Why Snoopy Loves Online Services: An Analysis of (Lack of) Privacy in Online Services

Vittoria Cozza[1], Zisis Tsiatsikas[2], Mauro Conti[3] and Georgios Kambourakis[2]
[1]*Electrical & Information Engineering Department (DEI), Polytechnic University of Bari, via Orabona, 4, 70125, Bari, Italy*
[2]*Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, Samos, 83200, Greece*
[3]*Department of Mathematics, University of Padua, Via Trieste, 63 - 35131, Padua, Italy*
vittoria.cozza@poliba.it, {tzisis, gkamb}@aegean.gr, conti@math.unipd.it

Keywords: Privacy, Offensive Security, Attack, Web-services, Flaws.

Abstract: Over the last decade online services have penetrated the market and for many of us became an integral part of our software portfolio. On the one hand online services offer flexibility in every sector of the social web, but on the other hand these pros do not come without a cost in terms of privacy. This work focuses on online services, and in particular on the possible inherent design errors which make these services an easy target for privacy invaders. We demonstrate the previous fact using a handful of real-world cases pertaining to popular online web services. More specifically, we show that despite the progress made in raising security/privacy awareness amongst all the stakeholders (developers, admins, users) and the existence of mature security/privacy standards and practices, there still exist a plethora of poor implementations that may put user's privacy at risk. We particularly concentrate on cases where a breach can happen even if the aggressor has limited knowledge about their target and/or the attack can be completed with limited resources. In this context, the main contribution of the paper at hand revolves around the demonstration of effortlessly exploiting privacy leaks existing in widely-known online services due to software development errors.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

During the last decade, online services and more specifically web-based ones, such as travel, educational, telecom and google, presented a huge outspread playing a vital role in the society's development. Current reports indicate that the market share of these services will present an additional 20% growth in terms of Compound Annual Growth Rate (CAGR) between 2016 and 2020 (Technavio, 2016). This blooming occasionally have been proved to put the end-users privacy at risk. This is because often these services present vulnerabilities which become a paradise for privacy invaders. Such weaknesses are mainly due to poor and hasty design and lack of security/privacy awareness from their developers.

Among others, the various vulnerabilities existing in online services can be exploited by invaders with the aim to excercise Open-source intelligence (OSINT) (Burattin et al., 2014) for extracting useful pieces of information regarding their victim. Conversely, consumers are gradually becoming more aware and concerned about privacy issues related to them. As highlighted in the Digital Agenda for Europe, concerns about privacy are among the most frequent reasons for people not buying goods and services online(European-Union, 2010). A similar consideration holds true for US. In fact, according to the US Consumer Confident Idx, 92% worried about their privacy (Thepaypers, 2015; Kambourakis, 2014).

This work analyses a variety of real-world examples of popular online services and reports on inefficiencies found in their authentication mechanism. We classified the discovered flaws in 4 classes according to (a) the ways a script kiddie can gain access to personal information, given the availability of an easily obtainable information related to the user (e.g. email address or phone number), and (b) and which kind of personal data are revealed if the attack is successful.

The rest of the paper is organised as follows. The next section presents the threat model that pertains to the considered scenarios. Section 3 details on the real world services attack scenarios we considered in this study. A discussion on the results is given in Sec-

Cozza, V., Tsiatsikas, Z., Conti, M. and Kambourakis, G.
Why Snoopy Loves Online Services: An Analysis of (Lack of) Privacy in Online Services.
DOI: 10.5220/0006207204310438
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 431-438
ISBN: 978-989-758-209-7
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

tion 4. Section 5 addresses related work. The last section concludes and gives pointers to future research.

