spoof pages offline takes longer. The host does not simply remove access to the website. Instead, the owner of the website is contacted to fix the problem, and it is reasonable to assume that small businesses do not self maintain their websites. Thus, they also need no contact their webmasters. With many parties involved and with the natural delay in communication and execution, the spoof pages remain online for longer, and each minute longer increases the chances of new account takeovers.

A common approach to protect users' accounts is the use of two-factor authentication. In most cases, the two-factor authentication would be able to solve the problem, preventing fraudsters to login with the users' stolen credentials. Although *mobile.de* has such level of security in place for many years, a committed fraudster can easily bypass that.

3.3 Bypassing two-factor authentication

In order to bypass two-factor authentication, fraudsters perform an online middleman attack. What we describe as middleman attack, goes in the same direction of what Bursztein et al. described as “manual hijackers" [8], a type of attack where fraudsters spend a significant non-automated effort to accomplish the wrongdoing. First the spoof website is configured to redirect the user to a spoof TAN-validation page after the victim enters the credentials. While the victim is waiting for an SMS in the spoof website, the fraudster, who already has the user-password combination of the dealer, quickly accesses *mobile.de* on his own device and tries to log in. *Mobile.de* systems detect a login attempt in the account and redirects the fraudster to the two-factor authentication page where he must now enter a TAN number. *Mobile.de* sends the TAN verification number to the dealer, who is expecting one at that moment. The dealer receives the SMS with the TAN and enters it in the spoof website. At this point, the TAN is sent to the fraudsters, who can effectively take over the dealer's account.

Since we are dealing with an online middleman attack, which means that the fraudster is in standby mode waiting for the victim, any other two-factor authentication method that does not effectively validate the device which is trying to login, would also fail. Email, PingID[3], Google Authenticator[4], etc…have the same drawback: the fact that the victim is not aware that the two-factor authentication request that he is about to authorize is not his own request, but a request started by the fraudster.

Over the years, our fraud-fighting team at *mobile.de* has gained knowledge on how spoof websites operate by investigating them. Although fraudsters are tech-savvy, they rarely protect their own

[2] http://wordpress.com/

[3] https://www.pingidentity.com/

[4] https://www.google.com/landing/2step/

spoof websites. In some of them, we were able to inspect the server-side code of PHP pages that implements the behavior we just described. A simple piece of code, gets the submitted values (user_id-password) and sends via email to a given address. Thus, when the credential is submitted, an email and an alert is sent to the fraudster himself to initiate the online middleman attack.

3.4 Timing the attack

Once the dealer account has been breached and the fraudster is logged in, one would expect an accelerated approach, where the fraudster would try to do most of his actions as fast as possible before he is detected. However, the most common modus operandi taken by fraudsters is a stealth approach.

Fraudsters stay logged in for several hours (even days), unnoticed, waiting for the right moment to attack. If no action is performed, the dealer, the users, customer support, and the system have no clues that something is happening. So far, only the *login event* happened, and that was approved by the dealer via two-factor authentication.

We have previously mentioned the next steps of the fraudsters – modifying contact information, publishing/editing attractive listings, and booking *lead-boosting* features. In our analysis of past account takeovers cases, we have identified and correlated two main factors that seem to influence the time chosen for the fraudster to start interacting with the system. *Mobile.de*'s web traffic volume and customer support working hours. The highest volume of users at *mobile.de* is in the evening, after working hours (between 7pm and 10pm). This is the time where most users access marketplaces, significantly increasing the amount of users that see the the fraudlent listing before the system or the dealer can detect it.

4 PREVENTING ACCOUNT TAKEOVER

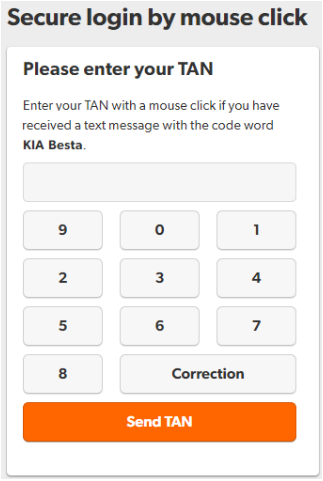
The most effective and preferable way to eliminate the account takeover problem is to prevent it before it happens. The costs to deal with it afterwards are significantly higher in terms of processes. For example, when an account is compromised, there are several steps that require attention such as: restoration of the dealer's inventory, restoration of dealer's profile information, complete reset of all credentials associated to a customer account, data rollback, reimbursement of charges done in the dealer accounts, and finally, the reactivation of the account.

On the other hand, if an account takeover attempt is detected before it happened, i.e. the prevention of it, the only necessary action of the system is to reset user credentials to reinforce security.

Unfortunately, there is not much a system can do to prevent that. The only event that produces any useful data is the login-attempt which conveys very little information about the user behind it. Thus, a solution must be implemented on a higher level of abstraction, and in our case is the two-factor authentication itself.

To tackle the prevention of account takeovers we implemented a *mutual two-factor authentication*. The mutual two-factor authentication allows the system to validate the user and allows the user to validate the system.

