Filozofická fakulta Univerzity Palackého v Olomouci
Katedra obecné lingvistiky

Role lingvistiky v kybernetické bezpečnosti

magisterská diplomová práce

Autor: Bc. Barbora Anna Janečková
Vedoucí práce: Mgr. Vladimír Matlach, PhD.

Olomouc
2021

Prohlášení

Prohlašuji, že jsem magisterskou diplomovou práci *Role lingvistiky v kybernetické bezpečnosti* vypracovala samostatně a uvedla jsem veškerou použitou literaturu a veškeré použité zdroje.

V Olomouci dne Podpis

Na tomto místě bych chtěla poděkovat především svému vedoucími práce Mgr. Vladimíru Matlachovi, PhD. za podporu, poskytnuté rady a konstruktivní kritiku, které pomohly tuto práci zkompletovat.

Abstrakt

Název práce: Role lingvistiky v kybernetické bezpečnosti
Autor práce: Bc. Barbora Anna Janečková
Vedoucí práce: Mgr. Vladimír Matlach, PhD.
Počet stran a znaků: 80 stran; 143 794 znaků
Počet příloh: 4
Abstrakt: Práce se zabývá možností aplikace metod kvantitativní lingvistiky a zpracování přirozeného jazyka na poli kybernetické bezpečnosti. Vedle teorie spojené s hesly (jako je jejich tvorba a síla, způsoby uložení a ochrana v databázích a typologie útoků na soubory kryptograficky chráněných hesel) se věnuje analýzám čtyř vybraných souborů hesel z reálného prostředí. V první řadě shrnuje jejich základní kvantitativní vlastnosti s cílem získat obecný náhled na dané soubory. Dále se věnuje sémantické analýze, pomocí které identifikuje tematickou povahu nejčastějších hesel. Vedle sledování obecných vlastností se tato práce také orientuje na otázku toho, co činí hesla unikátními. V neposlední řadě je provedena jazykově motivovaná analýza struktur hesel, v rámci které jsou popsány typické struktury rozložení konsonantů a vokálů v heslech. Na základě této analýzy je v závěru práce prozkoumána možnost využití těchto struktur při získávání hesel z jejich kryptograficky chráněné podoby.
Klíčová slova: kybernetická bezpečnost, korpusová lingvistika, zpracování přirozeného jazyka, hašovací funkce, hashcat, fonotaktika

Abstract

Title: The Role of Linguistics in Cybersecurity
Author: Bc. Barbora Anna Janečková
Supervisor: Mgr. Vladimír Matlach, PhD.
Number of pages and characters: 80 pages; 143,794 characters
Number of appendices: 4
Abstract: The work deals with the possibility of applying the methods of quantitative linguistics and natural language processing in the field of cybersecurity. Apart from the theory concerning passwords (such as their creation and strength, methods of storage and protection in databases, and the typology of attacks on cryptographically protected passwords), an analysis of four selected leaked password datasets is conducted. First of all, their basic quantitative properties are summarised in order to obtain an overview of the datasets. Next, a semantic analysis is conducted so as to identify the thematic nature of the most common passwords. In addition to describing the general properties of the datasets and passwords, this work also focuses on the question of what makes passwords unique. Last but not least, a linguistically motivated analysis of password structures is performed, within which typical structures of the distribution of consonants and vowels are described. Based on this analysis, the possibility of using such structures in obtaining passwords from their cryptographically protected form is investigated.
Keywords: cybersecurity, corpus linguistics, natural language processing, hash functions, hashcat, phonotactics

Obsah

Úvod

Předložená diplomová práce se pohybuje na pomezí lingvistiky (především jejích kvantitativních odvětví) a kybernetické bezpečnosti, jelikož objektem jejího zájmu jsou hesla, která si uživatelé online služeb volí pro přístup do svých účtů. Hlavní motivací pro tuto práci je prozkoumání možnosti přístupu k problematice čistě technologického oboru pomocí metod lingvistiky jakožto vědy převážně humanitní. Tento přístup staví nejen na předpokladu, že uživatelská hesla jsou z velké míry tvořena lexikem (které je vlastní lingvistickému bádání), ale také na možnosti sledování kombinatoriky užitých znaků v heslech, a to na základě primárně lingvistických vlastností – což je přístup doposavad jen zřídkavý, a tedy by jeho poznatky mohly oboru kybernetické bezpečnosti přinést nový úhel pohledu. Bádání na pomezí těchto dvou oblastí pak může být obecně přínosné online bezpečnosti v digitální době, jelikož ukáže na dosud málo popsané slabiny uživatelských hesel a potenciálně otevře otázku toho, zda jsou hesla dostačující k ochraně uživatelských informací.

V prvních částech práce bude nastíněn kontext, ze kterého vycházíme – a to nejen co se týká lingvistiky, ale především hesel. Po představení zásadních termínů a kontextu komputační a kvantitativní lingvistiky se v kapitole *Hesla: úvod do kontextu* budeme věnovat aspektům tvorby hesla z hlediska uživatele a impaktu, jaký má způsob tvorby na sílu hesla. Dále popíšeme způsoby, jakými jsou hesla ukládána v databázích služeb či aplikací, přičemž se především zaměříme na tzv. hašovací funkce. Poslední část kontextu bude věnována typologii útoků na kryptograficky chráněná hesla či jejich soubory. Následující kapitoly této práce se věnují měřením a analýzám čtyř souborů hesel z reálného prostředí, které unikly na internet. Kapitola *Datasety* uvede souhrnný popis těchto souborů hesel, a to jak z hlediska metainformací, kdy bude čtenář seznámen s geografickými a demografickými aspekty zdroje každého souboru, tak i z hlediska kvantitativního, kdy pomocí vybraných indexů popíšeme konkrétní vlastnosti zkoumaných souborů. V následující kapitole *Sémantika v datasetech* je provedena analýza sémantiky nejfrekventovanějších hesel, na základě které se pokusíme identifikovat jejich tematickou povahu. Zbylé části této práce se pokusí odpovědět na otázku *co činí hesla unikátními*. V kapitole *Unikátní hesla* tak nejprve nahlédneme na to, kolik procent jednotlivých datasetů zabírají hesla, která mají pouze jednoho uživatele a dále se zaměříme na jejich kvantitativní popis. Jedním ze sledovaných indexů je jejich délka, a tedy je provedeno srovnání délek unikátních a neunikátních hesel. Dále se budeme zabývat tím, jakou roli v nich hrají jednotlivé množiny znaků (tj. malá/velká písmena, čísla, speciální znaky) a prozkoumáme možnost toho, zda jsou unikátní hesla vskutku unikátní řetězce znaků, nebo zda se jedná o modifikace populárních typů. U množin užitých znaků setrváme i v následující kapitole této práce: *Sylabická struktura hesel*. Tentokrát však budou sledované množiny motivované jazykově: pokusíme se popsat, zda hesla typicky sdílejí konkrétní vzory rozložení konsonantů a vokálů. V závěrečné kapitole *Sylabické struktury a prolamování hašů*

hesel prozkoumáme možnost získávání hesel z jejich kryptograficky chráněné podoby způsobem, který těží ze zde zjištěných lingvistických poznatků.

Kromě samotného textu jsou poskytnuty čtyři přílohy, které čtenáři nabízí hlubší vhled do vybraných problematik. Tyto analýzy nebyly zařazeny do kapitol práce pro zachování konzistence textu – jedná se např. o téma numerických sufixů hesel nebo procentuální zastoupení nízko frekventovaných hesel v rámci souborů. Veškeré analýzy provedené v této práci byly provedeny pomocí programovacího jazyka R; kód užitý ke zpracování a analýzám hesel lze nalézt na přiloženém médiu.

Než se přesuneme k náplni této práce, je zde na místě uvést, že veškerá data a soubory hesel, které byly pro potřeby této práce využity, pochází z veřejně dostupných zdrojů; jsou to data, která jsou volně ke stažení. Ve snaze o zachování anonymity byla pro tři z analyzovaných souborů hesel zvolena zástupná jména; zároveň, pokud se názvy platforem objevily v heslech jako takových, byly anonymizovány.

(Korpusová) lingvistika a její relevance pro kybernetickou bezpečnost

Pro ukotvení lingvistického kontextu se úvodem této práce ve stručnosti zaměřme na to, jaké podobory lingvistiky budou pro tuto práci relevantní. Budeme zde dále využívat metody, které jsou spojeny především s oborem korpusové lingvistiky, popř. také obecněji se zpracováním přirozeného jazyka (známého také jako NLP, *natural language processing*).