2 THREAT MODEL

The work at hand builds over the fact that an adversary is able to access a significant amount of personal information about a user of an online service either if they are aware of just a single piece of data about the victim or not. By personal information we mean "any information relating to an identified or identifiable natural person (data subject)" (Council of European Union, 2016). Specifically, having in mind an attacker-centric model, we can designate two distinct types of attack: the *targeted* attack and the *random* one. According to a random attack the attacker does not obtain information in regards to a specific user, but they are able to penetrate the security apparatus with limited or moderate effort. On the other hand, a targeted attack is applicable when the perpetrator is in position to straightforwardly reveal a significant mass of personal information by obtaining a single piece of information, say, the victim's mobile phone number or their date of birth or a ticket booking code. In this respect, we can realise a two-step attack: (a) Obtain the necessary information about the victim to initiate the attack, and (b) Exploit the obtained information with the purpose of accessing the service and steal personal information about the victim.

Having the above in mind, an attacker can take advantage of web forms meant to provide a fast and easily accessible web service. This is especially true for services which for the sake of simplicity and usability allow clients to access their account without login. Instead, such web forms require as username an easily obtainable in most cases code (email, student code, booking code) plus a weak in terms of length and complexity password. While this situation renders user access easier and faster, it simultaneously exposes the user to privacy-breaching attacks. To sum up, we focus on two basic types of adversaries: internal and external ones. The main difference between them lies in the actual type and amount of information they need or already possess in order to initiate the attack. That is, an internal attacker belongs to the inner circle of the victim, say, she is the victim's colleague or friend. Obviously, an internal adversary can initiate the attack in a shorter period of time in contrast to an external one. This is because as an internal aggressor is considered honest, their environment inherently leaks more useful data regarding the victim.

3 REAL-WORLD SCENARIOS

Based on our threat model, we selected several categories of real-world applications and for each of them analysed the top 3-4 applications that were found to suffer from inherent privacy leaks. Naturally, this list is not exhaustive and the cases analysed here are only representative, meaning that there exist numerous others out there. Surprisingly, our study revealed that even large and well-known organizations that on a daily basis handle personal information of thousand of people all over the world, in travelling, telecommunication and other sectors. As explained further in this section, all these services fail to address with consistency the corresponding security/privacy standard practices (Calder and Watkins, 2010).

3.1 Classes

In the analysis of the various cases demonstrating vulnerable services, we use four different empirically derived classes, namely C1-C4:

- The first class (C1) refers to services which utilize at most 6-digit PINs.
- C2 refers to services which require public records such as email addresses and exactly 6-character long alphanumeric PINs.
- The class C3 corresponds to services which require the combination of public records (e.g., email address) in combination with a minimum 6-character long password to access the service. Opposite to C2, this class covers cases where the users are required to employ special characters in selecting their password.
- The last class (C4) refers to user login forms which require publicly-known information and are exercised through social engineering techniques.

3.2 Cases

This subsection details on the various real-world services that were pinpointed to be weak in terms of security. Furthermore, the vulnerabilities found are classified according to at least one of the four classes of section 3.1. We analyze examples from diverse sectors, including social networks, travel services, smartphone services, educational services, and Google.

3.2.1 Social Networks

We manually analysed the prevailing social networks in Europe according to Alexa, namely Twitter and

Google+. We report also a vulberability related to Facebook beta version, spotted initially by [Prakash et al.] (note however that this bug has been patched in the meantime). Google allows unified access to a plethora of services via the "My Account" web interface at https://myaccount.google.com/?hl=en. This means that after accessing it using their username and password, a user is automatically logged into all of the rest Google services, including Gmail. With this mindset, we analyse the case of breaking Gmail and then Google+ aiming to gain access to the user's Google services.

