Password is Dead - A Survey of Attacks

Sebastian Funke[1]

Abstract: The modern information society, with countless services between billions of connected devices has the need for scalable and usable authentication with a compromise between security and privacy [Er11]. Studies like [BP10], [Bo12] and [Ma13] show that common password-based systems are highly vulnerable with a variety of attacks from brute-forcing and dictionary attacks over statistical guessing and human prediction to shoulder surfing and social engineering. Furthermore, strong password-based security reduces usability with hardly memorable passwords, enforced by complex, invidious password policies and creates vulnerable password management overhead. Hence, password-based systems can't satisfy the security requirements of the highly connected future society. In this work I agree many other researchers [Ko04], that passwords are dead with a collection of alarming statistics, the presentation of fundamental password weaknesses and a survey of password targeted attacks.

Keywords: Passwords, Statistics, Password policies, human factor, Authentication

1 Introduction

Since many years now, researchers, including Bill Gates and IBM [Ko04] postulate that passwords are dead and have several problems: easily stolen, easily guessed and hard to remember. The weaknesses of passwords amplify with increased security requirements. Meanwhile, the amount of password targeting attack vectors is rising with fore-casted development of an overall connected world of billion potentially untrusted devices (Internet of Things) and services (Cloud). Financial, as well as, personal impacts of password theft become devastating.

Passwords were designed in the 1960's to regulate file access on large, central, shared mainframe computers [Th14]. But today they get (mis)used to protect the whole *digital identity* of everyone and control access to almost any personal data. From music-, clothing- and literature-taste, over access to cloud services full with documents, photos and videos, to financial transactions with online banking accounts.

Digital identity is defined as information used to present entities in information systems. Those entities may be persons, organizations, machines or software processes. With the up-coming new Internet of Things age, the distinction between digital/virtual and real identity gets fuzzy. To quote, Google's ex chairman, Eric Schmidt:

> Identity will be the most valuable commodity for citizens in the future, and it will exist primarily online.

[1] TU Darmstadt, CASED, Rheinstrae 75, 64283 Darmstadt, sebastian.funke.mail@gmail.com

Passwords, especially human-chosen, as a factor of knowledge to authenticate an entities identity, won't scale in the future and develop to a single point of failure. Finally, beside the well-known security impacts of password theft on companies and organizations, it will also increase the risk of a dangerous cyber-security threat: *Identity theft*. The FTC[2] even has a website dedicated to help victims of identity theft.

Recent studies show that consumer have in average 24 accounts [SH15b] and 74% of them reuse the same password for multiple accounts [Th14]. That causes a password theft domino effect and makes identity theft even easier for attackers [Te15].

A poor side-effect of countless breaches are password theft monitoring services like „Have IBeenPwned"[3], that collect disclosed credentials and allow a lookup for afflicted emails or usernames.

Motivated by this development I confirm in this work the statement that „Password is Dead" with a survey of password targeting attacks, together with correlated fundamental weaknesses of passwords.

Before I start with the survey of attacks, I motivate with some alarming statistics in subsection 1.1 and introduce an overview of password weaknesses in 1.2. In the following main part of my work (section 2), I analyze the root cause of the password attack landscape and consider all authentication perspectives. Client-based attacks (2.1) with the examples malware and browser-based attacks. Transport-based attacks (2.2) with differentiation between eavesdropping and impersonation. Classical server-based attacks (2.3), like brute-force attacks, dictionary attacks and rainbow tables. Human-based attacks (2.4) with guessability, social engineering techniques to steal passwords and physical attacks. And finally, password management based attacks (2.5) with password creation, generation, policies, password change and recover and password storage. In the last section (3) I summarize my findings and give an outlook of related technologies that improve password weaknesses.

1.1 Alarming Statistics

In this introducing subsection I give an alarming statistical overview over passwords, password targeting attacks and impact of password breaches with recent studies (2014) from TeleSign [Te15], Apple, CNN, TheVerge, etc. [Th14].

Password statistics

- 91% of passwords are one of the 1000 most common passwords
- 74% of people reuse the same password for multiple accounts
- only 44% change their password once created
- 21% use passwords older than 10 years

[2] US Federal Trade Commission Identity Theft report: https://www.identitytheft.gov/ (last checked 08/11/15)
[3] https://haveibeenpwned.com/ had 186,589,875 stolen credentials in their database (last checked 07/30/15)

- 54% of people use 5 or fewer passwords across their entire life
- „password“ and „123456“ are the most common passwords
- More than 50% forget their passwords
- 7 in 10 people no longer trust passwords to protect their accounts

Password theft statistics

- 80% of security incidents are caused by weak administrator passwords
- 2 in 5 people had a password stolen 2014
- Passwords with 6 lower-case letters are guessable in under 10 minutes
- More than 378 million people become cyber crime victims annually

Password breach impacts

- $113 billion: Global consumer cost of password hacks
- $5.4 billion: Average cost for each data security incident in the US
- $3000 and 500h: Average cost of identity theft recovery

1.2 Password Weaknesses

A passwords strength is defined by a function of length, complexity and unpredictability [UC09]. Unpredictability and complexity is close related to the passwords entropy (randomness or amount of information in bits) and a study from Florencio et al. shows [FH07] that human generated passwords have a low entropy (avg. 40.54 bits). Hence, are easy to guess for computers. With rising security requirements password strength should be increased, but as seen in the statistics above (1.1), most people never change their password and reuse it for multiple accounts.

The obvious reason for that is the decreased password usability or memorability, when humans have to input and remember strong passwords for several services. Especially with the recent human-machine interaction shift from keyboard (Desktop PC) to touch input (Smartphones), long and complex passwords became a big usability problem.