Korpusová lingvistika je jedním z mladších podoborů obecné lingvistiky. Ačkoli její počátky sahají do druhé poloviny dvacátého století, největší rozvoj zaznamenává v posledních letech pod záštitou komputační lingvistiky. Je to podobor stojící na průniku lingvistiky, informatiky a statistiky a objektem jejího zkoumání jsou úhrny textových dat velkých rozměrů (tzv. korpusy). (Čermák, 2017) Oproti předchozím metodám zkoumání jazyka má tu výhodu, že díky množství dat, se kterým nakládá, odráží současný jazyk takový, jaký je. Vytěžováním textových dat tedy nabíráme poznatky o reálném užívání jazykových jednotek v nejrůznějších kontextech. Jak Čermák (2017: 10) ilustrativně uvádí, v minulosti trpěli lingvisté jistou „informační podvýživou", jelikož neměli přístup k velkým vzorkům jazykových dat a obecně bylo jejich zpracování až mravenčí prací. Právě to je situace, do které vstupuje korpusová lingvistika, která lingvistům poskytuje onu „informační výživu" a nové metody zkoumání.

Korpusová lingvistika svým zaměřením poskytuje metody vhodné pro zkoumání jazyka nejen z hlediska akademického, ale umožňuje exaktní přístup k jakýmkoli objemným datům textové povahy. Právě zde se nachází její průnik s oborem kybernetické bezpečnosti. V této práci se budeme dále zabývat uplatněním jejích metod (či obecněji metod kvantitativní lingvistiky, popř. NLP) na soubory hesel. Motivace pro tento typ výzkumu je následující. Vycházíme z opodstatněného předpokladu, že hesla, které si uživatelé nejrůznějších platforem volí, jsou z velké části jazykové povahy: slova, fráze, jména – srov. např. seznam nejčastějších hesel publikovaný manažerem hesel NordPass (2021 [cit. 10-08-2021]). Obecně očekáváme, že některá hesla, vzory či formy budou více preferované než jiné. Lingvistika zde jakožto interdisciplinární obor může nabídnout nový, jazykový, úhel pohledu na tuto problematiku a potenciálně svými poznatky přispět metodice bezpečnosti v digitální době.

Terminologie a technické poznámky

Než se přesuneme k samotným analýzám souborů hesel, pozastavíme se nad terminologií, která bude dále v této práci užívána. Jedná se o termíny z oblasti korpusové lingvistiky, datové analýzy a kybernetické bezpečnosti; dále se krátce pozastavíme také nad nezbytnými technologickými poznámkami týkající se hesel a způsobu jejich uložení v databázích.

— Dataset
Dataset je pojem z datové analýzy (a podobných/přidružených oborů). Jedná se o anglicismus označující soubor dat. Zde jej užíváme pro označení analyzovaných souborů hesel.

— Token
Token je termín z oblasti korpusové lingvistiky, kterým se myslí každý jednotlivý výskyt slova (popř. jiné definované jednotky). Pokud máme text o délce dvaceti slov, jedná se o dvacet tokenů – nezávisle na tom, zda se některé z daných slov opakuje dvakrát, třikrát či vícekrát. V této práci budeme jako token označovat, obdobně jako v korpusové lingvistice, každý jednotlivý výskyt hesla. Pokud pracujeme s datasetem obsahujícím 100 000 hesel, je každé z nich individuálním tokenem.

— Typ
Typ je termín úzce svázaný s tokeny. Pokud token je každý jednotlivý výskyt slova v textu, poté typ je každý jeden tvar, který nabírá frekvenci dle počtu výskytu k němu se vážících tokenů. Např. pokud máme soubor znaků (tokenů) A, B, B, C, D, D, D, pak jejich typy jsou A, B, C, D s frekvencemi 1, 2, 1, 3. V této práci budeme tedy jako typy označovat soubory hesel, v nichž je každé heslo zastoupeno pouze jedenkrát.

— Hapax legomenon
Termínem hapax legomenon (pl. hapax legomena, zkráceně také *hapaxy*) označujeme slovo, které má v textu nízkou frekvenci. Obecně existují dvě pojetí: první z nich definuje hapaxy jako slova s frekvencí 1 nebo 2, druhé z nich definuje hapaxy striktně jako slova s frekvencí 1. V této práci se přikloníme ke druhému pojetí a termínem hapax legomena budeme označovat ta hesla, která mají v datasetu pouze jedno zastoupení.

***

- Pokud budeme dále v textu uvádět příklady jednotlivých hesel, budou vyznačena pomocí symbolů < > (např. <heslo>).
- Pokud užíváme označení speciální znaky, máme tím na mysli vše, co bude vyhledáno regulárním výrazem [^a–zA–Z0–9], tedy mezera a !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

Hesla: uvedení do kontextu

Hesla jsou nedílnou součástí online aktivity každého uživatele. Stejně jako klíč umožňuje přístup do domu či bytu a chrání náš majetek, tak i heslo umožňuje přístup do účtu, emailové schránky nebo profilu na sociální síti a chrání naše osobní údaje a data. S neustálým technologickým vývojem ve dnešním světě se lidská aktivita čím dál více přesouvá do digitálního světa: online komunikujeme s přáteli či s kolegy, spravujeme bankovní účty, nakupujeme nebo i pracujeme. Objem informací, který takto uživatelé online uchovávají (a kolik o nich uchovávají jednotlivé služby), je veliký, a není tedy divu, že právě hesla jsou často objektem zájmu útočníků na různé platformy, služby, aplikace atp. Motivace útočníků se různí, může se jednat např. o krádež za účelem zpeněžení daných informací nebo i jiné, nekomerční, důvody, jako je krádež identity, špionáž či vydírání. Neplatí také, že útočník se do cizího účtu dostává pouze tím, že by se pokoušel heslo uhodnout na základě určitých indicií (jako např. osobní informace majitele účtu jako je datum narození, u žen rodné příjmení atp.) – existuje řada nástrojů a metod, jak získat přístup k (i chráněným) informacím uživatele určité služby; této problematice se budeme věnovat dále.

V následující části nejprve shrneme některé aspekty základní uživatelské praxe, co se tvorby hesel týká, dále nastíníme problematiku síly hesla a ve stručnosti popíšeme základní principy útoků na soubory chráněných hesel. Mějme také na paměti, že pokud se zde bavíme o heslech, myslíme primárně ta, která si uživatelé tvoří do online účtů a aplikací (jako např. email, online služby aj.), není zde brán v potaz kupříkladu PIN, přístupová hesla do mobilních telefonů atp. (ačkoli i na ně by se mohly některé poznatky této práce také vztahovat).

— Tvorba hesla

V této práci vycházíme z předpokladu, že uživatelé si svá hesla volí sami a není jim žádné automaticky přiděleno. Ačkoli neexistuje obecný návod na tvorbu bezpečného hesla, lze najít jistou shodu v obecných doporučeních, která jsou poskytnuta různými platformami či orgány. *Národní úřad pro kybernetickou a informační bezpečnost* (NÚKIB), správní úřad zastřešující kybernetickou bezpečnost a kryptografickou ochranu, kupříkladu doporučuje využít kreativity a originality: heslo by mělo obsahovat ideálně dvanáct a více znaků, které budou kombinovat velká a malá písmena, čísla a speciální znaky, např. ve formě věty či souvětí (NÚKIB, 2019). Co se týká platforem, kde si uživatelé běžně zakládají účty a vytváří k nim hesla, množství z nich klade na zakladatele účtu požadavky co do minimálního počtu znaků, užití velkých i malých písmen, čísel, popřípadě i speciálních znaků: kupříkladu *Gmail*, emailová služba poskytovaná společností Google, vyžaduje, aby heslo mělo alespoň osm znaků a využívalo písmena, číslice a symboly; *Netflix*, služba poskytující online sledování filmů a seriálů, oproti tomu vyžaduje alespoň čtyři libovolné znaky, ačkoli k tomu uvádí, že nejlépe by heslo mělo mít

alespoň osm znaků a zahrnovat jak velká tak malá písmena, symboly i čísla.[1] Tyto nastíněné doporučení a požadavky cílí k tomu, aby heslo bylo co možná nejméně uhodnutelné – a to nejen *člověkem*, ale především *počítačem*. Na jedné straně je nutno účet chránit před potenciálním útočníkem snažícím se uhodnout heslo na základě indicií (jako jsou osobní údaje, důležitá data jako např. datum narození atp.), na straně druhé pak před útočníkem, který se zaměřuje na větší soubor chráněných hesel a disponuje speciálními nástroji pro jejich rozkrytí – což je v tomto ohledu poněkud problematičtější. V posledních letech se stále častěji setkáváme s tím, že jsou kradeny celé databáze různých služeb, které obsahují uživatelské údaje (a popř. i jiná data jako jsou čísla kreditních karet, soukromé konverzace atp.) a hesla, která mohou být chráněná či nechráněná (tématu ochrany hesel se budeme věnovat podrobněji na následujících stranách), přičemž snahy o získání hesel chráněných uvnitř databáze některou z kryptografických metod daly vzniknout řadě metod a nástrojů (jako jsou speciální počítačové programy), jak daná hesla získat. Právě před útoky na kryptografickou ochranu hesel nejlépe obstojí ta hesla, která jsou co možná nejvíce náhodná: ideálně by tak heslo mělo být co možná nejdelší sekvencí znaků kombinující velká i malá písmena, čísla a speciální znaky a být co možná nejméně předvídatelné, tedy se nedržet nějakých vzorů atp. – vzory totiž často podléhají jisté konvenci mezi uživateli, která pak útočníkovi usnadňuje práci, jelikož může využít předpokladu, že např. uživatelé preferují některé formáty hesel častěji než jiné (např. přidání čísla typicky na konec hesla). Náhodná hesla, ačkoli nesou výhodu vyšší bezpečnosti, však nejsou uživatelsky přívětivá, a to primárně z hlediska zapamatovatelnosti.