At the TAN validation page of *mobile.de* the user is faced with a randomly generated keyword. More specific a pair of a car-make and a car-model. Each login attempt gets a different make-model combination displayed on the interface. Figure 1 depicts *mobile.de*'s TAN validation page, where the user is presented with the randomly generated keyword "KIA Besta". At the same time, the user receives an SMS message (Figure 2), containing the same keyword. He should enter the TAN code given in the SMS only if he sees the keyword on the page.

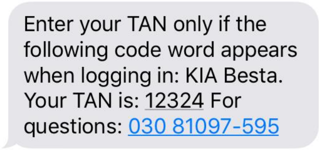


Secure login by mouse click

Please enter your TAN

Enter your TAN with a mouse click if you have received a text message with the code word **KIA Besta**.

Figure 1: Mobile.de TAN validation login page.



Enter your TAN only if the following code word appears when logging in: KIA Besta. Your TAN is: 12324 For questions: 030 81097-595

Figure 2: Mobile.de SMS message sent to a dealer.

If instructions are properly followed, this should eliminate the middleman attacks. In middleman attacks, when the dealer enters the credential on a spoof website, he must be immediately redirected to a spoof TAN validation page. However, this page will not contain the randomly generated keyword that will only be available once the fraudster tried to login on *mobile.de* with the stolen credentials.

In addition to that, to prevent scripting in this middleman attack procedure, the user interface prevents the typing of the code. It must be clicked on the displayed keypad where the numbers are also randomly shuffled. This implementation was effectively rolled out in May 2018. Figure 3 depicts the volume of dealer's account takeovers 12 months before, and 12 months after the release of this implementation (we normalized the numbers, dividing them by the maximum observed value). Comparing before and after the implementation, we see a reduction of ATO cases by 43%.

This mutual two-factor authentication was the only major implementation to tackle ATO prevention (together with a few dealer education initiatives), and we reasonably assume that it was the main contribution for this reduction in ATO cases.

Nevertheless, the number of ATOs is still higher than we would like, and no matter how sophisticated our tools are, user behavior

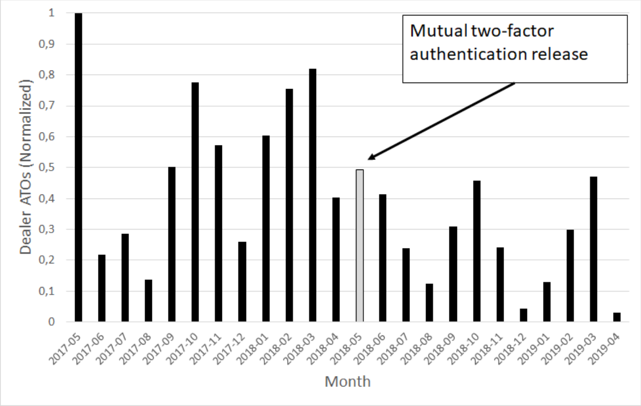


Figure 3: Normalized volume of dealer account takeovers per month (all numbers were divided by the maximum observed value).

and education plays a significant role in preventing malicious activities. We recurrently inform dealers (through mailing lists and in-person training) about the existing risks on the internet, and how they should be aware of protecting their accounts.

Our educational campaigns efforts are in line with the findings of Shay et al. [22]. They have demonstrated that users acknowledge some responsibility for keeping their accounts secure, and that their understanding of important security measures is incomplete. Unfortunately, that is not trivial, and previous literature has shown limited patience and cooperation of users for security measures [3, 5].

Often, users do not read the content of the SMS messages entirely, and simply rush to enter the TAN into the verification page. We have tried to mitigate this problem by carefully designing the SMS message in a way that the TAN code is at the end of the message. With this design, mobile devices that provide messages previews of the first N characters of the message, will not include the TAN number, forcing the user to open the message and hopefully noticing the unique mutual verification keyword.

Since every system, including *mobile.de*, will always have inattentive users that are potential victims of scams and account takeovers, we must also tackle the next problem in hand, i.e. detecting ATOs once it has already happened.

5 DETECTING ACCOUNT TAKEOVER

Identifying when an account has been hijacked is a result that in time will eventually happen. The owner of the hijacked account, or other users who use the system and interact with this account, will be able to identify it. The real challenge is to detect an account takeover as soon as possible, i.e. automatically. To achieve this goal, we deployed a machine learning, multi-variate Bernoulli Naive Bayes [17] model that, for each dealer-listing event, computes a score of the likelihood that this dealer account is under the control of a fraudster. The whole work consists of data gathering, feature engineering, model training, persistence, and finally real-time predictions. This machine learning model was rolled out in March 2018 in order to replace a legacy rule-based fraud detection implementation.

