Dissuading Stolen Password Reuse

Slim Trabelsi[1(✉)] and Chedy Missaoui[2(✉)]

[1] SAP Security Research, 805, Avenue Dr. M. Donat, Mougins, France
slim.trabelsi@sap.com
[2] Tessan Group, Rue des Jardins, Tunis, Tunisia
chedy.missaoui@gmail.com

Abstract. The whole security community agreed on the fact that login and password based authentication systems are one of the weakest point of the current systems. Despite this global consensus password based credentials are still the most used identification and authentication method used on internet. One of the main reason for this weakness is due to the password leak phenomena. For several reasons (described in this paper) password databases are frequently leaked and shared publicly. Once these passwords it will be very hard for a user to protect his digital life, especially if this password is used in several websites (what we call domino effect). In this paper we propose a solution to reduce the attempts for replaying stolen passwords. We measure the efficiency of this solution via a deployment and the analysis on a fake website exposed to a fake password leak.

Keywords: Passwords · Leakage · Hacking · Cyber security · Authentication

1 Introduction

Bruce Schneider said: "As insecure as passwords generally are, they're not going away anytime soon. Every year you have more and more passwords to deal with, and every year they get easier and easier to break.". Despite that fact, password authentication systems are still dominating the authentication landscape especially on internet websites (less inside big companies where certificate-based authentication is becoming more and more popular [10]). Many technological and cultural reasons are explaining this phenomenon [11] and this issue will stay for several years in the future. According to The Breach Level Index [1] every day more than 5 million records are stolen and only 4% are encrypted. The rest is in clear text or hashed and finally easily accessible to cyber criminals. A Large portion of this data is composed of credentials and all the content is at some point of time published for free on internet. One of the recent biggest clear text credentials disclosure was recently released [2], with more than 1.4 Billion entries compiled from several leaks. Almost all the big internet companies suffered at some point of time from a credential leak (Apple, Amazon [3], LinkedIn [4], Twitter [5], Microsoft [6]), and according to a recent study [7] 65% of data beaches result from weak or stolen passwords. And without being paranoid we are almost all concerned by a password leak at some point of time in our digital life. In some of the cases we are not even aware about the theft.

A. Saracino and P. Mori (Eds.): ETAA 2018, LNCS 11263, pp. 116–128, 2018.
https://doi.org/10.1007/978-3-030-04372-8_10

One password leak could have much more disease that expected, and this is due to the password replication custom from certain people to use the same password for many domains or a derivation a root password easy to guess. For example, if your Gmail account was leaked, a malicious user would try to replay it for Hotmail, Yahoo, LinkedIn, Facebook, Twitter or even your professional e-mail account. If the password is the same everywhere the whole digital life of a person can be ruined. This phenomenon is called domino effect [8]. The Mozilla bug tracker (Bugzilla) was severely hacked in 2014 due to a domino effect affecting one of their administrator who was using the same password Bugzilla management and his twitter account [9]. His Twitter password was leaked, and the hackers replay it on Mozilla. The result of this hack was the full access to all security notes including zero-day vulnerabilities, exploit code related to all Mozilla software. All the security experts were recommending to not use Firefox until all the security breaches are fixed. The domino effect is not only concerning basic password reuse, but it concerns password reshape. Due to the human limitation of memorizing various combinations of passwords related to all their online accounts, one option is to create passwords starting from a common root word. Like example a root password is *ILikefootball* and the derivations will be {*ILikefootball1234, ILikefootball&"'$$, footballILike9871*, etc.}. Some algorithms [12] can guess those types of variations and make the domino effect much more harmful.

For this reason, in this paper we propose a solution to try to discourage hackers to reuse stolen passwords to compromise users accounts. We measure the impact of this solution by deploying a fake banking website and leaking fake credentials. We observe then the reaction of the hackers when the dissuasion process is triggered.