Facebook beta version: The hack in (Prakash, 2016) refers to C1 class and presents a way to access any Facebook (FB) account by exploiting a flaw in the beta version of FB website(Facebook beta, 2016). More specifically, FB's policy, specified that whenever an "i lost my password" request is received, a special 6-digit code can be sent either to the user's email address or their phone number. For initiating the attack, the aggressor can use an network interceptor like for example the Burp Suite (PortSwigger, 2016) in order to sniff the requests and responses between the user and the FB's server. The next step for the attacker would be to bruteforce the n parameter of a special string `lsd=AVoywo13&n=XXXXXX` existing in the initial http POST request. This attack is initiated using the interceptor and providing a range of 6-digit codes, which will be searched in order to find the correct one. For every one of the aforementioned codes, a request to the beta FB webpage is generated. In the case of a correct code the http response is equal to a 302 found web page, otherwise it is a 200 OK. After receiving the 302 found response, the adversary is able to set a password for any user account. This allows her to access user's personal information, including contacts, messages, photos, etc. The attack was feasible in the FB beta version because a blocking mechanism has not been activated.

Google Plus (via Gmail) Case: This case pertains to the C4 class and reports on the widely-known gmail service (Gmail - Free Storage and Email from Google, 2016). It specifically describes how an attacker could easily compromise a user's Gmail account by combining social engineering techniques with publicly available information. That is, Gmail service allows a user to send a password recovery request by providing the username related to the account of interest. The recovery process requires two steps. First off, the verification of two relatively easy to obtain pieces of data is required. The first one corresponds to the date the user created the account, while the other is related to the last time the victim logged in to their account. In the latter step, the attacker has to provide five recently contacted email addresses. Therefore, the attacker can easily bypass this countermeasure by sending the same email (in carbon copy) to 4-5 other addresses, including that of the victim.

Regarding the first step, we assume that by following a social engineering strategy one could relatively easy obtain the necessary data. For example, if the attacker and the victim work in the same office or if the victim responds in some fake email, the attack can be initiated. This happens because the aggressor would be in position to know the last time the victim logged in the service of interest. For the second step, the 5 email addresses may be falsified, created by the attacker only for this purpose. In any case, if the attacker follows the aforementioned steps, the unaware victim will have seemingly conformed to the service policy, and thus the service will allow the attacker to reset the password.

3.2.2 Travel Services

We analyzed the most popular (Monkey, 2016) accommodation companies in Europe namely Booking, TripAdvisor, Airbnb, and (low-cost flight companies) RyanAir, Eurowings, and Easyjet.

TripAdvisor, and Airbnb require the user to login by email and a 6-char long password, which combines digits, letters, and special characters. Also, they require email confirmation after password lost, therefore, in terms of login policy, these services are considered generally safe. On the downside, as detailed next, we were able to discover some vulnerabilities in Booking, Ryanair, Eurowings, and Easyjet web forms.

Booking: This case pertains to the far farmed Booking company (Booking, 2016) and can be classified under C2 class. This company provides an online service for booking an accomodation based on user ratings, comments and price comparison. It also offers an easy way to manage an already existing booking to a user, without requiring full website registration and authentication, but by simply accessing the "make change to your booking online" option. The service requires a 9-digit booking code and a 4-digit PIN to access the account. If the attacker knows the booking code, for example, using Social Engineering, and bruteforces the PIN, then they can access the booking information.

RyanAir: This scenario belongs to the C2 class and revolves around the well-known and currently the biggest in Europe in terms of budget, low-cost flight company Ryanair. In the following, we describe this attack scenario by following a two-step approach, and we particularly concentrate on two kinds of vulnera-

bilities that could be exploited by a potential aggressor:

1. Step 1 - If the attacker knows the booking details (flight date, origin, destination) for a future flight, as well as the email address used by the victim for accessing Ryanair they can retrieve a victim's valid booking reference. While this information may not be directly useful to the attacker, it serves as a stepping stone for step 2.
2. Step 2- Once the attacker obtains a valid booking reference and the victim's email address, they are able to retrieve all the victim's booking history.

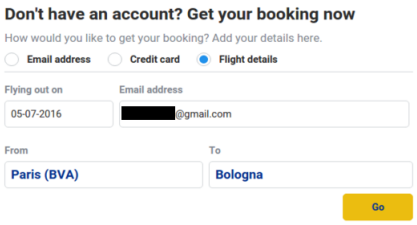

Figure 1: Form to retrieve the booking reference for a future flight.