5.1 Data overview

The foundation of a good fraud detection mechanism is based on the data available and how we can smartly derive relevant features out of it. In this section, we will as much as possible, provide all information necessary for the reader to understand our detection system, but we will deliberately omit details that could potentially provide information on how to circumvent the system. We categorize the data we used in our machine learning in four main groups:

- **Dealer login**: these data sources include information of login events of the dealers. It contains data points such as device fingerprint, browser info, IP address, request status (success, failure), timestamps, etc. In average, we observe around 85,000 dealer login events on a daily basis. From these events we are able to engineer features that describe the dealer login behavior: how often each dealer logs into the system, duration of sessions, preferable day of the week, time of the day, devices used, and so on. With these information, we are able to support our detection method in identifying out of the ordinary login events in a dealer account.
- **Dealer inventory**: these data sources contain information of the dealer usual inventory. For example, some car dealers only deal vehicles of a certain price range, or from a particular car producer. Some other dealers only deal new cars (near zero mileage), different from other dealers specialized in used cars. These data help to identify whenever an unobserved outlier vehicle enters the inventory of a given dealer.
- **Dealer behavior**: In *mobile.de*, there are different methods that a dealer can use to upload a listing: using service providers, manual uploads, using an API or importing from different sources or file types. We combine this information with other data points to build the regular dealer behavior model. It describes how often dealers add new listings, how fast they do it, which channels they use, and when they do it. We additionally include other actions such as booking of pay-per-use features that increases listing visibility. Previous works have demonstrated the effectiveness of characterizing users' behavioral patterns, and leverage this knowledge to later detect compromised accounts [10, 20].
- **Listings make-model**: These data contain the profiles of vehicles. Given a vehicle and the whole set of the its features, together with some additional metadata such as mileage, vehicle condition, location, we are able to estimate the price or any of the vehicle's feature. In addition to that, we have estimations of demand on each vehicle type. This supports the detection system in identifying suspicious outliers. In a very simple example, the system can detect when a given vehicle is too cheap, aiming at attracting more leads.

At the end, we have a total of 27 top-level features that we use to predict the probability that a listing of a dealer was uploaded or modified during an account takeover.

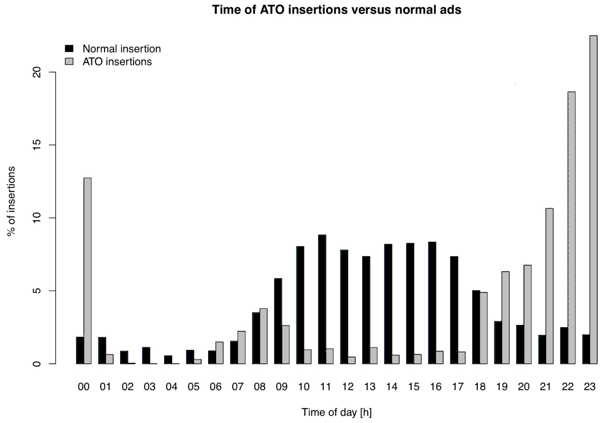


Figure 4: Distribution of legitimate listings and ATO listings per hour of the day.

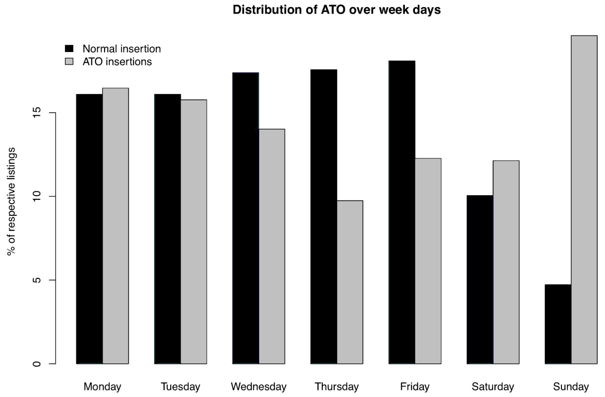


Figure 5: Distribution of legitimate listings and ATO listings per day of the week.

5.2 Data Analysis

We conducted several analyses in each of the features used in our model. In this subsection, we list a few (most interesting) examples that depicts some of the relevant features used to detect account takeovers. As we have previously mentioned, fraudsters have particular times when they perform the attacks. Figure 4 clearly depicts this behavior. We can see that non-ATO listings are uploaded more often during working hours (between 8:00 and 18:00), and ATO listings are more likely to come online at late hours (between 18:00 and 1:00).

Figure 5 shows the same phenomenon for day of the week: On Sundays only around 5% of non-ATO listings are published, in contrast to around 20% of the ATO listings. We have mentioned that SMS phishing is more effective due to design factors of mobile devices. We know which phone number dealers expose to the public, and can distinguish between mobile and landline numbers. Figure 6 clearly depicts this occurrence, showing that 80% of ATO listings insertions happens on dealer accounts which have their mobile phone number exposed to the public.