Figure 1 depicts the triangle of authentication system aspects with drawbacks between security, privacy and usability. This means, we are forced by security policies to create unusable passwords, that are hard to remember and regarding studies [Ma13], [FH07] easy to guess for computers. Password guessability, or the ability to withstand guessing by a particular password cracker with particular training data, is a security metric introduced by Weir et al. [We10] and one of the main weaknesses of passwords created by predictable humans. I will go in detail about this weakness in subsection 2.4 about human-based password attacks.

The vicious circle between usability and security led to the development of alternative authentication systems and related technologies like attribute based authentication, single

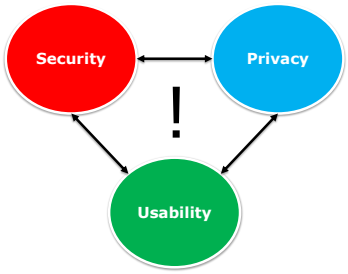

Fig. 1: Triangle of authentication system aspects

sign on, multi-factor authentication e.g. with biometrics and tokens. Unfortunately, many of those systems, e.g. biometrics have crucial privacy implications and often lack usability. The theft of biometric identifying data, as well as, the disclosure of identifying attributes among identity providers is an underrated privacy problem. Human biometrics and second channel authentication devices (Token, Smartcard, Phone, ...) as additional authentication factors of inheritance and ownership, increase security, but at the same time decrease usability again.

Finally, in the fast changing Internet with billion devices, password management processes, like passwords change, recovery and creation, become multitudinous, vulnerable and hard to deploy. Password recovery mechanisms become the weakest link in the security chain of password based authentications, e.g. when attacker use reconnaissance and social engineering to guess answers of security questions to reset their victims password [Ho12]. Security catastrophes like Heartbleed[4], where users are finally forced to change all their passwords, make the lacking usability of password management processes obvious. Hence, usability and scalability of current password-based authentication systems becomes another crucial weakness.

[4] http://heartbleed.com (last checked 08/11/15)

2 A Survey of Attacks

In this section I will explain attacks on password-based authentication systems. They rest upon the before mentioned password weaknesses (1.2) and are mostly well known since the invention of digital passwords. You can categorize those attacks by five authentication perspectives: client-, transport-, server-, human- and password management based.

Furthermore, you can classify those attacks as offline- or online-attacks. An online-attack is an attack on a running system with special constraints e.g. realtime contraints, for the attacker. Whereas, an offline-attack is conducted on the attackers machine, e.g. on a network dump, or a password hashlist.

For every authentication perspective exist complex drawbacks between security, privacy and usability aspects, that lead to a high diversity of attack vectors.

2.1 Client-based Attacks

Client-based attacks appear at user/client machine level, mostly in form of password stealing malware and are usually online- or realtime attacks. They are hard to mitigate and very efficient against any kind of system, not only password-based systems. As examples, I explain in the following subsections password stealing malware and malicious client-side browser code.

2.1.1 Malware

One of the most common approaches to obtain a victims password is a client-side malware, that sends the local stolen credentials to the attacker. Since malware is a broad term, I will focus on malware with the purpose of stealing passwords (Keyloggers and Trojan Horses).

Keylogger The easiest kind of password stealing malware is a keylogger, either software- or hardware-based. Software-based keyloggers can be applied on the clients machine as malicious software to record every keystroke of the victim with the corresponding context and send the results stealthy to the attacker. Methods to intercept keystrokes can be: Hook- (most popular), WinAPI- and Kernel-based [Di13]. Furthermore they can be distinguished in polling, event-copying and event-monitoring keyloggers. Screenmilker [Li14] is a special example of an software-based Android keylogger, that makes screenshots to extract entered passwords.

A hardware-based keylogger could be applied as interceptor between the keyboard and the machine or as passive eavesdropper for wireless keyboards. The deployment of hardware keylogger requires physical access to the victims machine or environment and can be classified as an insider attack. Keysweeper[5] from Samy Kamkar, is a sophisticated example of

[5] https://github.com/samyk/keysweeper (last checked 08/11/15)

an easy to deploy wireless keyboard eavesdropper camouflaged as a functioning USB wall charger, sending the stolen passwords over GSM to the attacker. A study from 2005 stated that 15% of all corporate machines had keyloggers installed and that 60% of all US firms got invaded by keyloggers every year.

Trojan Horses A Trojan Horse is a malware with the purpose of remote control a machine and consists out of a server and a client. Thereby the attacker places the server on the victims machine and controls the server over the client on his machine. The attacker can then choose from a variety of post-exploitation actions e.g. he can install a keylogger and he can extract the passwords (IM, email, browser, ...) saved on the victims machine. Sophisticated versions of the infamous banking bot-trojan hybrid ZeuS[6] use Man-In-The-Browser techniques like form grabbing to steal passwords. Trojan horses can be spread with user interaction over phishing emails or unattended with drive-by downloads[7] on malicious websites.

2.1.2 Browser Attacks

Browser based attacks in form of malicious plugins or Cross-Site-Scripting (XSS), are another kind of client-based attacks and can be used to steal website passwords in the browser. Special crafted malicious browser plugins like the proof of concept Zombie Browser Pack[8] can intercept password inputs on websites.

XSS as a persistent threat on place three of the OWASP top 10[9], historically falsely classified as not very critical, can also be used to steal browser stored passwords. By exploiting browsers login form auto-fill capabilities[10], an XSS injected JavaScript code can extract the victims login credential in a stealthy auto-filled background form.

2.2 Transport-based Attacks

Transport-based attacks focus on intercepting passwords in transit, e.g. in authentication transmissions between server and client. They can be classified in most cases as realtime online-attacks, but in a situation of analyzing a captured network dump file, they can also be classified as offline-attacks.