As given in Figure 1, for Step 1 above, assuming the attacker knows that a certain person is going to fly with Ryanair to a given destination on a particular date, they can insert this information (flight date, email, origin, destination) in the web form available at https://www.ryanair.com/us/en/check-in. As observed from the figure, this form has been designed to ease the passenger to retrieve their next booking information and eventually proceed to check-in, but it does not require any password or other authentication method. That is, by only inserting the aforementioned information, the user as well the potential attacker will have in return the code of the victim's next flight. It is to be noted that all the aforementioned pieces of data may be known to people that are connected to the victim, including their colleagues. In any case, Ryanair is well-known to sell extremely discounted flights nearly every week and for selected destinations. This means that it is almost always feasible to book a flight for less than 10 Euro. This in turn means that an assailant would be able to book (and even with a good deal) a flight for its target, only with the purpose to access a valid booking reference to be exploited in the next step of the attack.

Regarding the second step of the attack, as shown in Figure 2, in order for someone to access the victim's booking history, they have to connect (but not login) to the corporate website, and just select the "Manage my Booking Webpage" option (Ryanair: Manage Your Booking, 2016). Next, by choosing "View Booking History" and inserting one valid booking reference and the passenger's email, one can straightforwardly obtain access to their personal information. Specifically, as observed from the Figure 2, this attack reveals the full booking history, including details of any future flight bookings of the victim. Actually, as it is depicted in Figure 2, there is no option in the menu for the client (passenger) to delete these historical booking information, and we realized that these log files are kept for at least the current and the previous year. Once again, a valid booking reference can be retrieved as in Step 1, or by exercising a bruteforce attack, or even by performing shoulder surfing. It is also important to note that while the first step allows for obtaining a booking flight reference pertaining to a future flight, the second step can be exercised having either a past or future flight booking reference. Summarizing, the most important finding in this case is that virtually everyone is able to access personal information about the victim by possessing two quite effortlessly obtained information about them. The results obtained from such a search query are easily observable in Figure 2 that zooms out the details of a single booking among those returned in our example. Also, keep in mind that Figure 2 presents two distinct names which correspond to different persons. Precisely, the first name (Vittoria) corresponds to the owner of the account, that is, the person who possesses valid credentials for this account, while the second name (Ahmad) is the person on behalf of which the account owner booked a flight and perhaps she travelled with. Therefore, regardless of the individual for whom the flight has been booked, this flaw can be exploited to reveal information for both individuals. All the aforementioned attacks are clearly feasible since Ryanair clients are not required to login before booking their flight or access their his-

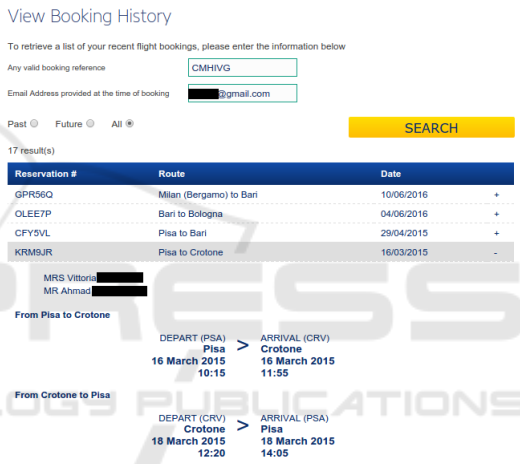

Figure 2: Travel history of a passenger by only entering a valid flight booking reference and the passenger's email.

torical data. Starting from Sept. 7 2016 Ryanair is finally encouraging its clients to create an account on the website, e.g. by offering them a 10 Euro voucher on their next flight as well as by advertising a personalized flight experience for those who create a user profile.