This paper is organized as follows: in Sect. 2 we describe the different reasons and factors that leads to a password leak. In Sect. 3 we list the different channels used to spread stolen credentials. In Sect. 4 we describe our honeypot case study. In Sect. 5 we introduce our solution to dissuade the stolen password reuse, and we evaluate the impact of this solution on our honeypot. In Sect. 6 we declare our ethical considerations applied to conduct this study. In Sect. 7 we describe our state of the art study, then we conclude.

2 How Credentials Are Leaked

There is a multitude of reasons at the origin of a password leak. In this section, we give a non-exhaustive list of methods and attacks used by attackers to obtain credentials from websites, systems and people.

2.1 Vulnerability Exploit

The software vulnerability is defined as a weakness of a failure existing in the source code of the system that can be exploited by an attacker to perform malicious actions. Exploiting such vulnerability can require writing a code or execute a work-flow process in a different way from what it was initially designed. An SQL injection attack is for example exploiting a bad input validation vulnerability and can lead to the entire database dump including the password tables.

To exploit vulnerabilities cyber criminals, had the good idea to make script kiddie's life easier by developing easy to use and automated tools called exploit kits. These tools will target a system make an analysis, identify all the potential vulnerabilities and execute the related exploit attack. This kind of tools contributed to the democratization [13] of micro-bloggings attacks and resulted of many data breaches, including credential dumps.

2.2 Social Engineering and Phishing

The social engineering attacks, is based on the exploitation of human trust to extract confidential information from a victim. It is based on a psychological manipulation that masquerades an entity of trust to the victim in order to ask for personal or confidential information. One of the most known method of this attack in information security is called Phishing attack. A phishing attack is mainly spread though e-mails, it takes the appearance of a professional or serious e-mail (management, bank, support team, etc.) but it redirects to a pitfall. For a massive Phishing attacks, Phishing kits are available to automate the fake e-mail distribution, the deployment of trap servers and the collection of credentials.

2.3 Keyloggers and Malwares

Some malwares are exploiting vulnerabilities of the systems to access their databases or file set, some others install keyloggers to capture all the keyboard entries of the victim. Credentials are then collected and sent through the network to the attacker servers. Even if most of the antiviruses can detect traditional keyloggers, some malicious browsers plugins remain undetected and continue to steal keyboard typing. Other types of malware are used to intercept system and configuration files to identify credentials.

2.4 Easy to Guess and Default Passwords

In all the best practice recommendations related to the password setup, the rule number one is to not chose an easy password. This rule is elementary event if some persons are still ignoring it. This issue becomes really dramatic when system administrators are committing the error in wide scale. We can refer to a practice that was spread among hardware vendors to set default passwords[1] for systems (usually the same one). Big industrial companies were targeted by attackers exploiting[2] the default password vulnerability. Or in some other cases system administrators chose to use personal identifiers of the users to create passwords like birthdate or social security numbers, etc. This would open the floor to easy guessing attacks like the Yale vs Princeton case.

[1] https://www.scmagazine.com/russian-researchers-leak-default-passwords-packaged-to-icsscada-software/article/527829/.

[2] https://www.nytimes.com/2002/07/26/nyregion/princeton-pries-into-web-site-for-yale-applicants.html.

2.5 Honeypots and Traps

Cyber criminals are permanently inventing new strategies to collect people credentials, some of them are elaborated and require a long-term effort. In some cases, they create real websites and services like discussion forums, adult websites, storage platforms or free virtual machines. These platforms are of course collecting all the credentials created by their users and rely on the domino effect [8] to compromise other accounts from their users. Even if some studies pointed out this phenomenon [14] very few statistics are available to quantify the impact of such sophisticated attacks.

3 Where Credentials Are Published

There are several sources sharing stolen credentials. Depending on the freshness and the quality of the data, these sources can be paying or free.

3.1 Commercial Sources

One of the main motivation to leak data and more specifically credential is the financial gain that could be generated from this action. We observe frequently cyber criminals selling credentials on the black markets in the dark web marketplaces. The prices and the popularity can vary with the freshness and the sensitivity of the data sold. In 2016[3] for example a hacker was selling a bunch of US government credentials in the dark web for very high prices. In this case the credentials sold are very sensitive, rare and fresh. Then a chain of resellers will appear in order to invest in this kind of merchandize and create mini-websites to sell the credentials per entry or per package of 10. This kind of stolen credential will be cascaded through several sub-sources until becoming free at some point of time. There is a real illegal business in the password resell. Without being a talented hacker, a simple reseller can generate a lot of money just by collecting and reselling credentials. A lot of people were arrested[4] for running such kind of credential reselling business.