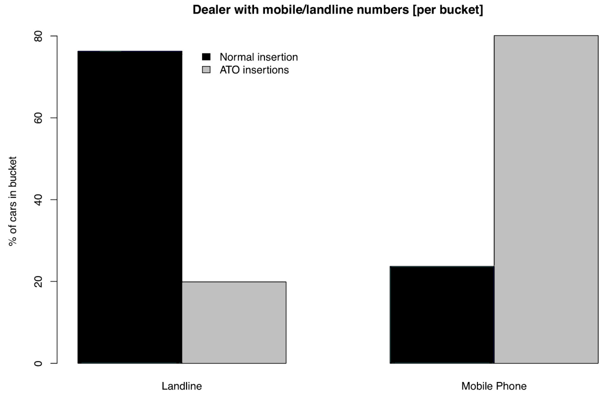


Figure 6: Distribution of legitimate listings and ATO listings per phone type (mobile or landline)

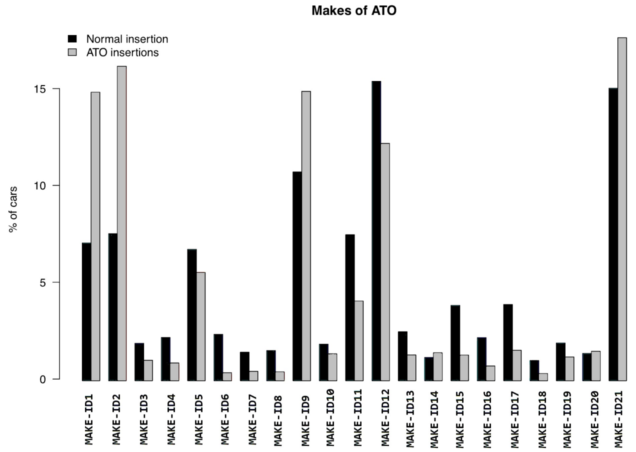


Figure 7: Distribution of legitimate listings and ATO listings per car make (make names have been anonymized).

Further, we looked at characteristics of the listings to learn predictive features. For example, Figure 7 shows the distribution of non-ATO and ATO listings in different vehicle manufacturers (makes) and more specifically, on make-models (Figure 8).

Another important discriminative feature (not depicted) is the price distribution of non-ATO and ATO listings. We know that ATO listings are usually assigned with a more attractive (lower) price, and this is validated in the observation of the data.

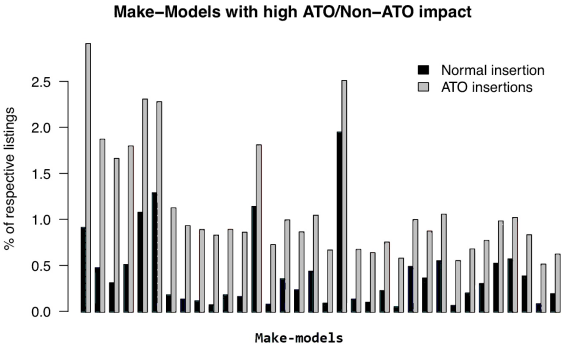


Figure 8: Distribution of legitimate listings and ATO listings per car make-model. Each pair of columns represents a vehicle make-model.

From each feature analysis[5], it is possible to understand the impact of it on fraud detection and estimate a fraud probability, a task performed during the training of the model.

5.3 From Rules to Machine Learning

For decades, services providers have been tackling the fraud detection challenge with rule-based approaches. Up to date, several financial institutions and online websites, still apply static rule-based checks to detect whether a user or a transaction is malicious. A lot of time must be invested to evaluate rules' performance with recurrent analysis, fixing weights, orders and exceptions.

Nevertheless, these rules can be reverse engineered by the fraudsters, bypassed, and become obsolete. Today, several companies struggle with legacy fraud check implementations, where new technology demands the implementation of new rules, which in some cases can contradict each other. In rule-based implementation, one solution to retire old rules is to constantly monitor their individual performance (e.g. precision and recall).

In our case, we have decided to convert these rules into features that could be used by the model, and we allowed the model to decide the features' importances.

For example, one common scenario is when a login is performed from a country that is different from the user's registration country. A simple rule would have to choose between flagging this event (with a given weight, or enforcement) or let it pass. In our work, we converted our legacy fraud check rules into features. In this example, the feature would be a Boolean value describing if the *"login country matches registration country"*.

5.4 Evaluation Different Models

As previously mentioned in the beginning of this section, we decided for a multi-variate Bernoulli Naive Bayes model. There are several other machine learning models that one could use to achieve similar or better results. We have experimented and compared test results of the Bernoulli NB model with others (Random Forest[7], Gradient Boosting[12] and Linear SVC[14]) in terms of *precision*, *recall* and *f1-score* in two different datasets (see Table 1 and Table 2).

Table 1: Evaluation results of different machine learning models for ATO detection in dataset 1 (56 days of data from 01/03/2019 until 26/04/2019)

Model	Precision	Recall	F1
Categorical NB	0.76	0.94	**0.84**
Random Forest	0.85	0.67	0.75
Gradient Boosting	0.86	0.67	0.75
Linear SVC	0.76	0.78	0.77

Table 2: Evaluation results of different machine learning models for ATO detection in dataset 2 (56 days of data from 05/05/2019 until 30/06/2019)

Model	Precision	Recall	F1
Categorical NB	0.62	0.86	0.72
Random Forest	0.67	0.47	0.55
Gradient Boosting	0.85	0.59	0.70
Linear SVC	0.95	0.73	0.83

The selection of which model to roll out was given by a combination of factors: *explainability*, implementation effort and performance.

Each time the model detects that a dealer account is under the control of a fraudster, this account is suspended, all listings are put offline, and a case is opened for costumer support to manually evaluate it. Given the fact that human agents are evaluating the output of the model, *explainability* of results plays an important role. We must provide a model that supports the customer support agents to understand why exactly a dealer and a listing was predicted as an ATO case. Also, the Bernoulli NB model implementation was the most convenient, to fit our currently existing infrastructure.