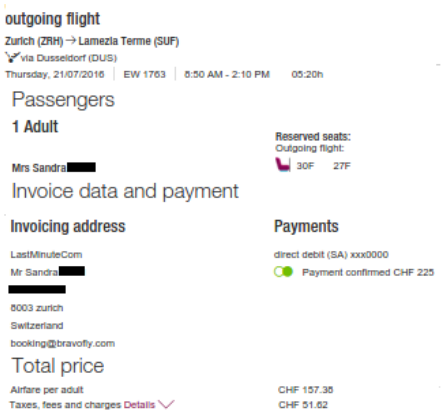

Figure 3: Retrieved invoice data for a Eurowings flight.

Eurowings: This case applies to the low-cost flight company Eurowings and can be classified under C2 class. More specifically, excluding the typical login option via the use of username and password, Eurowings allows a traveller to check their flight details by providing a booking code (6-char) and their surname. The vulnerable form is accessible at https://www.eurowings.com/skysales/MyDashboard.aspx?culture=en-GB. Besides the flight details, once the traveler expands link button "Booking overview: Here you will find all details about your booking", they obtain also the invoice data and payment that includes the victim's physical address as in Figure 3. Therefore, if the attacker knows the person, the only piece of data they need to additionally acquire is a valid flight code. This can be done by, say, dumpster diving and enables the attacker to gain access to personal data related to the victim. A similar case has been found in the widely-known Easyjet airline company.

3.2.3 Mobile Phone Related Services

In order to survey mobile phone related services, we selected a well-known Canada company. We also analyzed services that are based on the availability of a mobile phone number, specially parking services: PaybyPhone, and Phone and Pay because these are popular in UK. In the following, we detail on cases that found to be vulnerable.

WIND mobile Canada: This attack scenario refers to WIND Canada (Wind: Login to My Account, 2016) mobile communication company and belongs to the C1 class. The majority of user login pages out there support the username/password login method. However, as observed from the corresponding login form, this page provides the user with two options. The first one is the common way in which a user provides a username and a password, whereas the second asks the user to insert their phone number and a PIN. The service demands a 4-digit PIN which can be bruteforced easily. According to (Brute Force Calculator, 2016), such a PIN requires 11110 password combinations or 1 sec to brute force it. If the attack is successful, the assailant can access sensitive information related to the victim's call history, including information about the date, the number, the type of the call, the amount charged, and finally the remaining balance.

PaybyPhone: This case belongs to C1 class and focuses on a worldwide PayByPhone company (Paybyphone: Manage your account, 2016). The aforementioned company offers a service meant to be used for parking fee payment in a fast and easy way. To login to their account the customer needs to provide their mobile phone number and a 4-digit PIN. Nevertheless, a person's mobile phone number is known in their social circle or exposed in publicly available documents, including telephone catalogues, CVs, etc. In this respect, this piece of information is not secret. On the other hand, as with the previous case, the 4-digit PIN can be easily bruteforced or obtained with legacy social engineering techniques. The result is that the assailant can access the legitimate user's account and view their personal information related to vehicle details and payment info. This way, location information, regarding the victim, is exposed as well. A similar vulnerability to the PaybyPhone case has been spotted in a Phone and Pay company located in the UK (Phoneandpay: Customer Account Login, 2016).

3.2.4 Educational Services

For this category we analyzed services like the ORFEAS system operated by the Hellenic American Union.

ORFEAS: This scenario pertains to C2 class and refers to ORFEAS online service (Orfeas: Student Exam Results, 2016), which is used by English language tutors with the aim of viewing examinees results regarding the Hellenic American Union exams. This service requires two codes for accessing current and past examinees results. The first code needed is the school code which can be obtained exercising common social engineering techniques. For example, a person working in such an institute is in position

Table 1: Analysis of Cases (PR = Public Record, PN = Phone Number, SC= School Code, BF=Brute Force, BC=Booking Code).