Zapamatovatelnost je pragmatický aspekt tvorby hesla, u kterého můžeme předpokládat, že figuruje na úkor bezpečnosti. Je však pochopitelný – dnešní běžný uživatel má až desítky (ne-li více) hesel, mnoho z nichž používá na denní bázi. Je pro něj tedy logicky pohodlnější tvořit hesla, která se snadno pamatují. Jak uvidíme na následujících stránkách, často se jedná o jednoduché numerické postupky či kombinace, jména, názvy platforem, kam uživatel heslo zadává, popřípadě také o kulturní odkazy jako např. přidání *007* na konec hesla (dle fiktivní postavy agenta Jamese Bonda) atp. Co mají takto tvořená hesla společné, je to, že jsou do jisté míry konvenční, vycházejí z lexika a jsou vázána k sociokulturnímu kontextu. Tento trend však bohužel nenahrává bezpečnosti v kyberprostoru – spíše naopak. Takto tvořená hesla jsou poté slabší kvůli jejich konvenční povaze, a tedy náchylnější k prolomení útočníkem. Než se přesuneme k problematice útoků, zaměřme se na to, co je to síla hesla a jak je možno ji spočítat.

— Síla hesla

Obecně se silou hesla myslí jeho odolnost vůči útokům – např. počet pokusů, které musí útočník provést, aby se mu podařilo dané heslo uhodnout v případě systematického hádání prováděného pomocí speciálních počítačových programů, které mohou brát v potaz i další ochranu hesel jako např. šifrování, hašování atp. (více k ochraně hesel v databázích bude uvedeno na následujících stránkách) (Pavlíček, Sedláček, 2020). Hesla lze z hlediska síly rozdělit do dvou kategorií: (I) hesla náhodná, čímž se rozumí hesla náhodně generovaná např. manažerem hesel nebo,

[1] Uvedené informace jsou platné k 20. 5. 2021.

obecněji, mechanismem generující náhodné řetězce. Ty nesou výhodu toho, že neužívají žádný *vzor* – s jejich rostoucí délkou se snižuje pravděpodobnost, že budou sdílena více uživateli a zároveň toho, že jejich jednotlivé znaky jsou na sobě nezávislé, a tedy je těžší uhodnout jejich souslednost. Další kategorií jsou (II) hesla předvídatelná, která jistý *vzor* užívají – do této kategorie lze zařadit většinu uživatelsky vytvořených hesel, při jejichž tvorbě hraje roli výše diskutovaná zapamatovatelnost. Předvídatelností se zde myslí to, že užití znaků v hesle je na sobě závislé, řadíme sem tedy např. lexikum, kdy následující znak se odvíjí od znaků, které mu předchází, číselné postupky nebo triviální kombinace – nejedná se tím pádem o sekvenci *náhodnou*, ale *předvídatelnou* (či *pseudonáhodnou*). V neposlední řadě je faktorem síly hesla také jeho délka a množina znaků, které jsou v něm obsaženy. (Pavlíček, Sedláček, 2020)

Síla hesla je dnes na mnoha online službách regulována ze strany poskytovatele: při tvorbě nového hesla se uživateli typicky zobrazuje indikátor, který zvolené heslo hodnotí např. na škále variující od *slabé* po *silné* (např. při zakládání emailového účtu na *seznam.cz* jsou hesla o délce < 6 znaků označována jako příliš krátká, škála je poté vyznačena barevně[2]). Obsáhlý výzkum populárních online služeb byl proveden (de Carné de Carnavalet, Mannan, 2014), kde byly zkoumány regulační principy tvorby hesel při zakládání nových účtů. Existují dvě hlavní metody, které jsou k této regulaci využívány: (I) požadování určité délky a variability co do znaků/čísel/spec. znaků; (II) detekce vzorů jako jsou slova či jednoduché klávesnicové či číselné sekvence. Často ovšem není metodika ověření síly hesla transparentní – jinými slovy nevíme, co jsou daná kritéria silného hesla. Tento výzkum poukázal mimo jiné na to, že v tomto ohledu existují jisté nesrovnalosti. Příkladem uvádějí např. hesla <Password1> nebo <football#1>, která byla jednou platformou označena jako *velmi silná* (pravděpodobně z důvodu užití více množin znaků: malá/velká písmena, čísla, popř. symbol), kdežto jinde se vyskytla na opačném konci škály jako *velmi slabá*. (de Carné de Carnavalet, Mannan, 2014)

Jednou z obecných měr síly hesla je entropie. Entropií se v nejobecnějším smyslu myslí *míra nahodilosti* či *nepředvídatelnosti*; její jednotkou je bit. Entropie je pojem, který byl zavedený původně ve fyzice, kde popisuje míru nahodilosti složek fyzikálního systému, jinými slovy také nespořádanost systému (Krámský, 1959) a dále byla Claudem Shannonem (1951) přejata do teorie informace. V případě hesel tak platí, že čím vyšší entropie, tím vyšší míra nepředvídatelnosti, což značí silnější heslo. Avšak pro každou z výše zmíněných skupin, tedy hesla *náhodná* a hesla *předvídatelná*, počítáme entropii jiným způsobem, jelikož povaha těchto hesel se co do náhodnosti zásadně liší. Zatímco v prvním případě je heslo generováno pomocí náhody z množiny daných prvků, a tedy jsou jeho jednotlivé znaky na sobě nezávislé, ve druhém případě existuje závislost jednoho znaku na druhém. Pro skupinu (I), hesla tvořená počítačem, entropii (H) počítáme pomocí logaritmu o základu 2 z počtu možností kombinací vzorcem:

$$H = log_2(b^l) = l\ log_2\ b$$

[2] Platné k 24. 7. 2021

kde l je počet náhodně vybraných symbolů z množiny o velikosti b. (Burr et al., 2013) Ilustrujme si tento výpočet na příkladu: mějme heslo tvořené náhodně vybranými malými písmeny ($b = 26$) o délce osm znaků ($l = 8$). Dosadíme-li do vzorce:

$$H = 8 \times log_2 26 = 8 \times 4{,}70 = 37{,}6$$

získáváme hodnotu entropie: 37,6 bitů. Tato hodnota je nejen ukazatelem síly daného hesla, ale také nám říká, že pokud by se útočník snažil prolomit toto heslo hrubou silou, musel by pro ověření všech možností spočítat v krajním případě, kdy je hádané heslo poslední možná kombinace, až $2^{37{,}6}$ (= 208 318 498 661) hašů (Pavlíček, Sedláček, 2020). Takto vyčíslitelná entropie je jednou z výhod náhodně generovaných hesel, jelikož na jejím základě jsme schopni určit sílu hesla a tím determinovat jeho odolnost vůči útokům. Další výhodou je to, že tato hesla nebudou podléhat vlivu uživatelské preference a tím se stávat náchylnější k prolomení (pro diskuzi k tomuto viz následující část *Typy útoků na databáze*).

Oproti tomu stojí druhý typ hesel, která jsou nenáhodná. Jak již bylo zmíněno výše, jedná se o hesla, která si uživatelé tvoří sami, a jejichž jednotlivé prvky jsou na sobě závislé – lexikum. Uvažme například písmeno *b* v češtině: to, že po něm bude následovat *a*, je výrazně pravděpodobnější, než že za ním bude následovat např. *q* nebo *x*. V porovnání s předešlým typem hesel se tak jedná o výrazný rozdíl, jelikož tato závislost není změřitelná výše uvedenou entropií. Tento jev závislosti znaků byl zkoumán Claudem Shannonem (1951), který takto analyzoval předvídatelnost a entropii anglického jazyka. Zjistil, že první písmeno nese největší váhu informace, jelikož je nejtěžší jej uhodnout. Další písmena jsou již jednodušší díky danému kontextu. A vzhledem k tomu, že se u uživatelských hesel očekává vysoké zastoupení lexika, dal Shannonův výzkum vzniknout metodě určování entropie tohoto typu hesel. Co však musí být bráno v potaz je to, že v tomto případě se nebavíme o exaktním výpočtu entropie, ale spíše o jejím *odhadu*. Americký *Národní institut standardů a technologie* (NIST) ze Shannonových poznatků vycházel při návrhu následujícího postupu pro odhadnutí míry entropie uživatelských hesel v publikaci NIST 800-63-2 (Burr et al., 2013).