Finally, our performance evaluations showed that Bernoulli Naive Bayes was indeed performing better than the other models tested. The ATO detection with Bernoulli Naive Bayes model was rolled out in March 2018. Shortly after its release, our first evaluation shows (in test results) that in fact the model was performing better than others (see Table 1). However, a few months later, we noticed that the performance was decreasing (Table 2), and we attribute this to the change in behavior of fraudsters, i.e. the concept drift.

5.5 Capturing Concept Drift

In fraud fighting scenarios, one common challenge is that fraudsters change their behavior to adapt to the security checks of the system they are trying to breach. This phenomenon is known as concept drift [1, 13]. That is the main reason why rule-based approaches for fraud detection do not maintain an acceptable level of performance for long periods of time. Rule-based fraud detection can be easily reverse-engineered and learned by the fraudster through exhaustive try-fail approaches.

Machine learning models which fail to self-update (re-train) on a regular basis are also at risk. To prevent that, we train our ATO

[5] Futher analyses were left out due to space limitations and to prevent the risk of providing information on how to circumvent our detection model.

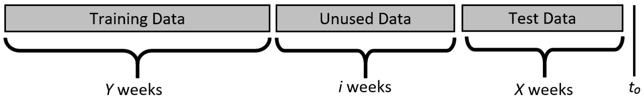


Figure 9: Concept drift sensitive model training strategy.

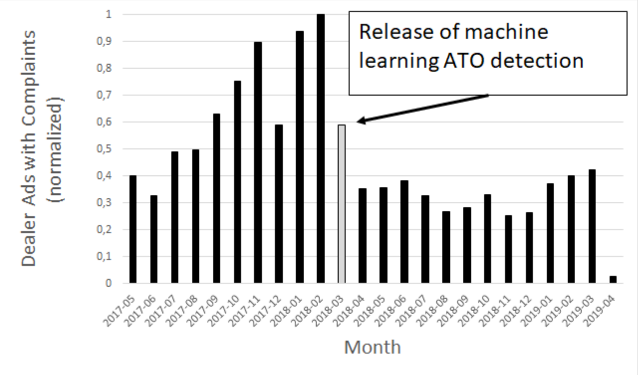


Figure 10: Normalized distribution of Bad Page Views (BPV) on dealer listings per month. In order to omit the real statistics, numbers were normalized, divided by the maximum observed number (February 2018).

detection model on a regular basis, and we developed a flexible time-sensitive sliding window approach to select training data.

The basic assumption, is that fraudsters change behavior, but it does not happen overnight. It is a slow and arduous learning process. On this basis, we assume that the last X weeks of data, represents the current behavior of the fraudsters. Given that, the training strategy is to fit the training that the best performs on these most recent weeks. Thus, if a model is trained on a given day t_0, the test data is $[(t_0 - Xweeks)\ to\ t_0]$.

The static configuration of our training setup uses Y weeks of past data before the test data, i.e. starting Y weeks from t_0, until X weeks to t_0 $[(t_0 - Yweeks)\ to\ (t_0 - Xweeks)]$

The sliding window strategy approach, iterates the training and test process several times, and in each iteration it considers a fixed amount of data that starts and ends a week before the previous iteration $[(t_0 - Yweeks - (i * 1week))\ to\ (t_0 - Xweeks - (i * 1week))]$. The number of iterations (i) varies from 0 to a arbitrarily chosen high enough value. The strategy is more easily understood on Figure 9.

With this setup, on a regular basis we train the detection model several times, and the best performing model on the latest X weeks of test data is the one used in production during the follow week.

5.6 Results

The performance of the model has been already shown in Subsection 5.4 (see Table 1 and Table 2). For the concept drift modeling strategy, comparing test results would not make sense, since the strategy deliberately fits the best training data into the test. We are able to show the resulting improvement of this strategy evaluating real detected cases. The concept drift implementation was rolled out in the end of August 2018. We compare *precision*, *recall* and *f1-score* on the 6 months before and after the implementation. We observed 18,9% improvement in *recall*, and 17,8% improvement in *precision*. Overall, *f1-score* was improved by 18,6%.

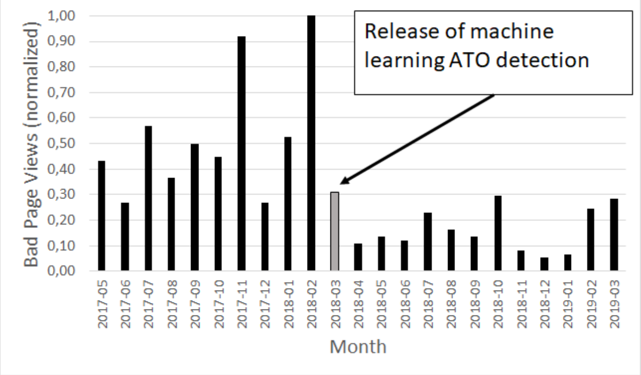


Figure 11: Normalized distribution of user complaints on dealer listings per month. In order to omit the real statistics, numbers were normalized, divided by the maximum observed number (February 2018).