Class	Required information			Reveals Personal Data	Effort to bruteforce			Online Service
	Data	How to obtain	Effort (in terms of BF)		Combs.	Block. Mech.	Entropy (bits)	
C1	PN 4-digit PIN	Social Eng.	Trivial	Long-term	10K	No	5.8	Phone and Pay, PaybyPhone, Canada Wind
C1	6-digit PIN	BF	Trivial	Long-term	1M	No	9.7	Facebook beta version
C2	Email 6-char BC	PR Social Eng.	Easy	Long-term	2G	No	20.1	Ryanair, Eurowings, Easyjet
C2	Email 6-char BC	PR Social Eng.	Easy	Short-term	2G	No	20.1	Booking
C2	SC 6-char PIN	Social Eng. BF	Easy	Short-term	2G	No	20	Orfeas
C3	PN 6-char Pass	Social Eng.	Hard	Long-term	735G	Yes	26.3	Twitter, TripAdvisor, Airbnb
C4	Emails Date	PR Social Eng.	Hardt	Long-term	6P	Yes	30.3	Google Services via Gmail

to access the code and leak it to the adversary. Also, several times the teachers write down these codes on scrap of papers and leave them in plain sight, say, on their desk. The other code is a 6-digit password which combines lower-case letters and numbers, but not special characters.

4 ANALYSIS OF CASES

We consider a qualitative evaluation of three main factors which characterize the attack and have to do with reward vs. effort from an attacker's viewpoint. These three factors express i) the type and the amount of information the adversary would need in order to initiate the attack, ii) the type and amount of effort needed to run the attack, and iii) the type and amount of personal information gained after exercising the attack, that is, if it is fast outdated or of longterm validity. From the real case scenarios presented in the previous section, which all have in common at least one easily obtained information, we identify five basic attack classes summarized in Table 3.2.3.

Generally, a bruteforce is feasible when one deals with short-length homogenous credentials. To calculate the duration of such a process, one can access an online service as that in (Brute Force Calculator, 2016). For example, as already pointed out, to bruteforce a 4-digit PIN, in a numeric space, employing a BSDi DES-based ciphertext format, one would need 11,110 password combinations. In the absence of an additional defensive mechanism, including CAPTCHA (Conti et al., 2016) or IP address blocking, a key space of this size can be searched instantly, and thus the password strength is practically very low.

In many cases, online services employ obsolete security standards or policies in authentication mechanisms with the purpose of protecting users' privacy. Moreover, despite Best Current Practices (BCP) and other guidelines do exist for the multi-factor authentication (MFA)(NIST, 2016), as stressed by this work still several popular services out there do not employ them. For this reason, these standards should regularly be reviewed in order to keep up with the current computational growth. Very popular services as Facebook, Amazon and Fidelity still allows the minimum length of a newly created password equal to six characters. Nevertheless, they do have protection against online attacks; a slow-down answer or a fail-ban mechanism that will reduce the effectiveness of the attack. This tactic is generally adopted as a means to balance between security and usability (Florêncio et al., 2014).

To sum up, all flaws shown in the various cases of Section 3.2 could have been avoided by following the privacy by design strategy (PbD) (Privacy By Design, 2016). However, it seems that the designers of these services focused mostly on usability rather taking into account privacy from their design state. PbD enables engineers to take into account privacy issues from the beginning of the development cycle, hence developing systems which inherently protect privacy. Furthermore, online services should regularly update their security mechanisms in order to follow proportionally the growth of computer systems (Moore's law). Also, this should be done for the dated services with the aim to regularly update their security level by including mobile phone verification, geolocation, (Hang et al., 2015) and captcha (Conti et al., 2016). All these mechanisms are used in order to avoid bruteforce attacks.

5 RELATED WORK

The work done so far in demonstrating and remediating security and privacy flaws in online web-based services is rather large. So, in this section, we only address related work that are related to the following issues contained in the OWASP Top Ten (Mark, 2016) and pertain to authentication mechanisms in general.