3.2 Free Sources

In the previous section, we exhaustively described the leaked password lifecycle in the illegal commercial circuit that ends-up in to a free sharing platform. According to most of the recent studies, text sharing websites like PasteBin are the most commonly used platforms to share free stolen credential or to advertise on sales by sharing part of the stolen databases. Hacking forums like hackforums.net, offensivecommunity.net, or bestblackhatforums.eu, are also popular places to share this kind of data, even if the access is restricted (needs account creation and works with a credit compensation

[3] http://www.businessinsider.fr/us/hacker-selling-credentials-government-sites-2016-7.

[4] https://thehackernews.com/2018/01/leakedsource-operator-charged.html.

system based on the contribution). Some torrent hosts are also used to share huge databases. These sources are easily accessible by most of the users on internet and offers a huge collection of stolen passwords that is maintained and enriched over the time.

Some legal websites are also offering the possibility to check whether their credentials were leaked at some point of time. Websites like have I been pwned[5] gives the possibility to provide your login or password and find how many times they were leaked. They also offer commercial services to sell the data per domain or to alert when a credential is leaked. This kind of websites are collecting the publicly available leaked databases. Some discussions are still ongoing on the morality of making legal business by offering services based on stolen passwords.

4 Dissuading Stolen Password Reuse

In this paper, we propose a new solution to deter and prevent malicious people from reusing stolen or hacked credentials to illegally access users accounts. We put in place a system that will threaten the authors of this illegal access tentative exploiting the stolen credentials.

4.1 Concept

When a website or a domain is hacked, the administrator is at some point of time notified about the issue. The administrator of hacked website will then put in place a password change process to all their users by notifying them and asking to change password (ideally using two factor authentication). Once this step done, the administrator will observe the account updates from the user. Once a login tentative using the old stolen credential is detected, the attacker will be redirected to a honeypot version of the website. He will be invited to perform an account 'recovery' process (that seem very legitimate to the attacker). At this step the same system used in our honeypot can be used by the domain host.

During this process (described in Fig. 2), the attacker will be asked to provide information to recover the blocked user account such as: email address, (second recovery email address), phone number and a new password. He will be then asked to confirm all this information by sending a verification email, an SMS code or a phone call. At the same time, all the attacker's navigation metadata will be collected: IP address, browser fingerprint (user agent, list of installed plugins, language, screen size, Operating system etc. …), VPN provider and address if used, Internet Service Provider, IP Geolocation. We also inject tracking cookies and we create a virtual profile of the attacker. We might also try to scan his IP address to detect open ports, and detect running applications.

[5] https://haveibeenpwned.com/.

When the attacker reaches the state “Warning” in Fig. 3, the honeypot will display a warning message containing all his data and explaining that he is in a law infringement that could lead him to court judgement (see the warning message displayed Fig. 1). The machine signature is then blacklisted in order to block any other tentative.