[6] https://www.symantec.com/security_response/writeup.jsp?docid=2010-011016-3514-99 (last checked 08/11/15)
[7] Drive-By-Downloads exploit vulnerabilities of browsers and browser plugins like Flash and Java
[8] https://github.com/Z6543/ZombieBrowserPack (last checked 08/11/15)
[9] http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf (last checked 08/11/15)
[10] http://www.martani.net/2009/08/xss-steal-passwords-using-javascript.html (last checked 08/11/15)

2.2.1 Eavesdropping

Also known as network sniffing or wire-tapping, an attacker as man-in-the-middle (MITM) can eavesdrop the communication between two parties, to obtain the password during an authentication challenge-response process. Technically, in a hub routed LAN or WLAN this situation appears out of the box, in a switched LAN, the attacker has to activate the switch's promiscuous mode. Wireshark[11] is one of the most popular network analysis tools and contains many features e.g. to extract passwords of common network protocols from captured traffic. Even though most authentications are encrypted e.g. over HTTPS, an attacker could use weaknesses in TLS to strip or downgrade the encryption.

Sophisticated attackers like secret agencies may also act as a legitimate party in the middle of the communication. Therefore they provide a trusted certificate, decrypt the communication, extract the password and encrypt it again.

2.2.2 Impersonation

Generally spoken, impersonation attacks are not focused on stealing passwords, but on bypassing an authentication system and impersonating the victim. Finally, impersonation can be used to create an eavesdropping man in the middle situation or to change the victims password. I will present two soft-techniques to achieve impersonation.

IP-Spoofing One technique is IP spoofing, respectively ARP[12] poisoning. Thereby the attacker forges an ARP package, to injects his IP for a different MAC-Address (mostly the network routers MAC) in the ARP table of a victim to impersonate a different machine and get all traffic between victim and this machine. MAC spoofing is a similar approach, where the attacker changes his MAC address to the address of the machine he wants to impersonate.

Session Hijacking An authentication session can be hijacked, to impersonate the victim itself, by sniffing the session token. This attack is very common in open (unencrypted) wireless networks and many special purpose tools like zANTI[13] for Android exist to hijack sessions (e.g. Facebook). With the captured session token, the attacker can act as the victim and could change the password to compromise the hijacked service.

2.3 Server-based Attacks

Server-based attacks focus on password theft at server/service level and can be online and offline. Online attacks on servers authentication interfaces can be automated very well,

[11] https://www.wireshark.org/ (last checked 08/11/15)
[12] ARP - Address Resolution Protocol (Used to translate between IP- and MAC-Addresses)
[13] https://www.zimperium.com/zanti-mobile-penetration-testing (last checked 08/11/15)

but can be mitigated easily, by implementing password re-entry penalties and password aging policies[14]. If the server got hacked and the securely stored, hashed passwords were stolen e.g. from the database, the attacker can leverage an offline attack. Anyhow, for both classes of server-based attacks, the attacker can use the same kind of classical attacks, that are explained below.

2.3.1 Bruteforcing

The most popular attack on passwords is the brute-force attack, where the attacker challenges the authentication system with all possible character combinations until the correct password was found. The number of combinations in the worst case is determined by C^l with C as character space and l as password length. Professional cracker facilitate GPU cluster[15] with SIMD multi-processing instructions[16] or distributed/cloud-computing to speed-up their attacks. A simple[17] offline brute-force attack against an average MD5 password[18] takes only around 3 days. With 2 characters more it would take already around 27 years (see Fig. 2).

	Password Length						
Char. Space	**6**	**7**	**8**	**9**	**10**	**11**	**12**
10 [0-9]	1 ms	10 ms	100 ms	1 s	10 s	2 m	17 m
26 [a-z]	<1 s	8 s	4 m	2 h	2 d	42 d	3 y
52 [A-Z;a-z]	20 s	17 m	15 h	33 d	5 y	238 y	12.4k y
62 [A-Z;a-z;0-9]	58 s	1 h	3 d	159 d	27 y	1.6k y	102k y
92 (+symbols)	13 m	21 h	84 d	22 y	2.1k y	202k y	19m y

Fig. 2: Maximum brute-force computation time with 1 billion passwords per second in an offline attack on MD5 passwords[19]

This approach has a 100% success rate, only constraint by the attackers processing power. But it doesn't scale for passwords longer than 10 characters, described by Rob Graham as „expontential wall of brute force cracking"[Go12], seen in figure 3.

Tools like THC Hydra[20] can be used to execute multi-threaded, fast, online brute-force attacks against authentication services on numerous protocols.

On one hand side, the average password length increases (Fig. 4), that makes brute-force attacks less successful, but on the other hand side, regarding Moore's Law[21], the overall processing power will double every two years. Hence, the brute-force attack risk is ris-

[14] Systems force users to change the password after a specific password expiration time
[15] CUDA: https://developer.nvidia.com/cuda-zone (last checked 08/11/15)
[16] SSE and OpenMP http://openmp.org/wp/ (last checked 08/11/15)
[17] Standard pc from 2011 with powerful GPU: 1 billion passwords per second, http://www.hammerofgod.com/passwordmachine.php (last checked 08/11/15)
[18] 8 characters out of 26+26+10 ([A-Z;a-z;0-9])
[19] http://www.lockdown.co.uk/?pg=combi (last checked 08/11/15)
[20] https://thc.org/thc-hydra/ (last checked 08/11/15)
[21] http://www.mooreslaw.org/ (last checked 08/11/15)