The authors in (Liginlal et al., 2009) present a solution strategy to avoid human errors aiming to eliminate privacy leaks. More specifically, this solution is based on the application of three different strategies, namely error avoidance, error interception and error correlation. Moreover, the work in (Kraemer et al., 2009) tries to investigate how human and organizational factors affect computer vulnerabilities by applying a macroergonomic approach in order to identify the bonds between human and organizational factors. Furthermore, the authors in (Barlow et al., 2013) rely on the factorial survey method aiming to train employees to avoid security policy violations.

In(Gejibo et al., 2012), the authors deal with the usage of mobile phones, PDAs and other mobile communication devices for managing sensitive data related to e-health in developing countries. Mechanisms for secure storage and end-to-end encryption are introduced as an extension of the openXdata standard (Switch to openXdata, 2016). As highlighed also in the current work, the need of a more fast and easy to use system, that could be used by, say, a mobile device, has as a counter-effect the implementation of a less secure system. In (Ohata et al., 2016) the authors consider a provably secure password reset protocol and show its efficiency via a prototype implementation. Several online services could take advantage of such a protocol. Indeed, the services that adopt a password-based authentication usually support a mechanism with which a user can reset their password, but this mechanism can be unsecure, as analyzed in Section 3. The authors, in (Xie et al., 2011), concentrate on software development errors. They conduct a qualitative research based on interviews with the aim to investigate the reasons that affect developers and make them prone to development errors. They state that their results indicate lack of relation between the developers awareness and their behaviour. As about 24% percent of the top 10 million web sites are built upon the content management system WordPress, it comes as no surprise that content management systems in general and WordPress in particular are frequently targeted. By using manual and statistic analysis, the authors in (Trunde and Weippl, 2015) elaborate on possible ways of detecting common programming errors. Their analysis has been conducted starting from public available security exploits. In (Tiwari, 2014) the author shows simple ways to add Security to Web Development, highlighing poor designed programming language features that favor SQL injections. Also, he discusses ways that such errors can be avoided. Similarly to our work, in (Hang et al., 2015) the authors evaluated the strength and the weaknesses of a widely used autentication mechanism, namely the security questions. As a plus, the authors propose a location-based authentication with security questions as a more usable and secure fallback authentication scheme.

6 CONCLUSIONS

The paper at hand presents a handful of real-world web services which present striking flaws that render them easy targets to even naive or script kiddies kind of attackers. We categorize these cases in four distinct classes based on the required effort to launch the attack and gain access to personal information. In a nutshell, the crux of our work lies on the fact that several web services used by a plethora of people on everyday basis still suffer from basic both intentionally and unintentionally introduced flaws, which in turn reveal that either the situational awareness in cyber security and privacy is low or the designers continue to consider the usability aspect above anything else, including security. This also strongly suggests that holistic approaches like the PbD one should be given more attention by all the stakeholders in an effort to end up to services that present a drastically reduced attack surface.

ACKNOWLEDGEMENTS

Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). This work is also partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061), the EU-India REACH Project (agreement ICI+/2014/342-896), the Italian MIUR-PRIN TENACE Project (agreement 20103P34XC), and by the projects “Tackling Mobile Malware with Innovative Machine Learning Techniques”, ”Physical-Layer Security for Wireless Communication”, and “Content Centric Networking: Security and Privacy Issues” funded by the University of Padua.

REFERENCES

Barlow, J. B., Warkentin, M., Ormond, D., and Dennis, A. R. (2013). Don't make excuses! discouraging neutralization to reduce it policy violation. *Comp. Sec.*, pages 145–159.

Booking (Accessed Feb 3, 2016). http://www.booking.com/.

Brute Force Calculator (Accessed Feb 3, 2016). calc.opensecurityresearch.com.

Burattin, A., Cascavilla, G., and Conti, M. (2014). Socialspy: Browsing (supposedly) hidden information in online social networks. In Lopez, J., Ray, I., and Crispo, B., editors, *CRiSIS 2014*, pages 83–99, Cham. Springer International Publishing.