Tento postup by pak měl být aplikován v případě služeb, kam si uživatelé vytváří heslo sami, jelikož se u nich předpokládá preference lexika a obecně jazykových rysů, např. užívání velkého písmene primárně na začátku a speciálních symbolů (po vzoru interpunkce) na konci hesla. Odhad entropie pro takováto hesla je v metodice NIST prováděn následujícím způsobem:

1. entropie prvního znaku jsou 4 bity,
2. entropie dalších sedmi znaků jsou 2 bity/znak,
3. entropie devátého až dvacátého znaku je 1,5 bitu/znak,
4. entropie dvacátého prvého a dalších znaků je 1 bit/znak,
5. přičte se dalších 6 bitů entropie heslům, která obsahují jak velká písmena, tak i čísla/symboly,
6. přičte se dalších 6 bitů entropie heslům, která se nenachází ve slovníku. (Burr et al., 2013: 107)

Ilustrujme si tento způsob na příkladu: mějme heslo o délce 11 znaků, které je spojením dvou slov obsahujících velká i malá písmena, doplněných o jednu číslici

– ilustrativně uveďme *SlovoSlovo[číslo]*. Odhad entropie (*H*) spočítáme dle výše uvedeného schématu jako:

znak	S	l	o	v	o	S	l	o	v	o	[číslo]
H	4	2	2	2	2	2	2	2	1,5	1,5	1,5

Výsledná hodnota entropie pro toto heslo je dle jeho délky 21 bitů. Ověříme-li, zda vyhovuje následujícím kritériím 5 a 6, přičteme další, „bonusové“, bity:

H znaků dle délky	21
Obsahuje velká písmena + čísla/symboly	+ 6
Nenachází se ve slovníku	+ 6
Výsledná *H*	33

čímž získáváme výsledný odhad entropie: 33 bitů. Srovnejme tento *odhad* entropie s hodnotou, kterou bychom vypočítali způsobem pro náhodná hesla; množina *N* zde obsahuje 62 znaků (26 malých a 26 velkých písmen, 10 číslic), délka hesla *L* je zde 11; dosadíme-li do vzorce:

$$H = 11 \times log_2 62 = 11 \times 5{,}95 = 65{,}45$$

získáváme 65,45 bitů – tedy téměř dvojnásobně vyšší hodnotu. Ta by však platila v případě, že by heslo bylo tvořeno náhodně, a tedy by přítomnost určitého znaku nedeterminovala pravděpodobnost následujících znaků. V námi zkoumaném hesle však každá pozice vytváří úhrn znaků, které po ní budou s největší pravděpodobností následovat, a tedy i výsledný odhad entropie je nižší, jelikož obsazení jednotlivých pozic nevzniklo vlivem náhody.

Co je nutno zmínit obecně k nenáhodnému typu hesel, je to, že ačkoli jsou uživatelsky pohodlná tím, že nekladou vysoké nároky na zapamatovatelnost, jsou problematická nejen z hlediska entropie, kterou nelze exaktně vyčíslit (jak jsme zde mohli vidět, v tomto ohledu se nedostaneme o moc blíže než k hrubému odhadu), ale i v jejich četnosti užití různými uživateli – tomuto tématu se však budeme věnovat v další části.

Závěrem části věnující se síle hesel poznamenejme k entropii to, že z hlediska síly a bezpečnosti hesla se jedná spíše o indikátor pro metodiku tvorby hesla. Požadavky kladené platformami (např. že heslo musí mít určitou minimální délku, musí obsahovat malá i velká písmena, číslici atp.) cílí na to, aby uživatel zvolil co možná největší variabilitu znaků, čímž míří právě na zvýšení entropie a síly hesla. Obrátíme-li však úhel pohledu na druhou stranu a nahlédneme problematiku prolamování hesel z hlediska útočníka, do prolamování hesel vstupuje více faktorů, než je jen jeho entropie. Uveďme příklad hesla, sekvenci *C-3PO-R2-D2*. Ačkoli by se na první pohled mohlo zdát, že se jedná o poměrně silné heslo – obsahuje písmena, číslice i symbol – ve skutečnosti je možné, že by bylo útočníkem prolomeno poměrně rychle, jelikož se jedná o postavy populární sci-fi série *Star Wars*.

V případě, že by se toto heslo právě z důvodu své vazby na popkulturu ocitlo ve slovníku užitém k útoku na databázi, stává se příliš jednoduchým a jeho entropie takřka bezcennou. Než se zaměříme na typologii útoků na hesla, nastíníme to, jak a kde jsou hesla ukládána.

— Způsoby uložení hesla v databázích

Hesla lze v databázích ukládat mnoha způsoby. Pokud se chce uživatel přihlásit do svého účtu, vedle uživatelského jména/emailu (či jiné identifikační položky) zadává své heslo. Daná platforma uživatelovy přihlašovací údaje srovná s těmi, které se vyskytují v databázi, a pokud jsou zadány správně, je uživateli povolen přístup. S heslem se v tuto chvíli může dít vícero věcí: může být srovnáno s údaji v databázi tak, jak je – tedy jako řetězec znaků, avšak to dnes již není běžná praxe. Mnohem častěji je po zadání uživatelem heslo zašifrováno či jinak převedeno na jinou podobu a až poté srovnáno s šifrou či danou podobou v databázi. Je tomu tak primárně pro ochranu uživatele. V případě úniku dat či krádeže databáze se takto útočníkovi do rukou nedostávají hesla jako taková, ale jejich šifrovaná či jinak chráněná podoba. V následující části se tedy zaměříme na způsoby, kterými jsou hesla v databázi uložena, přičemž popíšeme ty základní, s většinou z nichž se zde dále setkáme.

1) Plaintext

Plaintext je anglicismus, který lze volně přeložit jako „prostý text“ či „čistý text“. Pokud bude na následujících stránkách uvedeno, že heslo bylo v databázi ukládáno ve formě plaintextu (popř. plaintextové podobě), znamená to, že nebylo zašifrované či jinak chráněné: bylo uloženo „tak, jak je“ – jako prostý text. To znamená, že kdokoliv s přístupem do databáze vidí znění hesla. Pokud tedy ukradená data obsahovala hesla v plaintextu, znamená to, že máme přístup ke všem heslům bez jakékoliv další práce jako je dešifrování či prolamování – k tomuto však později.

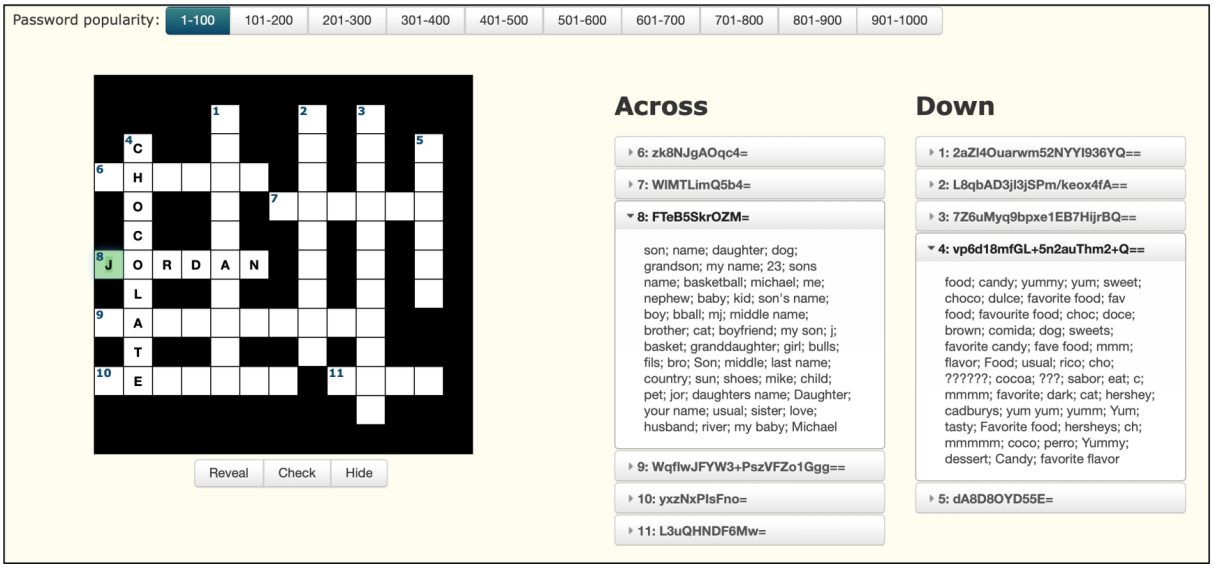
Co se týká analýz a výzkumu týkající se běžné uživatelské praxe tvorby hesel, práce s hesly v plaintextu je jedním ze základních předpokladů. Díky tomu, že je není třeba dešifrovat či prolamovat, nevystavujeme se riziku toho, že k analýze z takto chráněných hesel získáme jen část z nich (obecně tu jednodušší část), a že tak výsledky analýzy nebudou reprezentativní pro uživatelskou praxi dané platformy či aplikace.