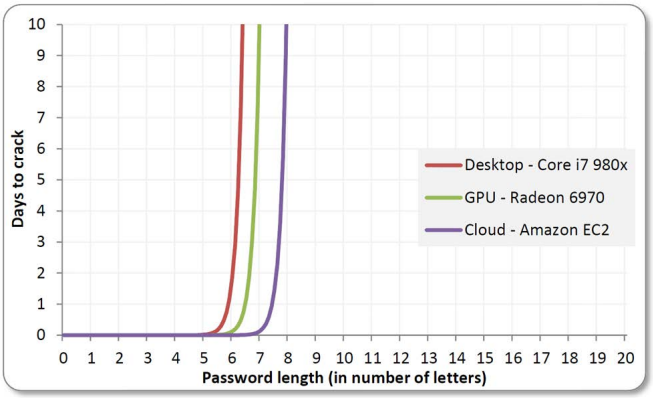

Fig. 3: Brute-force cracks well shorter passwords. But take days or months for longer passcodes, even when using Amazon's cloud-based EC2 service.

year	study	length	% digits	% special
1989	Riddle et al.	4.4	3.5	—
1992	Spafford	6.8	31.7	14.8
1999	Wu	7.5	25.7	4.1
1999	Zviran and Haga	5.7	19.2	0.7
2006	Cazier and Medlin	7.4	35.0	1.3
2009	*RockYou leak*	7.9	54.0	3.7

Fig. 4: Password strength history [Bo12]

ing and mitigation strategies like: re-entry time penalties, password-aging- and password-strength-policies, should be applied mandatory.

2.3.2 Dictionary Attacks

A dictionary attack is basically a brute-force attack, but with a list of passwords (dictionary) instead of all character combinations. The success depends only on the way how the dictionary gets generated and ordered, as well as its size. Hence, there is no guarantee to crack all passwords with this technique, but it significantly increases the speed and cracks most of the weaker passwords [Go13]. This type of attack can also be called *Guessing Attack*, because the construction of a reasonable dictionary can be compared to guessing the password. More about the aspects of human chosen randomness and guessing passwords in subsection 2.4.1 about guessability.

John the Ripper (JtR)[22] as one of the most famous cracking tools uses intelligent *mangling rules* to construct sophisticated dictionaries with predictable patterns, like number and symbol substitutions, adding of year numbers, etc. More specific it uses a character-by-character *Markov model*, first introduced by Narayanan and Shmatikov [NS05] for dic-

[22] http://www.openwall.com/john/ (last checked 08/11/15)

tionary creation. Weir et al. formally modeled 2009 password structure with a *probabilistic context free grammar (PCFG)* to extract mangling rules out of training data like a breached password corpus [We09]. Thereby, dictionaries can be created ordered by probability of password structure and component strings of characters. Hence, depending on the training data, e.g. context-specific dictionaries[23], it allows very efficient password guessing. This technique is still considered as the state-of-the-art cracking method, used in recent password studies from CMU [Ma13].

Furthermore, Narayanan's approach in JtR got significantly[24] improved 2015 with the „Ordered Markov ENumerator (OMEN)" cracker of Castelluccia et al. [Dü15]. After 0.2 billion guesses, OMEN even outperforms Weir's PCFG cracker by 20% for the RockYou test set as seen in Fig. 5.

Algorithm	Training Set	#guesses	Testing Set		
			RY-e	MS	FB
Omen	RY-t	10 billion	80.40%	77.06%	66.75%
	RY-t	1 billion	68.7%	64.50%	59.67%
PCFG	RY-t	1 billion	32.63%	51.25%	36.4%
JtR-Markov	RY-t	10 billion	64%	53.19%	61%
	RY-t	1 billion	54.77%	38.57%	49.47%
JtR-Inc	RY-t	10 billion	54%	25.17%	14.8%

Fig. 5: Performance comparison between state of the art cracker [Dü15] with training set from RockYou password leaks (RY-t) and test sets of leaked passwords from RockYou (RY-e), Myspace (MS) and Facebook (FB) [Dü15]

Dan Goodin described 2013 in his in-depth article: Anatomy of a hack, how an unexperienced reporter could decipher 47% of a list of 16,000 hashed passwords in one day and a professional cracker could decipher 90% in 20 hours [Go13]. This simulation shows how weak and easy to guess average passwords really are.

2.3.3 Rainbow Tables

Rainbow tables can be seen as a combination of the two previous attacks to reduce the cracking complexity of offline attacks to table lookup complexity. The attacker pre-computes the hash value of all possible password combinations and saves the result as rainbow table, to effectively lookup the table against a hash to find the corresponding plaintext password. This type of attack is only limited by the attackers storage capabilities and is very efficient if the attacker knows the hash function and salt mechanism used to store the passwords.

Rainbowcrack[25] is the most popular tool for fast, time-memory trade-of hash lookup. *N* hashes can represented in a chain with start- and end-value, thereby reduce the storage and limit the re-computation time to N hashes. As example, for MD5, a rainbow table for a 6

[23] Password contexts of victim: country, language, service, age, hobbies, birth years, ...

[24] Castelluccia et al. evaluated their OMEN cracker as 8 times faster than John the Ripper with Narayanan's password indexing

[25] http://project-rainbowcrack.com/ (last checked 08/11/15)

out of 64 ([A-Za-z0-9./]) password, would be $64^6 \cdot 16$ Bytes ≈ 1 TB big. Context-specific[26] rainbow tables can be bought and downloaded from the developers website.

2.4 Human-based Attacks

Passwords are usually created by humans. Hence, if the authentication infrastructure and technology is secure, the human factor is still the weakest part in the security chain. I show with recent studies, how it is possible to create very efficient dictionaries for guessing attacks, based on the guessability metric of passwords. Furthermore I introduce several other human-based attacks like social engineering and physical attacks.