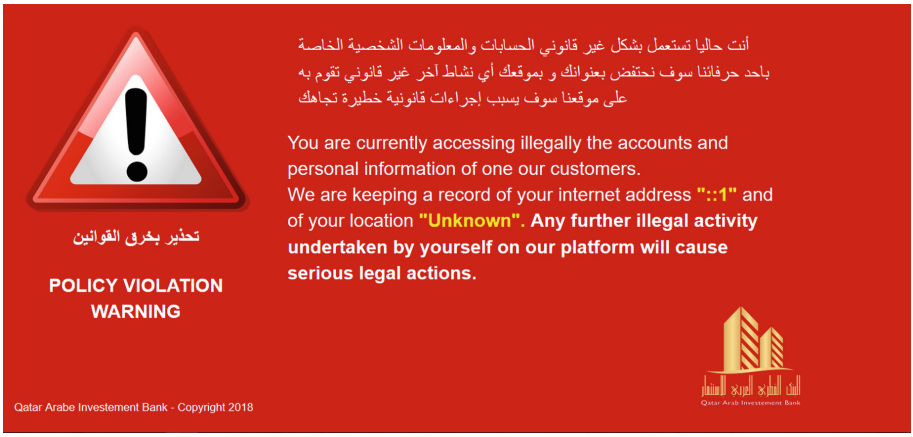

Fig. 1. Violation warning message

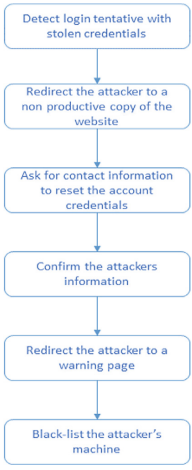

Fig. 2. Dissuading password reuse process

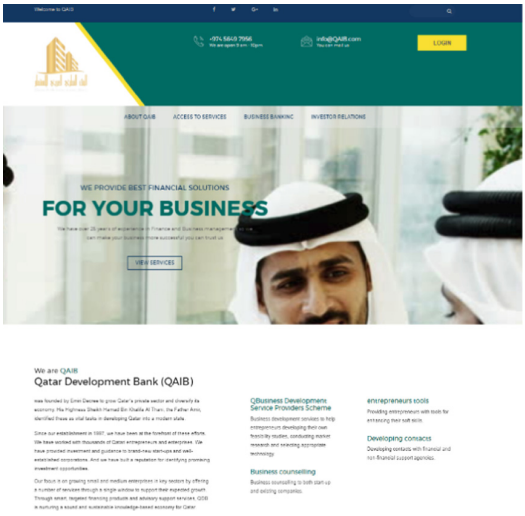

Fig. 3. Honeypot website capture

5 Experiments, Measurements and Validation

The goal of this study is to try to identify the profile of the persons that are illegally re-using leaked passwords shared on internet. We try to capture their behaviour and their anonymity degree. We also propose a counter-measure to reduce the re-usage motivation of the attacker.

5.1 Honeypot Bank Website

We decided to create a fake website of middle eastern bank. We also generated fake credentials dataset (containing Arabic names as logins). We choose Middle Est due to the convergence of several studies identifying their banks as the most targeted ones by the various attackers.

We generated 3300 credentials distributed over 10 well known websites for credentials sharing on the surface web and the dark web. Here are the links to the sites:

- https://pastebin.com
- https://www.pastefs.com
- https://slexy.org/recent
- http://n0z.de/index.php
- https://pastie.ru
- https://justpaste.it
- https://pastelink.net/read
- https://ideone.com/recent
- http://nzxj65x32vh2fkhk.onion (Stronghold)
- http://depastedihrn3jtw.onion

We started the experience on March 2nd 2018 and we recorded for a duration of three weeks. We made 11 rounds of distribution to these sites (until March 11th), to ensure a good visibility. For every site, we publish a specific set of credentials to easily identify the site origin of the interaction.

Architecture
The Honeypot system was deployed on a cloud hosting service with a decoupled system backup to save data in case of attack (see Fig. 4).

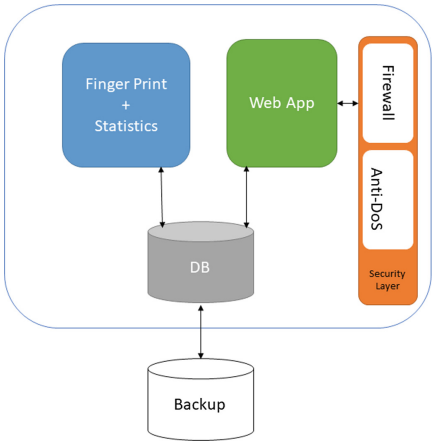

Fig. 4. Honeypot architecture

The Web application is exposing the web interface (Fig. 3) and implementing the workflow of the user interaction including the deception system put in place (will be described further in this paper). As mentioned previously, in order to protect our system and the data collected we put in place a firewall system and an anti-DoS attack framework. This security system is intercepting all the incoming requests to the web app. All the interactions and the credentials are persisted in a local DB that is also connected to the fingerprint and the analysis engine. This component is charge of collecting the navigation information and the traces left by the users while they are accessing the website. The statistics engine is in charge of the analysis and the computing of all the events and the interactions happening in the system in order to facilitate our study.