2) Šifra

Některé platformy volí cestu ochrany hesla jeho zašifrováním. To znamená, že na vytvořená hesla je aplikována určitá šifrovací funkce a dále jsou hesla v databázi ukládána v šifrované podobě. Klíčovou vlastností šifry je její *obousměrnost*, tedy to, že pokud známe klíč šifry, dokážeme získat původní heslo (oproti tzv. haši, který je jednosměrný – více k hašovacím funkcím viz další část). V tom také leží jejich zásadní slabina, jelikož pokud útočníkovi padne do rukou klíč k šifrovací funkci, dokáže jím rozkrýt všechna hesla, na které tato funkce byla aplikována; dále je zde také problém soukromí, jelikož k takto uloženým heslům má přístup i např. administrátor serveru. Kromě toho také nevhodně zvolený šifrovací algoritmus může způsobit čitelnost metainformací o heslech, jako je např. jejich délka, frekvence, rozdělení na podřetězce atp. Příkladem tohoto může být využití blokové

šifry, kterou použila firma Adobe k ochraně uživatelských hesel – tomuto případu se dopodrobna věnujeme níže.

Šifrování hesel je problematika, která sahá za hranice tématu této práce – zde budeme dále pracovat s hesly v plaintextu a později s haši. Pro bližší ilustraci slabin šifer však uveďme případ, který se stal firmě Adobe v roce 2013, kdy došlo k úniku dat 153 000 000 uživatelů. Tato data zahrnovala uživatelská jména a interní identifikaci, emailové adresy, šifrovaná hesla a nápovědy, které si sami uživatelé k heslům vytvořili. Obecně měl tento únik tři zásadní problémy: (I) všechna stejná hesla sdílela šifrovou podobu, (II) nápovědy byly uloženy v plaintextu, (III) k šifrování byla užita tzv. bloková šifra, tedy dlouhá hesla byla rozdělena na bloky po stejném počtu písmen. To může vést k následující situaci: pokud by se chtěl útočník dostat do libovolného účtu, nemusel by dešifrovat heslo – stačilo by se mu podívat na nápovědu (která jako plaintext nebyla chráněná) a pokud by mu tato nápověda nestačila, stačilo by najít onen šifrový kód u jiného uživatele a zjistit jeho nápovědu, popřípadě pokračovat k dalšímu atd.; případně využít povahy blokové šifry a hledat jisté sekvence a jejich nápovědy. Hádání na základě indicií se takto stává až hrou se slovy – jak metaforicky, tak i doslova. Nedlouho po úniku dat byla, jako vtip, vytvořena online křížovka, která využívá právě uniklých dat z Adobe a nápověd uživatelů.



_obr_Adobe (dostupné z: https://zed0.co.uk/crossword/ [cit. 12-07-2021])

Na *_obr_Adobe* vidíme výše zmíněnou křížovku. Čísla 1–11 se odkazují k šifrovým kódům hesel; po rozkliknutí jednotlivých čísel se rozbalí nápovědy – což jsou právě ty uživatelské, které unikly v plaintextu. Např. pod číslem 8 a šifrovým kódem *FTeB5SkrOZM=* se dle nápověd *name*, *Michael*, *bulls* ad. skrývá jméno basketbalisty Jordana. Ačkoli se tato forma využití úniku dat může zdát jako pouhá kratochvíle, autor křížovky na dané webové stránce (https://zed0.co.uk/crossword/ [cit. 12-07-2021]) uvádí jako další motivaci demonstraci toho, že běžná uživatelská hesla velmi často nejsou bezpečná, a také toho, že by uživatelé neměli vkládat stoprocentní důvěru ani tak velkým společnostem jako je Adobe v to, že jejich data uchová v bezpečí. Únik dat z Adobe je takto dobrým příkladem důležitosti

bezpečného chování v online prostoru, co se hesel a čímkoli s nimi spojeného týká, jelikož si nikdy nemůžeme být jisti, *kdy* a *jak* tato data uniknou.

3) Haš (anglicky *hash*)

Kryptografická hašovací funkce se užívá mimo jiné[3] k *ochraně dat*, v našem případě jsou její pomocí chráněna hesla v databázi. Vstupem této funkce jsou data libovolné délky, přičemž na jejím výstupu je unikátní sekvence znaků, jejíž délka je pevně daná (tzv. hašovací kód); jinými slovy: nezáleží, zda je vstupem např. řetězec písmen o délce tří nebo třiceti znaků – v obou případech bude haš stejně dlouhý. (Klíma, 2005)

Ideální hašovací funkce mají tři základní vlastnosti. První z nich je *jednosměrnost*, což značí, že z haše nelze výpočetně odvodit původní data; pokud tedy máme heslo převedené na haš, neměla by existovat možnost odvodit zpět jeho původní podobu; uložení hesla ve formě haše by tak mělo být nevratné. Toto také zaručuje, že pouze a jen uživatel zná své heslo, jelikož dále v databázi jsou tato hesla ukládána právě jako haš – při útoku na platformu a úniku informací se pak útočníkovi do rukou dostávají nikoli hesla, ale právě haše. Druhou základní vlastností je *bezkoliznost*, což je v zásadě to, že pro odlišná data nebude vycházet stejný haš – v našem případě, že odlišná hesla nebudou sdílet haš (a tedy by bylo možné se např. do jednoho účtu přihlásit oběma hesly). (Klíma, 2005) Třetí vlastností je to, že stejná data budou mít vždy stejný haš a provedeme-li jen drobnou změnu, haš se změní tak, aby se nedala odvodit podobnost či blízkost původních dat.

Zaměřme se nyní blíže na bezkoliznost. Jak Klíma (2005) uvádí, naprostá bezkoliznost je nemožná, jelikož existuje výrazný nepoměr mezi prakticky neomezenou množinou všech možných zpráv/dat (popř. hesel, mějme však na paměti, že hašovací funkce jsou aplikovatelné na široké spektrum dat) a omezenou množinou hašovacích kódů (u hašovacího algoritmu MD5 se např. jedná o 2^{128} hašů) – nejde tedy o to, aby kolize byla nemožná, ale o to, aby nebyla v rámci výpočetních možností útočníka. Případ, kdy narazíme na dva stejné haše, které zastupují jiná data, označujeme jako kolize; pokud se jedná o více hašů, pak užíváme označení multikolize či r-násobná kolize. (Klíma, 2005) V praxi by ideálně ke kolizím docházet nemělo, avšak opak je pravdou. Možnost výpočtu pravděpodobnosti nalezení kolize lze ilustrovat pomocí tzv. narozeninového paradoxu, který si zde ve stručnosti představíme. Narozeninový paradox se zabývá otázkou, *kolik lidí je třeba náhodně vybrat z populace, abychom mohli s pravděpodobností vyšší než 50 % říct, že alespoň dva z nich mají narozeniny ve stejný den*, přičemž dokazuje, že aby byla pravděpodobnost větší než 50 %, stačí takto náhodně vybrat pouhých 23 lidí – což se na první pohled zdá být v kontradikci s intuicí, dle které bychom očekávali vyšší počet. Pro výpočet jsou stanovena následující východiska: (I) nezapočítáváme přestupné roky, bereme v potaz standardních 365 dnů, (II) pro každý den v roce je stejná pravděpodobnost toho, že někdo bude mít narozeniny (tedy že neexistují skryté vzory jako např. dvojčata či preference určitých měsíců oproti jiným atp.).

[3] Lze je použít také např. pro komparaci souborů či dat větších rozměrů: použijeme-li na ně hašovací funkci, získáváme řetězce, které je možné srovnat, a tedy určit, zda jsou soubory dat stejné, nebo se (byť jen trochu) liší.

Podstata přístupu k tomuto problému tkví v tom, že spíše, než abychom počítali pravděpodobnost shody *P(A)*, která by byla obecně náročnější, vypočítáme pravděpodobnost toho, že dvojice mezi sebou narozeniny nesdílí *P(A')*, kterou následně odečteme od 1, čímž se dostaneme ke kýženému výsledku. Uvážíme-li výše zmíněných 23 lidí, výpočet bude vypadat následovně:

$$P(A') = \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \frac{362}{365} \times \frac{361}{365} \times \ldots \times \frac{343}{365} \approx 0{,}493$$

$$P(A) = 1 - 0{,}493$$

$$P(A) = 0{,}507 = 50{,}7\ \%.$$

P(A') je zde výpočtem pravděpodobnosti toho, že žádný z oněch 23 lidí nesdílí narozeniny s někým jiným. Pro prvního člověka je tato pravděpodobnost 365/365 = 1, pro druhého 364/365 ≈ 0,997 atd. až ke dvacátému třetímu člověku. Odečtením od jedné se dostáváme k opačnému problému, *P(A)*, a tedy získáváme odpověď na otázku *jaká je pravděpodobnost toho, že alespoň dva lidé z této skupiny sdílí datum narození*. Pokud nahlédneme obdobným prizmatem na databázi hašů, zjišťujeme, že z hlediska kolizí se nacházíme ve velmi podobné situaci.