2.4.1 Guessability

Password guessability, introduced as a main weakness of passwords in subsection 1.2 can also be described as the science of guessing passwords and is based on the human lack of creating random passwords with high entropy.

Historically, old studies from 1979 stated, that an attacker could guess 20 - 50% of all passwords with dictionary sizes in the range of 2^{20} - 2^{30} [Bo12]. Nick Berry's PIN analysis from 2012 [Be12] has shown significant predictability of human chosen PINs, as seen in Fig. 6.

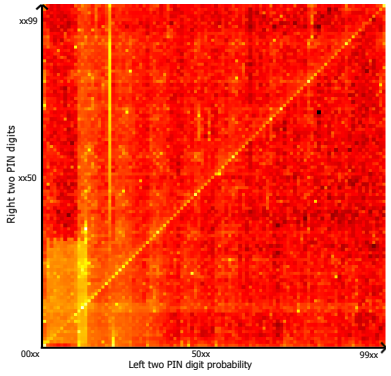

Fig. 6: PIN distribution heat map. Light dots indicate frequently used PINs, diagonal dots are repeated numbers (e.g. 4444), the vertical line in the left are PINs between 1940-1999

Recent studies from Bonneau et al. (Cambridge) [Bo12] and Cranor et al. (Carnegie Mellon University) [Ma13] found many interesting properties of passwords in relation to different demographics. With a scientific and privacy preserving analysis on 25,000 to 70 million legitimate passwords and a state of the art cracking technique [We09], they found that (computer) scientists create significantly stronger passwords than business people.

[26] Hash function should be known or detected by attacker, e.g. with https://md5hashing.net/hash_type_checker (last checked 08/11/15)

The well known, most common passwords: password, 123456, 12345678, qwerty, abc123, 111111, monkey, etc., are still widely used, as seen in recent TeleSign statistics [SH15b] in Figure 7.

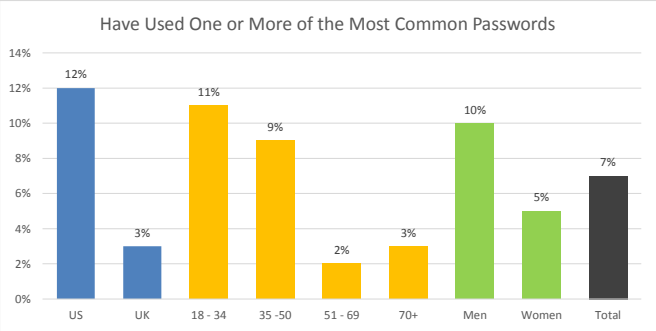

Fig. 7: Statistics about the usage of the most common passwords[27]

They identified common patterns: names in passwords, keyboard pattern, service-context, hobbies and things user like in general. Letter and number substitutions (O = 0, I = 1, E = 3, ...), as well as letter and symbol substitutions (S = $, I = !, ...). Furthermore they found the symbols: !, @, # and * as the most common used password symbols. Another study from 2014 [La14] analyzed 5000 leaked, *PCI-DSS*[28] compliant, supposedly strong production passwords and confirmed the previous study results. Below, I present a summary of the detailed predictable human password habits found by the study of 2014 [La14]:

- 9.6 characters average password length
- 6.1 lower-case letters in average
- 2.2 numbers in average
- 1.1 upper-case letters in average
- 0.2 symbols in average
- 86% use only 1 uppercase (mostly first)
- 40% use only a single number
- 20% use a year (1900-2015)
- 90% only use 1 symbol (! 29% , . 19% , @ 15% and # 14%)
- 68% use repeated symbol (e.g. ***)
- 10% use a password similar to their username
- 13% use default password pattern (e.g. P4$$w0rd, defaultPass*ServiceName)
- 75% use passwords containing dictionary words (168,000 english dictionary)
- 7% use passwords containing keyboard pattern

An attacker could use reconnaissance on his victim and the habits above with the mangling rule techniques described in subsection 2.3.2 and generate a very effective dictionary with information about the victim e.g. birthday, name, hobbies etc. Those study results can be used to create better password policies, but this decreases the memorability of passwords and new habits will appear over time and increase password guessability again. Obviously a vicious cycle.

[27] https://www.telesign.com/site/wp-content/uploads/2015/06/TeleSign-Consumer-Account-Security-Report-2015-FINAL.pdf (last checked 08/11/15)

[28] PCI-DSS: Payment Card Industry Data Security Standard

Summarizing, although the average password length and theoretical strength is rising, forced by password policies, it was never easier to crack passwords with computer aided password guessing, even for unexperienced attackers as described in subsection 2.3.1.

2.4.2 Social Engineering

Social engineering (SE) is a broad term, defined as any act that influences a person to take an action that may or may not be in their best interest and is a common attack on humans as weakest component of IT systems. Between 2009 and 2011, 48% of 850 international IT professionals were victim of social engineering with 25 or more attacks [Go11]. Social engineers have a huge repertoire of diverse techniques. In the following paragraphs I focus on some techniques used for password theft.

Phishing Phishing, as one of the most famous social engineering technique used for password theft is on the rise, caused by increasing, creative attack vectors in new authentication systems. In October 2013, RSA identified more than 62,000 phishing attacks[29].

Those attacks can be faked emails, SMS (Smishing), phone calls (Vishing) that encourage the user to tell the attacker their credentials, e.g. in a stealthily, faked, attacker hosted login interface. More creative approaches can involve QR-codes, reflected XSS, Click-Jacking[30] and Tab-Napping[31].