Fingerprint
A browser fingerprint is the combination of several identification parameters that will make the browser uniquely identifiable. The browser fingerprint can be quantified into a signature calculated by the combination of numerical values associated to the different parameters. In our study we choose of the following parameters to compute the signature: Browser Type, Browser version, Browser name, Operating system, Is Beta, Is Crawler, is Win16, Is Win32, Supports frames, Supports tables, Supports Cookies,

Supports VB Scripts, Supports JavaScript, JavaScript version, Supports Java Applets, Supports ActiveX Controls, User IP Address, User Host name, Remote port, Country, Language, Plugins, Time-zone.

All these elements combined will generate a signature used to identify distinct users running traditional browsers. In order to compute this signature, we create a matrix with all these parameters and for every new entry we increment a numerical value. The union of these values will generate a signature vector.

State Machine

When a user accesses the honeypot website, he has the possibility to execute certain actions in the bank account. Every action is part of a global workflow that we depicted in Fig. 5. This state machine model will be used to make statistics on the behavior of every user visiting the website.

When a user visits the bank homepage page we do not record any trace (not useful). When the user logs-in with a stolen credential then the tracking starts. Once the user is logged in, an information message invites him to update the account password and the contact information. If the attacker decides to update the account information, he will have the choice to update the password, the e-mail address or the phone number. In case of e-mail address update, a conformation from the mail box is needed to check the validity of the address. There is no phone number verification. Once this information updated the attacker must login again. Now the "manage bank account" button appears in the interface, if the attacker click on this button a warning message is displayed (Fig. 1). This warning message corresponds to the countermeasure put in place to limit the stolen credential reuse. We will detail the countermeasure in the following section of the paper.

5.2 Observations

The credentials were published for 15 days. We are aware that most of the experimented hackers will first verify the presence of the Bank on internet before any interaction with the website. For this reason, we suppose that most of the users are curious and intermediary and beginner's gold diggers.

We recorded in total 741 interactions (we define an interaction as an evolution in each step of the interaction workflow described in Fig. 5). These interactions are made by three categories of users: TOR protected users, Proxy protected users and non-protected users (accessing via private and public internet connections). 449 interactions are done by non-protected users; this represents more than 60% of the total interactions. 51 interactions by TOR users 6% and the rest 244 using web proxies 33% (Fig. 6).

Users using a web or a TOR proxy don't have a unique browsing finger print; those kinds of proxies are usually sharing fake browsing information in order to anonymize users and make their finger print not unique. For the non-protected users we identified 88 unique signatures (this probably corresponds to 88 unique users). On these unique signatures we were able to locate the IP addresses per country.

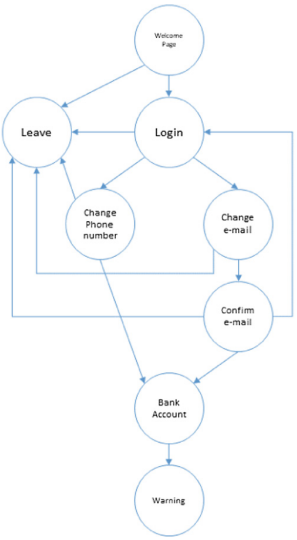

Fig. 5. Honeypot interaction state machine

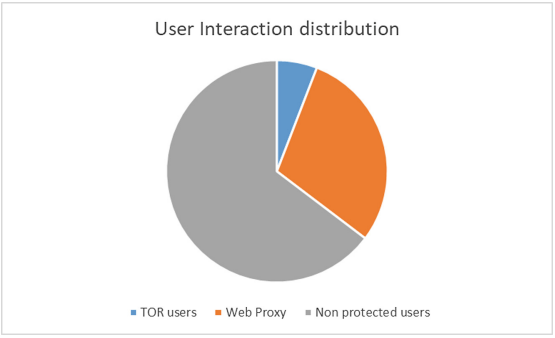

Fig. 6. User interaction distribution per browsing mode

5.3 Measurements and Results

Based on several parameters that we identified (date, time, selected user account, new password set) we guess that 43 unique users were navigating using a proxy or TOR. This number is not totally exact due to the difficulty to identify unique signatures. This raises the number of total unique users to 131.