Pro konkrétní hašovací funkci lze vypočítat, kolik hašových kódů je třeba vygenerovat, než bude pravděpodobnost výskytu kolize (tedy situace, kdy dvě různá hesla sdílejí jeden haš) větší než 50 %. Tento počet vypočítaných hašů je ve skutečnosti neintuitivně nižší, než je běžné očekávání. Vzhledem k omezenému počtu hašů, který je dán kombinatorikou, je určitým limitem hašovacích kódů, jelikož kolize mohou být potenciálně zneužity pro přístup do účtu uživatele. Narozeninový paradox dal mimo jiné vzniknout konceptu tzv. „narozeninového útoku“ (anglicky *birthday attack*), který využívá právě kolizí hašů. Samotný koncept tohoto útoku je však téma sahající mimo rámec této práce.

Jak tedy můžeme vidět, ideální vlastnosti hašů nejsou ve skutečnosti stoprocentní a nelze tvrdit, že haše uživatelská hesla uchovají v naprostém bezpečí. Útočníci, kteří se snaží nalézt původní hesla, mohou využívat různých metod, jak z hašů původní hesla získat (blíže k tomuto viz následující část *Typy útoků na soubory hašů*). Jako příklad jedné z těch jednodušších lze uvést frekvenční analýzu. Jak bylo zmíněno výše, haš je sice unikátní pro *heslo*, ale ne pro *uživatele*: jinými slovy, pokud dva lidé sdílejí heslo, sdílejí i haš. Takto lze haše seřadit dle frekvence od těch nejčastějších po ty, které se vyskytují pouze jednou a srovnat právě nejčastější výskyty s haši populárních hesel jako např. <password> nebo <123456>. Tímto postupem je potom možné odhalit část chráněných hesel. Pro silnější ochranu hašovaných hesel se proto dále využívá např. šifrování samotného haše, nebo přidání tzv. *soli* (anglicky *salt*), popř. *pepře* (ang. *pepper*).

Sůl je unikátní, ideálně náhodně generovaný řetězec znaků připojovaný k heslu (před, nebo i za něj) před hašováním. Sůl je zajištěním toho, že každé heslo je unikátní, více komplexní a, především, že i haš bude uživatelsky unikátní. Obecně sůl ztěžuje útočníkovu práci z hlediska výpočetní náročnosti útoku, jelikož pro solené

haše nelze využít uživatelské preference konkrétních hesel. Nevýhodou soli je to, že je veřejná – v databázi se ukládá vedle hašovaného hesla.

Pepř je svou podstatou podobný soli, taktéž se jedná o ideálně náhodně generovaný řetězec přidaný k heslu před hašováním. Hlavním rozdílem je zde to, že je tajný – neukládá se vedle hašovaného hesla v databázi. Bývá také pro všechna hesla stejný a používá se jako komplement soli – pokud by dvě stejná hesla nebyla před pepřením solená, sdílela by stejný haš. Je takto dalším možným stupněm ochrany, jelikož jeho hodnota je uložena zpravidla jinde, než jsou hašovaná hesla. Pokud se útočníkovi dostane do rukou databáze hašů se solí a neznámou hodnotou pepře, měla by hesla lépe obstát v útoku. Jak však Pavlíček a Sedláček (2020) uvádí, pepření nemusí být efektivní, pokud útočník dokáže společně s databází získat hodnotu pepře.

Hašovacích funkcí rozlišujeme mnoho typů, lze se setkat např. s *MD5*, *SHA-1, SHA-256* nebo *bcrypt*, kdy každý z nich je založen na jiném algoritmu výpočtu a je tak typologicky jiný (srov. *_tab_has-priklady*).

Hašovací funkce	**Haš hesla <password>**
MD5	5f4dcc3b5aa765d61d8327deb882cf99
SHA-1	5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8
SHA-256	5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8
bcrypt	$2y$12$m1QCiA9UCBUSAI7.GrimBeZIHvoeiJ1R1ZTmc64bmMSKRVRaN0rfa

_tab_has-priklady

Na první pohled se jedná o rozdíl v textové reprezentaci, kterou jednotlivé hašovací funkce typicky využívají. Zatímco v případě MD5, SHA-1 a SHA-256 se jedná o šestnáctkovou soustavu, v případě bcrypt je to base64. Dále haše rozdělujeme na *rychlé* (zde zastoupeny MD5, SHA1 a SHA256) a *pomalé* (jejichž zástupce je zde bcrypt). Rychlost/pomalost zde značí výpočetní náročnost haše – zatímco rychlé haše jsou poměrně jednoduché k výpočtu, pomalé haše jsou výpočetně náročnější, a tedy i doba jejich výpočtu je obecně delší. Pomalé haše jsou tak výhodnější pro ochranu uživatelských hesel, jelikož ztěžují práci útočníka, který by se potenciálně snažil o rozkrytí původní podoby hesla.

— Typy útoků na soubory hašů

Dostane-li se útočníkovi do rukou databáze hašů uživatelských hesel, má několik možností, jak z ní získat původní hesla – každá z těchto možností s sebou nese jisté výhody i nevýhody. V následující části popíšeme útoky zprostředkovatelné programem *hashcat*, a to primárně z toho důvodu, že tento program bude využit i pro potřeby experimentu popsaného dále v této práci. Zatímco některé z těchto typů útoků jsou standardní praxí (např. útok hrubou silou či slovníkový útok), jiné jsou zde popsány specificky pro daný program (např. útok pomocí masek).

Obecně tyto útoky fungují na principu toho, že program postupně tvoří haše různých textových řetězců, pro které testuje, zda se nachází v cílovém souboru hašů.

V případě, že nalezne stejný haš, je heslo „prolomeno“ (za předpokladu, že nedošlo ke kolizi) a uloženo do separátního souboru vedle příslušného haše.

1) Útok hrubou silou (anglicky *brute force attack*)

Útok hrubou silou funguje na principu zkoušení veškerých možných kombinací (variancí s opakováním) znaků z dané množiny – jsou jím tedy nejvíce ohrožena slabá hesla: typicky krátká slova či krátké sekvence znaků. Rozhodující vlastností je zde nízká délka hesla: typicky se uvádí, že útok hrubou silou má smysl zkoušet pro hesla o maximální délce 7–8 znaků (Pavlíček, Sedláček, 2020), což platí v případě rychlých hašovacích funkcí (jako je např. výše zmíněná MD5). Pokud je heslo dlouhé (tedy je tvořeno > 8 znaky), je útok hrubou silou značně neekonomický co do času a výpočetní kapacity počítače: uvážíme-li, že např. pro heslo o délce 12 znaků by bylo třeba ověřit všechny možné kombinace 94 možných znaků standardní klávesnice, dostáváme se do vysokých čísel – 94^{12} možných kombinací. Rychlost prolomení je pak závislé jak na typu haše, tak na výkonnosti počítače, který výpočet provádí, avšak se můžeme dostat až do řádu měsíců či let. Pro takováto hesla je pak výhodnější přistoupit k prolomení jinak, a to např. předem připraveným seznamem hesel – což je jinak známé jako tzv. *slovníkový útok*.

2) Slovníkový útok (anglicky *dictionary* nebo také *wordlist attack*)

Princip slovníkového útoku je poměrně intuitivní: útočník má předem připravený seznam potenciálních hesel (tzv. slovník, popř. se lze v české literatuře setkat i s anglicismem *wordlist*). Tento typ útoku počítá s jistou uživatelskou konvencí při tvorbě hesel vycházet ze známého lexika, popř. používat socio-kulturně vázané výrazy jako např. <R2-D2> (odkazující se k filmové sérii Star Wars), <X Æ A-12> (jméno syna Elona Muska) nebo <14159265> (prvních osm desetinných míst Ludolfova čísla). Na prvních příčkách se takto často objevují jména, názvy oblíbených týmů, notoricky známé heslo *password*; z čísel pak jednoduché postupky či sekvence, popř. kulturní odkazy jako je již zmíněné <007>. Slovníky obsahují až stovky milionů populárních hesel. Útočník si může sestavit slovník vlastní, nebo může využít některého z těch, které jsou online ke stažení. Mezi známé slovníky patří např. *RockYou*, který pochází z úniku dat stejnojmenné platformy a je tvořen více než 14 miliony hesel (a budeme se mu dále v této práci věnovat, jelikož je jedním z vybraných souborů pro analýzy), dále *Crackstation-human-only*, který je kompilací hesel z vícero webových databází a obsahuje přibližně 64 milionů hesel, *m3g9tr0n_Passwords_WordList_CLEANED*, který obsahuje až 122 milionů hesel, nebo nejnovější *RockYou2021* o rozsahu až 8,4 miliardy hesel. Kromě toho je možné tyto seznamy dále rozšířit přidáním pravidel definujících modifikace, které mají být na celém seznamu hesel provedeny – lze například přidat za kombinace znaků přiřadit letopočty (které jsou obecně častým jevem mezi hesly, srov. např. výsledky studie (Veras et al., 2012)) atp., což zvyšuje pravděpodobnost úspěchu útoku (Kim, 2015) – tento typ útoku je pak označován jako hybridní (anglicky *hybrid attack*).