Insider Attacks So called insider attacks are the main entry point for big IT system breaches and often don't get published. A malicious employee might extract saved passwords from his or his co-workers machines or if written on paper, from their desks. In 2013, personal data from roughly 40 million customer got stolen from Target[32] over access credentials from the company's refrigeration vendors [DS14]. Even though many insider attacks never get published, it got estimated that at least 80 million insider attacks occur in the US ever year, with total cost in tens of billion dollars a year and regarding KPMG insider attacks are still rising. Hardware-based Keyloggers, like mentioned in section 2.1.1, are already a more sophisticated form of a physical human-based insider attack. Complex, growing IT infrastructures, the social media explosion and bring your own device (BYOD) policies endanger companies internal IT and can lead to increased insider attack password theft.

[29] http://searchsecurity.techtarget.com/feature/Social-engineering-attacks-Is-security-focused-on-the-wrong-problem (last checked 08/11/15)
[30] Used to stealthy login, in an invisible background form
[31] A browser tab, e.g. opened over click-jacking, changes its appearance, e.g. GMail favicon, etc., over time to a fake login interface and the user accidentally identifies it later as his legitimate GMail tab and sends his credentials to the attacker
[32] http://www.target.com/ (last checked 08/11/15)

2.4.3 Physical Attacks

Physical attacks typically require the attacker to be physically present near the victim. As simple it is, the most effective way to steal a complex password is to physical threat or blackmail the victim, also called rubber-hose cryptanalysis. Dumpster diving is a common physical method for reconnaissance, but also for password theft, if employees write passwords down or print them out. USB baiting, by spreading malicious USB sticks, e.g. on parking spaces, is an often used physical attack to smuggle malware in internal company networks. Side-channel attacks on systems with running crypto algorithms, that statistically analyze characteristics like temperature-changes, power-consumption, EM-emissions, etc., are another physical attack to extract cryptographic secrets like passwords from, even air-gapped, machines. The most popular type of those attacks is the Tempest attack, that extracts data in general from EM-emissions. Recently shown, this attack can be easily deployed with a SDR[33] and radio receiver as a hand-palm sized device hidden in a pita bread to extract RSA and ElGamal keys [Ge15].

Shoulder Surfing simply describes an attacker who spies out a victims password over his shoulder, when he enters the password. This principle can be extended with cameras to record the victims lips movement, since many people tend to speak their password silently when typing it. Especially in case of PIN inputs, criminals can exploit subtle characteristics of input terminals and user behaviors, e.g. monitor the remaining heat pattern on the keys. Heat pattern monitoring can reduce a 4 digit PIN number space from 10.000 to 25 possibilities [MMS11].

2.5 Attacks on the Password Management

Passwords as single proof of identity rely on critical background maintenance processes. In the following subsection I will explain risks and weaknesses in common password management process, including: changing of passwords, recovery of forgotten passwords, secure password storage and the (guided) policy restricted creation of passwords.

2.5.1 Password Creation, Generation and Policies

Passwords life-cycle start with human creation or machine generation. As mentioned in subsection 2.4.1, human chosen passwords lack randomness and uniqueness. Hence, password strength meters and password creation guides were invented to support people during policy compliant password creation. A study from Shay et al. [Sh15a] analyzed the impact of several password meters and creation guides on guessability and usability. Analyzed approaches include: live feedback password creation, guided password creation and creation with random insertion with several password policies. Regarding their results, many password strength meters, used in production systems, still allow weak passwords and force

[33] Software defined radio

the people to develop the password habits presented in subsection 2.4.1. Furthermore, they found out that a multistep guided password creation wizard increases the memorability and theoretical security, but the created passwords are around 5% more likely to guess than passwords created with a common pattern policy.

Machine generated passwords are strong with the the drawback of very weak memorability. Since many password strength meters still suggest weak passwords, the implementation of secure password generation seems not trivial. Many such systems use predictable pseudo-random number generators and sources of low entropy and thereby increase the guessability. For example, most routers use weakly generated WPS[34] PINs and can be guessed in max. 4-10 hours with the Open Source tool Reaver[35] [Ga11].

2.5.2 Password Change and Recovery

Password recovery processes, easily abused with social engineering, often break the security model of password based authentication systems [Ho12]. A common approach are so called security questions, that help service provider to identify users who forgot their password and want to reset it. With a little bit social engineering and reconnaissance, the answers to those questions, e.g. your mothers birth name, your favorite band, etc., are easily guessed. Providing wrong or random answers decreases the abuse risk, but increases the lock out risk. Other approaches were proposed, e.g. Facebook uses password reset by default over at least three vouching friends. Considering 83 million Facebook accounts are fake, an attacker can easily reset[36] victims password, if he has control over three fake friend on their friend list.

Also the process of changing passwords is often troublesome and not user-friendly. A study confirmed that only 44% of people change their passwords. But it is an important security best practice to change passwords frequently, enforced in some companies with password expiration policy, to reduce brute-force attack risks. Especially after security catastrophes like Heartbleed users were forced to change almost all of their passwords and faced frustration and waste of time changing their passwords in countless non-uniform web interfaces.

2.5.3 Password Storage

Caused by complex password policies, user have problems to memorize their passwords and create ways to store their passwords locally. Passwords should never be written down, but many people do it anyway and there is even a market for password books[37]. 13% write

[34] WPS: Wifi Protected Setup

[35] https://code.google.com/p/reaver-wps/ (last checked 08/11/15)

[36] http://www.rafayhackingarticles.net/2012/08/hack-facebook-account-by-exploiting.html (last checked 08/11/15)

[37] The Personal Internet Address & Password Log Book, http://www.amazon.com/gp/product/1441303251 (last visited 08/11/15) is Amazon's second best selling book in the category Internet and Telecommunications

their passwords down [Ma13] or store them unencrypted in a local file and thereby create other attack vectors, like password theft by insiders, dumpster-diving and malware.