In addition to that, we present the results of some of the business metrics that evaluate the detection system deployed in production. The two main metrics we track are:

- **Number of complaints on dealer ads**: At *mobile.de*, buyers have the opportunity to click a button and submit a complaint whenever they believe that the given listing is unfit or irregular.
- **Bad Page Views (BPV)**: This is the ratio of all page views on a fraudulent listing over the total amount of views.

Our machine learning fraud detection model was rolled out in March 2018. Figure 10 depicts the impact of the number of complaints, and Figure 11 shows the impact on the BPV metric.

In these metrics, the real impact of the *detection* layer can only be evaluated comparing the months before February 2018 to the two following months (March and April). The *prevention* layer strategies were rolled out in May 2018. Thus, the results after this date are a combination of both fraud fighting layers together.

With both layers combined, we can see a clear reduction on the numbers in both metrics. In fact, comparing a year over year (12 months before and 12 months after) measurement, we see a reduction in BPV of 69% and 46% in complaints.

6 DESIGN IMPLICATIONS

In this section, we summarize all our learnings in online fraud fighting situations and provide design recommendations to address the prevention and detection of fraud.

- **SMS phishing**: We have identified that phishing is the first contact point of a fraudster and a victim in the account takeover scenario. Phishing occurs through different medias of communication, and according to our customers reports, SMS-phishing (mobile phones messages) is the biggest problem. Up to date, there is technology to easily spoof SMS messages. Differently from emails, service providers do not have spam filters and on top of that, mobile phone browsers fail to provide mechanisms [4, 21] to support users to properly identify spoof and non-secure websites. With that in mind, the best way to mitigate this problem is by educating users. Users should be informed and aware that this is a common problem, and in some cases it should be made explicit that the system will not contact users via SMS, especially with a link that requires login. Previous literature suggested that users' attitudes towards security measures can be changed [22].

- **Legitimate pages getting hacked**: In this work, we also presented our findings on spoof websites. They are not exclusively unsecure (non HTTPS) websites. Although fraudsters are able to obtain SSL certificates for their spoof websites, this is not a common case. The most common case that we identified is the hacking of legitimate secure websites, modified to host spoofing pages. To mitigate this risk, enhancing the mutual verification process can support users to better identify the system they are interacting. In addition to that, spoof pages may provide referral information that can be leveraged (see Section 7).
- **Online middleman attacks**: another important point identified in our work, is the fact that malicious attacks are not exclusively automated. Fraudsters are idle online and security measurements that prevent automated attacks are not enough [8]. Given the fact that users are not always aware with which system they are interacting with (the real or the malicious one), it is imperative that the real system implements clear and unique mutual verification authentication measurements. In our case, the mutual two-factor authentication method is a simple, yet effective solution.
- **Dormant infiltrated fraudsters** : In our efforts to detect account takeovers, we have identified a common pattern in fraudulent activity. Once fraudsters take over a user account, they stay quiet, keeping the session alive until the most profitable time of day (or day of the week) to start their malicious activities. Long user sessions with unusual idle time of users should be addressed by the system. Terminating sessions during user inactivity is not enough. Fraudsters are able to simulate minor, non-malicious user activities that keep the session alive. The best solution is to compare with previous user activity to identify abnormal behavior.
- **Fraudsters' goals**: Once fraudsters takeover accounts, they have very specific goals, and given the fact they have a limited amount of time before they are flagged, they try to optimize their outcomes in a very short time. Understanding the goals of the fraudsters and analyzing fraud positive cases data is the best way to create detection mechanisms. In our case, fraudsters try to optimize leads, and in automotive marketplace scenario, the best way to do so is by reducing price, and increasing listings visibility. Thus, it is important that the system monitors abnormal activity, not only on users, but on items as well. One could argue that fraudsters could learn from this and avoid attracting users to fraudulent listing. In this case, we are very pragmatic and believe that a fraudulent listing that is not seen by users is, to some extent, harmless.
- **Social engineering**: Social engineering is the most difficult issue to address in fraud scenarios. People are the weakest link in information security systems [6], it frequently occurs out of the reach of the system, and the audacity of fraudsters is often underestimated. Depending on the system's purpose, social engineering has a specific aim that can be identified (in *mobile.de*'s case it is the pre-payment of a vehicle). Once identified, initiatives towards educating the potential victims are the only solution and if properly conducted, can be very effective.
- **Fraudster behavior change**: Detecting fraud is a constantly and rapidly changing challenge. Pre-defined rules are the most common approach to stop fraud but, are easily bypassed. If rules are too loose, fraudsters will take advantage of it, and if they are too restrictive, fraudsters can learn it and the system will eventually block legitimate users. Applying machine learning techniques is the natural evolution in fraud detection, and including concept drifting approaches is highly recommended. At *mobile.de*, we deployed a machine learning model that is automatically retrained on a regular basis, to always aim at the current and latest fraudsters behavior.
- **From rules to machine learning**: Businesses which currently rely on rule-based fraud detection methods are often concerned on abandoning their effective rules and migrating to a machine learning approach. Our solution was to transform these rules into features, that could be used by a machine learning model. Fitting these rule-based features into a model, will convey the same information the rule-based implementation had, and the model will automatically decide the importance of each rule.
- **Know your user**: Finally, the most important design implication towards effective prevention and detection of account takeover is to know your own user. That means, gather data of user activity, preferences and behaviors, analyze it, and later build and maintain user models. Having a clear picture of the user behavior is the best way to detect when something anomalous is happening, and the best way to trigger prevention mechanisms.