Calder, A. and Watkins, S. G. (2010). *Information Security Risk Management for ISO27001 / ISO27002*. It Governance Ltd.

Conti, M., Guarisco, C., and Spolaor, R. (2016). Captchastar! a novel captcha based on interactive shape discovery. In *ACNS*, pages 611–628.

Council of European Union (2016). EU General Data Protection Regulation 679/2016. goo.gl/Klj2kL.

European-Union (2010). A digital agenda for europe; brussels, 26.8.2010 com(2010) 245 final/2 (41 pages). goo.gl/acfDjs.

Facebook beta (Accessed Feb 3, 2016). beta.facebook.com.

Florêncio, D., Herley, C., and Van Oorschot, P. C. (2014). An administrator's guide to internet password research. LISA'14, pages 35–52.

Gejibo, S., Mancini, F., Mughal, K. A., Valvik, R., and Klungsyr, J. (2012). Secure data storage for mobile data collection systems. In *MEDES*, pages 131–144. ACM.

Gmail - Free Storage and Email from Google (Accessed Feb 3, 2016). https://mail.google.com/mail/u/0/#inbox.

Hang, A., Luca, A. D., Smith, M., Richter, M., and Hussmann, H. (2015). Where have you been? using location-based security questions for fallback authentication. In *SOUPS 2015*, pages 169–183.

Kambourakis, G. (2014). Anonymity and closely related terms in the cyberspace: An analysis by example. *JISA*, pages 2–17.

Kraemer, S., Carayon, P., and Clem, J. (2009). Human and organizational factors in computer and information security: Pathways to vulnerabilities. *Comp.Sec.*, pages 509–520.

Liginlal, D., Sim, I., and Khansa, L. (2009). How significant is human error as a cause of privacy breaches? an empirical study and a framework for error management. *Comp.Sec.*, pages 215–228.

Mark, C. (Accessed April 14, 2016). Owasp top ten project. goo.gl/MyTG7S.

Monkey, S. (2016). Most popular travel apps: Tripadvisor, expedia, and airbnb lead the pack. https://goo.gl/T9Xa37.

NIST (2016). Digital authentication guidelines. goo.gl/QL8O95.

Ohata, S., Matsuda, T., and Matsuura, K. (2016). Provably secure password reset protocol: Model, definition, and generic construction. *IACR Cryptology ePrint Archive*, page 345.

Orfeas: Student Exam Results (Accessed Feb 3, 2016). http://results.hau.gr/.

Paybyphone: Manage your account (Accessed Feb 3, 2016). https://www.paybyphone.com/account.

Phoneandpay: Customer Account Login (Accessed Feb 3, 2016). https://www.phoneandpay.co.uk/login.asp.

PortSwigger (Accessed Feb 3, 2016). Burp suite the leading toolkit for web application security testing. https://portswigger.net/burp/.

Prakash, A. (Accessed Feb 3, 2016). How i could have hacked all facebook accounts. goo.gl/Q3dnsH.

Privacy By Design (Accessed Feb 3, 2016). goo.gl/KTyzuw.

Ryanair: Manage Your Booking (Accessed Feb 3, 2016). goo.gl/cKGhi2.

Switch to openXdata (Accessed Feb 3, 2016). http://www.openxdata.org.

Technavio (2016). Report: Global it spending by online service and application market to reach $23 billion by 2020. http://goo.gl/vs4Bsi.

Thepaypers (2015). Us internet users still concerned about data privacy. goo.gl/5XCglf.

Tiwari, N. (2014). Simple ways to add security to web development. *Linux J.*, 2014(238).

Trunde, H. and Weippl, E. (2015). Wordpress security: An analysis based on publicly available exploits. In *17th IIWAS*, pages 81:1–81:7. ACM.

Wind: Login to My Account (Accessed Feb 3, 2016). goo.gl/W695oW.

Xie, J., Lipford, H., and Chu, B. (2011). Why do programmers make security errors? In *VL/HCC*, pages 161–164.