On the server-side, passwords have to be stored as hashes with salt, using a cracking resistant hash function, otherwise after a breach attacker can easily crack the passwords with rainbow tables. In one of largest case of password breaches of Adobe 2013, 150 million weak secured user records got exposed [Du13]. Embarrassingly they not only stored all passwords weakly encrypted[38] with a single key, they also stored the user defined password hints in plaintext. Hence, the attackers did not even need to crack the credentials, but could guess the passwords of most users using the password hint.

3 Conclusion

Summarizing, I showed several attack vectors and password weaknesses, explained the password policy driven vicious cycle between guess- and usability, concluded that passwords were never easier to crack and that they get misused to protect our whole (digital) identity.

Regarding Cormac et al. [HvO12] passwords will still persist in many legacy systems and it takes a while until new authentication systems get approved in the IT world.

In the meantime we have to improve password policies and spread security education, e.g. the password guidance report recently published by the GCHQ[39][GC15] highlights the importance of changing default passwords, cope with password overload, awareness of user- and machine-chosen password weaknesses, password storage mistakes, the prioritization of admin and remote user passwords and recommends protective account monitoring.

Password related technologies like Password Manager, Single-Sign-On (SSO) and Multi-Factor authentication are on the rise to improve usability and security of password bases authentication systems. And with FTC jurisprudence we move towards law enforced reasonable strong authentication mechanisms [SH15b].

SSO, already widely used in the web, e.g. the OpenID[40] standard, shifts the authentication responsibility to Identity Provider (IdP) like Google or Facebook and thereby increase the usability with password-less session-token based authentication. But, this development increased IdP based phishing attacks and introduces critical privacy implications as mentioned in subsection 1.2.

Password manager store passwords encrypted with a master password, either in a local database or in the cloud and improve storage and memorability issues mentioned in subsection 2.5.3. Unfortunately, they are also not the silver bullet. Silver et al. found many weaknesses in the most popular password managers, like: attacks on password auto-fill

[38] all passwords encrypted with 3DES in ECB mode is not a secure encryption, if the symmetric key is broken, all passwords are broken
[39] GCHQ (Government Communications Headquarters) is the British secret agency
[40] http://openid.net (last checked 08/11/15)

methods in browsers, password generation and web interface flaws [Si14]. In June 2015, LastPass[41], one of the most popular cloud-based password managers, got attacked and email addresses, password reminders, server per user salts, and authentication hashes were compromised.

Obviously SSO and password manager introduce a single point of failure and lock out risk, in case the master password or IdP password gets lost or broken during an offline attack. Hence, they dramatically increase the risk of digital identity theft and the domino effect mentioned in section 1. Future improvements should automate the password changing efforts for the stored passwords of different services, as well as introduce frequent password changing policies for the password managers master password and IdP password.

Further single point of failures mitigation and increased security can be established with an additional authentication factor on a second channel. The FIDO alliance[42] published an universal second factor standard (U2F) for heterogeneous integration of biometrics (Fingerprint, Iris, DNA, ...), SW/HW token, SMS, Phone and future behavioral identifying attributes in persisting password based authentication systems. Many services already support second factor authentication[43].

Finally, as seen several times during this work, consciously human chosen secrets will always be predictable. Hence, additional future multi-factors should be based on unconscious behaviors like eye movement, typing style or in general not only based on who we are, but also on what we do, where we go, when, what we have with us (wearables, phone, implants) and how we act when we are there [Ho12]. They can be added in parallel to the old password authentication to evaluate their effectiveness and over time with improved robustness and usability someday replace passwords completely.

References

[Be12] Berry, Nick: PIN analysis. September 2012. http://www.datagenetics.com/blog/september32012/ [Online; last checked 08/11/15].

[Bo12] Bonneau, Joseph: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: IEEE Symp. Security and Privacy. 2012.

[BP10] Bonneau, Joseph; Preibusch, Sören: The password thicket: technical and market failures in human authentication on the web. In: WEIS '10: Proceedings of the 9th Workshop on the Economics of Information Security. June 2010.

[Di13] Dimov, Ivan: Keyloggers: How They Work and More. August 2013. http://resources.infosecinstitute.com/keyloggers-how-they-work-and-more/ [Online; last checked 08/11/15].

[DS14] D., Upton; Sadie, Creese: The Danger from Within. September 2014. https://hbr.org/2014/09/the-danger-from-within [Online; last checked 08/11/15].

[41] https://lastpass.com (last checked 08/11/15)

[42] Fast-Identity-Online (FIDO) is an alliance of big companies with the goal to secure authentication over additional identifying factors, https://fidoalliance.org (last checked 08/11/15)

[43] https://www.turnon2fa.com (last checked 08/11/15) a campaign to activate second factor authentication in existing services

[Du13] Ducklin, Paul: Anatomy of a password disaster - Adobe's giant-sized cryptographic blunder. November 2013. https://nakedsecurity.sophos.com/2013/11/04/anatomy-of-a-password-disaster-adobes-giant-sized-cryptographic-blunder/ [Online; last checked 08/11/15].

[Dü15] Dürmuth, Markus; Angelstorf, Fabian; Castelluccia, Claude; Perito, Daniele; Abdelberi, Chaabane: OMEN: Faster Password Guessing Using an Ordered Markov Enumerator. In: Engineering Secure Software and Systems - 7th International Symposium, ESSoS 2015, Milan, Italy, March 4-6, 2015. Proceedings. pp. 119–132, 2015.

[Er11] Ericsson: More than 50 billion connected devices - taking connected devices to mass market and profitability. Technical report, February 2011.

[FH07] Florencio, Dinei; Herley, Cormac: A Large-scale Study of Web Password Habits. In: Proceedings of the 16th International Conference on World Wide Web. WWW '07, ACM, New York, NY, USA, pp. 657–666, 2007.