Co je zajímavým aspektem slovníkového útoku je to, že do něj potenciálně vchází určitá pragmatika ze strany útočníka, a to vzhledem právě ke kulturním vazbám, či popřípadě jiných specifik dotýkajících se dané databáze. Pokud by se útočník zaměřoval na soubor hesel pocházející např. z českého prostředí, ideálně by měl

použít slovník, který tento fakt odráží – ačkoli lze najít jistou shodu v nejčastějších hesel bez ohledu na jejich geografický původ (jako jsou např. číselné postupky či triviální alfanumerické kombinace), v datasetu z českého prostředí lze očekávat značné zastoupení českých jmen, lexika a odkazů k „českému“ kontextu jako jsou např. názvy sportovních týmů, měst atp. Tento princip lze vztáhnout na větší množství socio-kulturních artefaktů – pokud tedy útočník plánuje útok na specifickou databázi, je ideální jí „ušít na míru“ i užitý slovník, a tedy jej sestavit z dat z českého kontextu.

3) Útok pomocí masek (anglicky *mask attack*)

Útok pomocí masek je v podstatě obměnou (či vylepšením) útoku hrubou silou. Maskou se v tomto smyslu rozumí šablona toho, jak má vypadat počítačem generovaný řetězec k testování: každá pozice v masce definuje množinu znaků, které se na dané pozici mohou v hesle vyskytnout. Tedy místo toho, aby se zkoušely všechny možné kombinace znaků tvořící řetězec určité délky, vyzkouší se jich pouze omezené množství, jelikož kombinatorika tvorby hesla již není na principu *každý znak s každým*, nýbrž na principu specifických možností výskytu. Pomocí těchto masek se dále generují hesla, která se srovnávají se zkoumaným souborem. Pro definici masky jsou v programu *hashcat* užívány následující symboly (tabulka *_tab_masky_symboly*) (mask_attack [hashcat wiki], [cit. 17-08-2021]):

Zástupné symboly v maskách	
symbol	**zastupuje**
?l	*lower case*, tedy malá písmena [a–z]
?u	*upper case*, tedy velká písmena [A–Z]
?d	*digit*, tedy číslice [0–9]
?s	*special symbols*, tedy speciální znaky jako např. interpunkce, zavináč aj.
?a	*all*, tedy všechny znaky předešlých skupin (?l, ?u, ?d, ?s)
?h	*hexa*, tedy soustava [0–9a–f]
?H	*HEXA*, tedy soustava [0–9A–F]
?b	*binary*, tedy veškeré hodnoty znaku (0x00–0xff)

_tab_masky_symboly

Mimo to existuje také možnost uživatelského definování až čtyř množin různých symbolů, a to pomocí parametrů 1–4; jinými slovy existuje možnost nevyužít předem definované zástupné symboly množin (které jsou uvedeny v tabulce *_tab_masky_symboly*), ale vytvořit si skupiny vlastní. Ostatně, této možnosti bude využito v rámci této práce, kdy se pokusíme prolamovat haše pomocí masek založených na lexikálních vlastnostech hesel – k tomuto však později. Co se dále týká masek, je možné je kombinovat přímo s konkrétními sekvencemi znaků, čímž lze hledat modifikace konkrétních hesel. Pro vizualizaci principu tvorby masek viz následující tabulku *_tab_masky-priklady*.

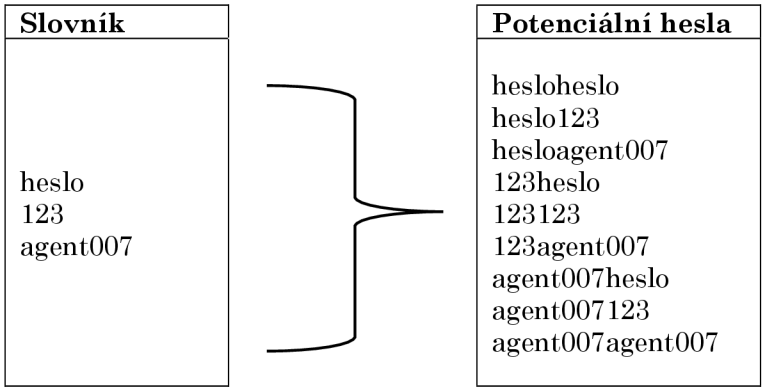
Maska	**Zastupuje hesla s vlastnostmi…**	**Hesla generovaná maskou**
?u?l?l?l?l?d?d?s	první písmeno velké, čtyři malé, dvě čísla, spec. znak	Karel01!
		Heslo33.
		Marta00*

heslo?d?d	slovo *heslo* doplněné dvěma číslicemi	heslo01
		heslo13
		heslo77
–1 aeiou –2 7890 ?1?1?2?2?2	tvořené dvěma uživatelsky def. skupinami	uu789
		ai007
		ia808

_tab_masky-priklady

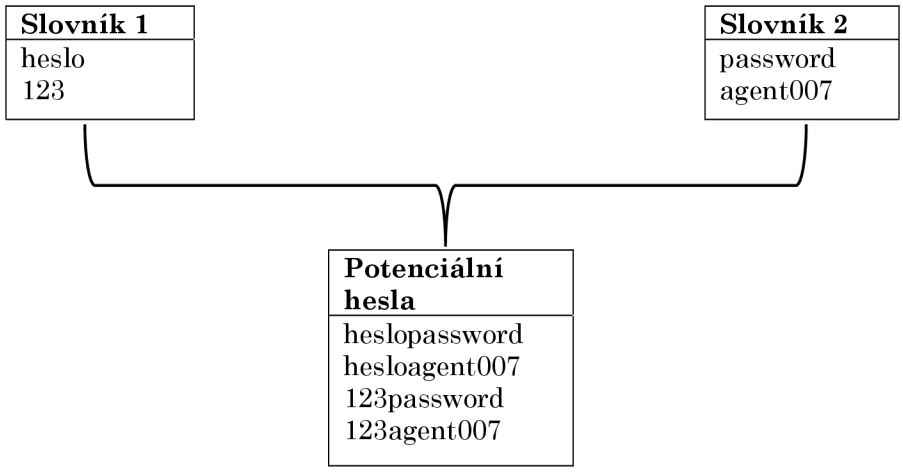
4) Kombinatorní útok (anglicky *combinator attack*)

Kombinatorní útok je typ útoku, který pracuje se slovníkem – tentokrát ale jiným způsobem než výše zmíněný slovníkový útok, a to tak, že provádí (I) kombinace veškerých hesel z dané množiny, nebo (II) konkatenace veškerých hesel ze dvou nezávislých množin (Hranický et al. 2018). Přístup (I) lze ilustrovat takto (viz *_obr_combinator_1*):



_obr_combinator_1

Vlevo zde vidíme námi zvolený (či přímo vytvořený) slovník, napravo poté veškeré kombinace z něj vytvořitelné. Přístup (II) oproti tomu provádí konkatenace dvou slovníků, lze jej ilustrovat takto (viz *_obr_combinator_2*):



_obr_combinator_2

Nesporná výhoda tohoto typu útoku spočívá v tom, že jsou nejen vyzkoušena všechna *známá* hesla, ale jsou i tvořena nová, která nemusí být obsažena nikde

jinde. Při volbě slovníku má potom útočník vícero možností: buď využije dva na sobě nezávislé slovníky hesel (např. z úniků dat), nebo si vytvoří slovníky vlastní – popřípadě zkombinuje tyto dva přístupy. Potenciálně je tento přístup také variantou spojení hesel s čísly, jak již bylo zmíněno u dřívějších přístupů. Zároveň je ilustrací toho, že pro vytvoření silného hesla nemusí být postačující pouze délka: jedná-li se o slovní spojení, je možné, že právě pomocí tohoto útoku by došlo k jejich kombinaci a následnému prolomení.

***

K útokům jako takovým je třeba poznamenat, že výše uvedený přehled není zdaleka kompletní. Zaměřili jsme se zde primárně na útoky poskytované programem *hashcat*, přičemž problematika prolamování hesel sahá daleko dál. Abychom uvedli alespoň pár dalších příkladů, uveďme, že kromě výše nastíněných metod mohou být útočníkovým nástrojem také tzv. duhové tabulky (anglicky *rainbow tables*). Ty obsahují již předpřipravené dvojice hesel a jejich vypočítaných hašů, a tedy jsou v podstatě ulehčením útoku hrubou silou – místo toho, aby se haš pro každou sekvenci počítal během útoku samotného, využijí se předem spočítané hodnoty z duhové tabulky. Proti duhovým tabulkám se používá výše zmíněná sůl – díky tomu, že je to náhodný řetězec přidaný k heslu samotnému před hašováním, neměl by pak výsledný haš být obsažen v předem vypočítaných tabulkách. Dále v prolamování hesel existují také východiska pokročilejší složitosti, jako je např. generování hesel pomocí Markovovských řetězců nebo jiných typů pravděpodobnostních gramatik.