On 741 interactions, the warning was reached 92 times 12%. As a reminder, according to Fig. 3, the warning is reached when the attacker decided to access the victim's account (after changing the password or not). The password was changed only 11 times. This represents only 1% of the interaction, this is an indication on the need for the attacker to leave the user account as it is and not raise any suspicion.

39 unique users changed the contact e-mail address, this represent 29% of the total users. 22 confirmed the validity of their address from their mail box, this represent 16% of the users.

The number of unique users who decided to ignore the warning and try again to access to the bank account is 19, this represents 14% of the whole users and they are all proxy/TOR protected.

We clearly observe the effect of the dissuading system put in place. Even if 14% of the users were not threaten by the message, 86% were and all the non-protected users who observed their navigation data decided to not follow-up on the illegal activity.

The table below summarizes all the numbers that we collected during the study (Table 1).

Table 1. Summary of all the measures of the study

Data	Number	Proportion
Total recorded interactions	**741**	**100%**
Non-protected interactions	449	60%
Web proxy interactions	244	33%
TOR interactions	51	7%
Warning reached	92	12%
Password changed	11	1%
Unique users	**131**	**100%**
Non-protected users	88	60%
TOR/Proxy users	43	40%
Changed their e-mail address	39	29%
Validated their e-mail address	22	16%
Users ignored the warning	19	14%
TOR/Proxy users ignored the warning	**19**	**100%**

6 Ethical Considerations

The honeypot deployed and the honey tokens distributed are completely fake and non-exploitable by attackers. The bank is a fake one, the services are fake and the names used for the logins are generated randomly. The data collected is only used for research purpose. All the data is deleted just after the study with a retention period of one month. The users that connected to the honeypot are not identified and their data is never crossed or combined with other datasets for identification purposes. The server used in the experiments was running only with patched software to reduce the exposure risk. We used different protection tools (anti-dos, firewall, input sanitizing, etc.). During all the experiment period, we checked permanently the logs of our systems in order to detect external access. Zero abnormal access detected. All these precautions were taken in order to avoid an external attack and an eventual collection of data.

7 Related Work

Most of the solutions proposed in the literature suggest bypassing the multiplication of password versions by adopting complex centralized infrastructures for authentication [17] and [18]. Multifactor authentication [16] and other hardware devices based solutions are popular in the literature but some of them are bypassed and they are not designed to dissuade hackers to reuse stolen passwords. Several studies were conducted to define and explain the domino effect phenomena due to the password reuse bad practice of the users [8]. Other studies explored the different password guessing techniques used from stolen credential databases [12]. These techniques are used to generate variants of a password root. These variants are frequently adopted by the users to vary their password collection set among the different domains and websites.

A Google study [15] proposed the first longitudinal measurement research tracking the origin of the different credential leak sources and their impact on user account (in term of re-use rate). This study tackles the origin of the leak and not the consequences and who is behind these consequences. The proposed mitigation techniques are based on two factor authentications.

8 Discussion and Conclusion

In this paper we proposed a new process dissuading hackers and malicious users to reuse stolen passwords to access illegally users accounts. In order to validate this solution, we created a fake banking website and spread 3300 fake credentials. We observed the behaviour of the users re-using these credentials and identified the different categories of hacker profiles. The results of our study give an idea on the type of users re-using stolen credentials, their degree of security precautions taken to perform illegal actions, and the impact of our dissuading warning-based message. One important observation, is that all the users who are not surfing behind an anonymous proxy are threaten by our prevention system especially when their navigation information are displayed. Concerning the other more precautious users only 19% ignored the system. This ratio is quite interesting according to our opinion and reflects the fear of being tracked by this category of users, that we promptly describe as vultures that want to dig some gold from crumbs resulting of big hacks.