[Ga11] Gallagher, Sean: Researchers publish open-source tool for hacking WiFi Protected Setup. December 2011. http://arstechnica.com/business/2011/12/researchers-publish-open-source-tool-for-hacking-wifi-protected-setup/ [Online; last checked 08/11/15].

[GC15] GCHQ: Password Guidance Simplifying Your Approach. September 2015.

[Ge15] Genkin, Daniel; Pachmanov, Lev; Pipman, Itamar; Tromer, Eran: Stealing Keys from PCs using a Radio: Cheap Electromagnetic Attacks on Windowed Exponentiation. Cryptology ePrint Archive, Report 2015/170, 2015. https://www.cs.tau.ac.il/tromer/papers/radioexp.pdf [Online; last checked 08/11/15].

[Go11] Goodchild, Joan: Social engineering attacks costly for business. September 2011. http://www.csoonline.com/article/2129673/social-engineering/social-engineering-attacks-costly-for-business.html [Online; last checked 08/11/15].

[Go12] Goodin, Dan: Why passwords have never been weakerand crackers have never been stronger. August 2012. http://arstechnica.com/security/2012/08/passwords-under-assault/ [Online; last checked 08/11/15].

[Go13] Goodin, Dan: Anatomy of a hack: even your 'complicated' password is easy to crack. wired.com, May 2013. http://www.wired.co.uk/news/archive/2013-05/28/password-cracking/viewall [Online; last checked 08/11/15].

[Ho12] Honan, Mat: Kill the Password: Why a String of Characters Cant Protect Us Anymore. November 2012. http://www.wired.com/2012/11/ff-mat-honan-password-hacker/ [Online; last checked 08/11/15].

[HvO12] Herley, Cormac; van Oorschot, Paul: A Research Agenda Acknowledging the Persistence of Passwords. IEEE Security&Privacy Magazine, 2012.

[Ko04] Kotadia, Munir: Gates predicts death of the password. February 2004. http://www.cnet.com/news/gates-predicts-death-of-the-password/ [Online; last checked 08/11/15].

[La14] Lampe, Jonathan: Beyond Password Length and Complexity. January 2014. http://resources.infosecinstitute.com/beyond-password-length-complexity/ [Online; last checked 08/11/15].

[Li14] Lin, Chia-Chi; Li, Hongyang; Zhou, Xiao-yong; Wang, XiaoFeng: Screenmilker: How to Milk Your Android Screen for Secrets. In: 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014. 2014.

[Ma13] Mazurek, Michelle L.; Komanduri, Saranga; Vidas, Timothy; Bauer, Lujo; Christin, Nicolas; Cranor, Lorrie Faith; Kelley, Patrick Gage; Shay, Richard; Ur, Blase: Measuring Password Guessability for an Entire University. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. CCS '13, ACM, New York, NY, USA, pp. 173–186, 2013.

[MMS11] Mowery, Keaton; Meiklejohn, Sarah; Savage, Stefan: Heat of the Moment: Characterizing the Efficacy of Thermal Camera-based Attacks. In: Proceedings of the 5th USENIX Conference on Offensive Technologies. WOOT'11, USENIX Association, Berkeley, CA, USA, pp. 6–6, 2011.

[NS05] Narayanan, Arvind; Shmatikov, Vitaly: Fast Dictionary Attacks on Passwords Using Time-space Tradeoff. In: Proceedings of the 12th ACM Conference on Computer and Communications Security. CCS '05, ACM, New York, NY, USA, pp. 364–372, 2005.

[Sh15a] Shay, Richard; Bauer, Lujo; Christin, Nicolas; Cranor, Lorrie Faith; Forget, Alain; Komanduri, Saranga; Mazurek, Michelle L.; Melicher, William; Segreti, Sean M.; Ur, Blase: A Spoonful of Sugar?: The Impact of Guidance and Feedback on Password-Creation Behavior. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. CHI '15, ACM, New York, NY, USA, pp. 2903–2912, 2015.

[SH15b] Solove, Daniel J.; Hartzog, Woodrow: Should the FTC Kill the Password? The Case for Better Authentication. July 2015. http://ssrn.com/abstract=2636366 [Online; last checked 08/11/15].

[Si14] Silver, David; Jana, Suman; Boneh, Dan; Chen, Eric; Jackson, Collin: Password Managers: Attacks and Defenses. In: 23rd USENIX Security Symposium (USENIX Security 14). USENIX Association, San Diego, CA, pp. 449–464, August 2014.

[Te15] TeleSign: Protect yourself with more than a password. June 2015. http://t3n.de/news/zwei-faktor-authentifizierung-infografik-614224/passwort_2fa_zwei-faktor-authentifizierung-infografik/ [Online; last checked 08/11/15].

[Th14] Theverge; Doctrackr; Myidkey; CNN; Technologyreview; Fidoalliance; Apple: Is the Password Dead? November 2014. http://www.visualistan.com/2014/11/is-password-dead-infographic.html [Online; last checked 08/11/15].

[UC09] US-Cert: Choosing and Protecting Passwords. May 2009. https://www.us-cert.gov/ncas/tips/ST04-002 [Online; last checked 08/11/15].

[We09] Weir, M.; Aggarwal, S.; de Medeiros, B.; Glodek, B.: Password Cracking Using Probabilistic Context-Free Grammars. In: Security and Privacy, 2009 30th IEEE Symposium on. pp. 391–405, May 2009.

[We10] Weir, Matt; Aggarwal, Sudhir; Collins, Michael; Stern, Henry: Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords. In: Proceedings of the 17th ACM Conference on Computer and Communications Security. CCS '10, ACM, New York, NY, USA, pp. 162–175, 2010.