7 FUTURE WORK

In order to remain undetected by its victims, one common behavior of spoof websites is to redirect users to the real website after the credentials have been stolen. After the inattentive user enters his credential in the fraudulent website, most of the times he is automatically redirected to a page in the real website domain (the login page, home page or an error page). This redirection carries referral information that can be leveraged to detect potential in-risk users.

Our preliminary analysis on traffic referral shows that, 49% of the dealers who at some point reached *mobile.de* from a referral URL that contains keywords such as “mobile", “admin", “login", “TAN", became a victim of account takeover. We parsed URL referrals for keywords which have the intention to trick users into the login process of *mobile.de*.

Our future work is to incorporate this information in our detection algorithms. First, we will collect all referral URLs of dealers who had account takeovers. Then we plan to automatically extract keywords and build a dictionary assigning a risk weight for keywords in referrals URLs. Finally, the idea is to include this risk assessment as a feature in our detection model. In simpler words, this feature will tell the model if a dealer reached *mobile.de* from a spoof website. Given our preliminary analysis, we believe that such implementation will bring significant increase in our detection accuracy.

8 CONCLUSIONS

In this paper, we presented a detailed understanding of how internet fraud happens in the case of account takeovers. We thoroughly described how fraudsters operate: setting up their tools, approaching the victims, stealing the credentials, and performing the attack. We also described our solution of a mutual two-factor authentication, that has effectively reduced account takeovers at *mobile.de* by 43%. Further, we described our solutions for converting a rule-based fraud detection system into a concept drift sensitive machine learning model that supports the detection of account takeovers and reduced the number of bad page views by 69% and the number of user complains by 46%.

Thus, the contribution of this paper is multifold. We provided the readers with a comprehensive understanding of how fraudsters operate in our account takeover scenario, and we provided effective guidelines to prevent and detect account takeovers.

Despite all our efforts to fight fraud, we believe that having educated and precautious users is the best solution to avoid fraud. Unfortunately, not every user is diligent regarding his credentials, and even the cleverest users can be tricked. We are confident that the work here described will help other online services providers to implement better security measurements, by learning from our experiences, following our suggestions and applying similar solutions.

From an ethical perspective, keeping users safe is a promise that every system must uphold. And, from the business perspective, keeping users safe is a necessity for growth and value.

9 ACKNOWLEDGEMENT

We would like to thank the Customer Support team at *mobile.de* for their countless hours of manual work in detecting fraud, and for providing us the ground truth to start our work. Thanks to Leandro Cruz for leading the engineering team. Additional thanks to Florian Wilhelm, who supported us in our initial machine learning implementations. Finally, we would like to thank members of the TnS and Data teams at *mobile.de* who have directly and indirectly been involved in this work.

REFERENCES

[1] H. A. Abbass, Jaume Bacardit, Martin V. Butz, and Xavier Llorà. 2004. *Online Adaptation in Learning Classifier Systems: Stream Data Mining.* IlliGAL report 2004031. Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

[2] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. 2016. Fraud detection system: A survey. *J. Network and Computer Applications* 68 (2016), 90–113. http://dblp.uni-trier.de/db/journals/jnca/jnca68.html#AbdallahMZ16

[3] Anne Adams and Martina Angela Sasse. 1999. Users Are Not The Enemy. *Commun. ACM* 42, 12 (1999), 40–46. http://dblp.uni-trier.de/db/journals/cacm/cacm42.html#AdamsS99

[4] Chaitrali Amrutkar, Patrick Traynor, and Paul C. van Oorschot. 2015. An Empirical Evaluation of Security Indicators in Mobile Web Browsers. *IEEE Trans. Mob. Comput.* 14, 5 (2015), 889–903. http://dblp.uni-trier.de/db/journals/tmc/tmc14.html#AmrutkarTO15

[5] Adam Beautement, Martina Angela Sasse, and Mike Wonham. 2008. The compliance budget: managing security behaviour in organisations.. In *NSPW*, Matt Bishop, Christian W. Probst, Angelos D. Keromytis, and Anil Somayaji (Eds.). ACM, 47–58. http://dblp.uni-trier.de/db/conf/nspw/nspw2008.html#BeautementSW08

[6] Seymour Bosworth and Michel E Kabay. 2002. *Computer security handbook.* John Wiley & Sons.