In the literature we can find some complex solutions [19] to identify TOR proxy users, solutions that we cannot implement on a research lab level. Some other approaches [15] suggest using malwares to track hackers activities, but this approach can have legal issues.

In our future work, we want to explore new methods to collect useful information from TOR or proxy users in order to enhance the efficiency of our dissuasion system and reduce the password replay tentative.

Acknowledgement. This work was partly supported by EU-funded H2020 project C3ISP [grand no. 700294].

References

1. http://breachlevelindex.com/
2. Database of 1.4 Billion Credentials Found on Dark Web. https://www.securityweek.com/database-14-billion-credentials-found-dark-web
3. How APPLE and AMAZON Security Flaws Led to My Epic Hacking. https://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/
4. LinkedIn Lost 167 Million Account Credentials in Data Breach. http://fortune.com/2016/05/18/linkedin-data-breach-email-password/
5. Passwords for 32M Twitter accounts may have been hacked and leaked. https://techcrunch.com/2016/06/08/twitter-hack/
6. Password Leak Lists Contain 20 Percent of Microsoft Login Credentials. https://www.forbes.com/sites/adriankingsleyhughes/2012/07/16/hackers-have-20-percent-of-microsoft-login-credentials/#54cb833c7e0d
7. 63% of Data Breaches Result from Weak or Stolen Passwords. http://info.idagent.com/blog/63-of-data-breaches-result-from-weak-or-stolen-passwords
8. Ives, B., Walsh, K.R., Schneider, H.: The domino effect of password reuse. Commun. ACM **47**(4), 75–78 (2004). https://doi.org/10.1145/975817.975820
9. Mozilla: data stolen from hacked bug database was used to attack Firefox. https://arstechnica.com/information-technology/2015/09/mozilla-data-stolen-from-hacked-bug-database-was-used-to-attack-firefox/?utm_content=buffer1c53c&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
10. 2017 state of authentication report. https://fidoalliance.org/wp-content/uploads/The-State-of-Authentication-Report.pdf
11. Herley, C., Van Oorschot, P.: A research agenda acknowledging the persistence of passwords. IEEE Secur. Priv. **10**(1), 28–36 (2012). https://doi.org/10.1109/msp.2011.150
12. Das, A., Bonneau, J., Caesar, M., Borisov, N., Wang, X.: The tangled web of password reuse. In: NDSS, vol. 14, pp. 23–26, February 2014
13. Akiyama, M., Yagi, T., Aoki, K., Hariu, T., Kadobayashi, Y.: Active credential leakage for observing web-based attack cycle. In: Stolfo, S.J., Stavrou, A., Wright, C.V. (eds.) RAID 2013. LNCS, vol. 8145, pp. 223–243. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41284-4_12
14. Claycomb, W.R., Nicoll, A.: Insider threats to cloud computing: directions for new research challenges. In: 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC), pp. 387–394. IEEE (2012)
15. Thomas, K., et al.: Data breaches, phishing, or malware?: understanding the risks of stolen credentials. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1421–1434. ACM, October 2017
16. Dasgupta, D., Roy, A., Nag, A.: Multi-factor authentication. In: Dasgupta, D., Roy, A., Nag, A. (eds.) Advances in User Authentication. Springer, Cham, 185–233 (2017). https://doi.org/10.1007/978-3-319-58808-7_5
17. Sun, H.M., Chen, Y.H., Lin, Y.H.: oPass: a user authentication protocol resistant to password stealing and password reuse attacks. IEEE Trans. Inf. Forensics Secur. **7**(2), 651–663 (2012)
18. Kontaxis, G., Athanasopoulos, E., Portokalidis, G., Keromytis, A.D.: SAuth: protecting user accounts from password database leaks. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 187–198. ACM, November 2013
19. Schneier, B.: Attacking tor: how the NSA targets users' online anonymity. The Guardian, vol. 4 (2013)