[7] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[8] Elie Bursztein, Borbala Benko, Daniel Margolis, Tadek Pietraszek, Andy Archer, Allan Aquino, Andreas Pitsillidis, and Stefan Savage. 2014. Handcrafted Fraud and Extortion: Manual Account Hijacking in the Wild.. In *Internet Measurement Conference*, Carey Williamson, Aditya Akella, and Nina Taft (Eds.). ACM, 347–358. http://dblp.uni-trier.de/db/conf/imc/imc2014.html#BurszteinBMPAAPS14

[9] Matthew John Edwards, Claudia Peersman, and Awais Rashid. 2017. Scamming the Scammers: Towards Automatic Detection of Persuasion in Advance Fee Frauds.. In *WWW (Companion Volume)*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1291–1299. http://dblp.uni-trier.de/db/conf/www/www2017c.html#EdwardsPR17

[10] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2015. Towards Detecting Compromised Accounts on Social Networks. *CoRR* abs/1509.03531 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1509.html#EgeleSKV15

[11] Wei Fan. 2004. Systematic data selection to mine concept-drifting data streams.. In *KDD*, Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel (Eds.). ACM, 128–137. http://dblp.uni-trier.de/db/conf/kdd/kdd2004.html#Fan04

[12] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[13] João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.* 46, 4 (2014), 44:1–44:37. http://dblp.uni-trier.de/db/journals/csur/csur46.html#GamaZBPB14

[14] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. 2008. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th international conference on Machine learning*. ACM, 408–415.

[15] Robert Künnemann and Graham Steel. 2012. YubiSecure? Formal Security Analysis Results for the Yubikey and YubiHSM.. In *STM (Lecture Notes in Computer Science)*, Audun Jøsang, Pierangela Samarati, and Marinella Petrocchi (Eds.), Vol. 7783. Springer, 257–272. http://dblp.uni-trier.de/db/conf/stm/stm2012.html#KunnemannS12

[16] Claudio Marforio, Ramya Jayaram Masti, Claudio Soriente, Kari Kostiainen, and Srdjan Capkun. 2016. Evaluation of Personalized Security Indicators as an Anti-Phishing Mechanism for Smartphone Applications.. In *CHI*, Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris, and Juan Pablo Hourcade (Eds.). ACM, 540–551. http://dblp.uni-trier.de/db/conf/chi/chi2016.html#MarforioMSKC16

[17] Andrew McCallum and Kamal Nigam. 1998. A Comparison of Event Models for Naive Bayes Text Classification. In *Learning for Text Categorization: Papers from the 1998 AAAI Workshop*. 41–48. http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf

[18] Kevin D. Mitnick and William L. Simon. 2003. *The Art of Deception: Controlling the Human Element of Security.* John Wiley & Sons, Inc., New York, NY, USA.

[19] Andrea Dal Pozzolo, Olivier Caelen, Yann-Aël Le Borgne, Serge Waterschoot, and Gianluca Bontempi. 2014. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Syst. Appl.* 41, 10 (2014), 4915–4928. http://dblp.uni-trier.de/db/journals/eswa/eswa41.html#PozzoloCBWB14

[20] Xin Ruan, Zhenyu Wu, Haining Wang, and Sushil Jajodia. 2016. Profiling Online Social Behaviors for Compromised Account Detection. *IEEE Trans. Information Forensics and Security* 11, 1 (2016), 176–187. http://dblp.uni-trier.de/db/journals/tifs/tifs11.html#RuanWWJ16

[21] Ronak Shah and Kailas Patil. 2016. Evaluating effectiveness of mobile browsersecurity warnings. *ICTACT Journal on Communication Technology* 7, 3 (10 2016), 1373–1378.

[22] Richard Shay, Iulia Ion, Robert W. Reeder, and Sunny Consolvo. 2014. "My religious aunt asked why i was trying to sell her viagra": experiences with account hijacking.. In *CHI*, Matt Jones, Philippe A. Palanque, Albrecht Schmidt, and Tovi Grossman (Eds.). ACM, 2657–2666. http://dblp.uni-trier.de/db/conf/chi/chi2014.html#ShayIRC14

[23] Ding Wang, Debiao He, Ping Wang, and Chao-Hsien Chu. 2015. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment. *IEEE Trans. Dependable Sec. Comput.* 12, 4 (2015), 428–442. http://dblp.uni-trier.de/db/journals/tdsc/tdsc12.html#WangHWC15

[24] Ding Wang and Ping Wang. 2014. Offline Dictionary Attack on Password Authentication Schemes using Smart Cards. *IACR Cryptology ePrint Archive* 2014 (2014), 208. http://dblp.uni-trier.de/db/journals/iacr/iacr2014.html#WangW14

[25] Ding Wang and Ping Wang. 2018. Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound. *IEEE Trans. Dependable Sec. Comput.* 15, 4 (2018), 708–722. http://dblp.uni-trier.de/db/journals/tdsc/tdsc15.html#WangW18

[26] Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. 2003. Mining concept-drifting data streams using ensemble classifiers.. In *KDD*, Lise Getoor, Ted E. Senator, Pedro M. Domingos, and Christos Faloutsos (Eds.). ACM, 226–235. http://dblp.uni-trier.de/db/conf/kdd/kdd2003.html#WangFYH03

[27] Yongge Wang. 2012. Password Protected Smart Card and Memory Stick Authentication Against Off-line Dictionary Attacks. *CoRR* abs/1207.5497 (2012). http://dblp.uni-trier.de/db/journals/corr/corr1207.html#abs-1207-5497