Důležitým faktorem, který nesmí být v tématu prolamování hesel opomenut, je výpočetní síla počítače, na kterém útok provádíme. Ta určuje, kolik kombinací bude ověřeno za jednu vteřinu. To se odvíjí od výpočetních možností procesoru (také CPU dle *central processing unit*), popřípadě grafické karty (také GPU dle *graphics processing unit*), přičemž jsou to právě grafické karty, co je ideální pro prolamování hesel, jelikož svou rychlostí předčí CPU: v některých případech se jedná až o 50× vyšší rychlost (Pavlíček, Sedláček, 2020). Zajímavostí je zde to, že za rozvoj grafických karet a jejich výpočetní sílu mohou útočníci vděčit hráčům počítačových her, pro které původně byly výkonné grafické karty primárně určeny.

***

Jak jsme mohli v této části pozorovat, uživatelská hesla mohou být problematická z hlediska jejich ochrany, avšak (bohužel) ne vždy nutně problematická z hlediska útočníka. Uvážíme-li množství typů útoků poskytovaných různými programy, dostáváme se jako uživatelé do zneklidňující situace. O to víc, když uvážíme, že v praxi nebývá užit jen jeden typ útoku, ale zpravidla konsekventně více z nich: tedy, pokud heslo „ustojí“ útok hrubou silou, hrozí (ne zcela opominutelné) riziko, že bude prolomeno slovníkovým útokem, jelikož bude jedním z až sta milionů hesel ve slovníku obsažených. Obraťme tedy nyní naši pozornost k uživatelské praxi. Na zbylých stranách této práce se budeme zabývat analýzami hesel, které si vytvořili uživatelé čtyř online služeb. Nejprve se tak – pro ukotvení hesel do určitého kontextu – zaměříme na popis těchto zkoumaných datasetů.

Datasety

Následující část poskytuje obecný přehled datasetů, které jsou předmětem analýz této práce. Na jedné straně se tato část pohybuje na rovině metainformací, kdy dané datasety zasazuje do kontextu z hlediska jejich původu, geografického zařazení a cílové skupiny konzumentů daných služeb; na straně druhé se orientuje na kvantitativně-lingvistickou užitých hesel a snaží se o co nejpřesnější vystižení trendů, které se v nich vyskytují.

— Kritéria výběru a předzpracování
V této práci budeme dále operovat se čtyřmi vybranými datasety. Jak již bylo zmíněno v předcházející části, veškerá tato data pocházejí z otevřených zdrojů a jsou anonymizována – při tvorbě této práce byly využity pouze seznamy hesel bez jakýchkoli jiných informací uniklých z daných serverů. Za účelem zachování anonymity samotných služeb, odkud data unikla, byla třem z analyzovaných datasetů přiřazena krycí jména: *Alpaka*, *Kosatka* a *Krajta*. Poslední z analyzovaných datasetů, *RockYou*, přejmenován nebyl, a to z toho důvodu, že se jedná o známý seznam hesel, který je užíván jak útočníky při slovníkových útocích, tak akademiky při analýzách hesel (viz např. Veras et al., 2012 nebo Devillers, 2012).

Výběr těchto konkrétních datasetů podléhal důležitému kritériu, a to *v jakém formátu hesla unikla*. Pro získání autentického náhledu na to, jaká hesla se skutečně v praxi používají, bylo třeba vybrat datasety hesel, které unikly ve formě plaintextu, tedy nebyly nijak šifrované, hašované nebo jinak chráněné. Pokud bychom pracovali s datasetem hesel, která již někdo prolomil a tato prolomená hesla dal k dispozici online, vystavovali bychom se riziku, že by nám byla dostupná pouze jednoduchá hesla, která bylo možné z hašů útočníkem získat a tedy ta složitější by nepodlehla analýze. Výsledky měření by takto byly zkreslené, jelikož by byly zjištěny pouze z jednoduché podmnožiny hesel. Tento fakt bude možné sledovat i v rámci této práce, jelikož pro srovnání byl zařazen jeden dataset této povahy: dataset Alpaka (v detailu rozebrán níže).

Před analýzou proběhla i úprava samotných datasetů: byla z nich odstraněna hesla o délce > 30 znaků. Tento krok byl učiněn na základě jiných studií (viz např. Devillers, 2010) a také motivován empiricky. Při kvalitativním průzkumu datasetů bylo zjištěno, že sekvence o větších délkách často nejsou hesla, ale kusy kódu *HTML* nebo *JavaScriptu*, popřípadě jiné anomálie – takto je tomu například u datasetu RockYou (Devillers, 2010). Dalším z důvodů omezení délky hesel je obsah datasetu Krajta, který kromě hesel obsahoval také MD5 haše, které jsou tvořeny 32 znaky. Omezení délky zde tedy slouží jako způsob odstranění anomálií, které pro naše potřeby zkoumání obecných trendů nemají velkou výpovědní hodnotu. V případě RockYou bylo nutné provést další restriktivní úpravy – ty jsou popsány níže v dedikované části.

— **Dataset (I): RockYou** (rok úniku: 2009)
První z vybraných souborů dat nese název RockYou. Jak již bylo zmíněno v úvodu této části, vzhledem k tomu, že se jedná o standardně užívaný, citovaný i analyzovaný dataset, nebyl ani pro účely této práce anonymizován. Jedná se o seznam hesel, která původně patřila uživatelům platformy RockYou.com, tedy sociální sítě zaměřené na online hry – z velké části se tedy nejspíše jedná o uživatele mladšího věku (Devillers, 2010). Roku 2009 byla tato platforma napadena útočníky, kteří z ní odcizili celkem 32 603 388 uživatelských záznamů (Siegler, 2009 [cit. 10-07-2021]), a to včetně hesel, která nebyla v databázi nijak chráněná – tedy unikla v podobě plaintextu. Z tohoto důvodu je RockYou hojně využíván útočníky při slovníkových útocích nebo i akademiky při analýzách hesel. Dnes je tento dataset volně dostupný, avšak pouze ve formě seznamu typů hesel, nikoli tokenů – nelze na něm tedy zkoumat frekvence zastoupení jednotlivých typů hesel.

Jak již bylo předesláno výše, hesla obsažená v tomto datasetu bylo třeba filtrovat. Kromě úpravy, kterou prošly i ostatní datasety (tedy odstranění hesel o délce > 30 znaků), byl v tomto datasetu problém s kódováním jednotlivých hesel. Kromě UTF-8 se v něm vyskytovaly i jiné způsoby kódování, které nebylo možné v rámci jazyka *R* (který je pro analýzy této práce využit) správně interpretovat bez toho, aby to mělo dopad na funkčnost kódu. Z původního počtu 14 344 386 hesel bylo celkem odstraněno 218, u kterých se tento problém s kódováním vyskytoval – což lze označit za zanedbatelné množství.

— **Dataset (II): Kosatka** (rok úniku: 2015)
Dataset (II), který v této práci nese krycí název Kosatka, pochází ze Spojených států. Jedná se o společnost poskytující připojení k internetu a kabelovou televizi – předpokládáme tedy, že (I) její cílová skupina je širší, než je ta výše popsaného datasetu; (II) vzhledem k poskytovaným službám budou uživatelé primárně dospělí lidé. Co do množství jednotlivých tokenů, s celkem 590 298 hesly se jedná o nejmenší zde zkoumaný dataset.

— **Dataset (III): Alpaka** (rok úniku: 2015)
Třetí dataset, se kterým budeme v rámci této práce operovat, zde má krycí název Alpaka. Platforma, ze které pochází, se zaměřuje na online prodej techniky, domácí elektroniky a spotřebičů, hraček apod., očekáváme zde tedy také primární zastoupení dospělých uživatelů. Oproti předešlým dvěma datasetům má jistá specifika. Prvním z nich je prostředí, ze kterého pochází – zatímco ostatní datasety jsou orientované (primárně, nikoli výhradně) na anglofonní prostředí, tento je z prostředí českého. Hesla lexikální povahy, které se v něm vyskytují, jsou tak převážně v českém jazyce, což je také jeden z důvodů jeho zařazení. Obecně má tento dataset jeden zásadní nedostatek: hesla byla v databázi ukládána jako haše, nikoli jako prostý text. Zde analyzovaný dataset Alpaka je souborem hesel, které se útočníkovi provedlo prolomit, a tedy není reprezentativním zástupcem běžné uživatelské praxe. Z těchto důvodů u tohoto datasetu také nelze očekávat složitější struktury výstavby hesel, jelikož je velmi pravděpodobné, že delší či složitější hesla se útočníkovi nepodařilo prolomit. Ale i to je dalším důvodem jeho zařazení do zde prováděných analýz – poukáže na fakt, že pro autentický náhled na běžnou uživatelskou praxi nelze pracovat s datasety, které neunikly v plaintextu.

— **Dataset (IV): Krajta** (rok úniku: 2016)
Poslední z vybraných datasetů je dále v této práci označován jako Krajta. Stejně jako Kosatka a RockYou, i tento dataset je primárně anglofonní – daná služba je dostupná ve Spojených státech, Kanadě a Mexiku. Jedná se o službu, která nabízí peněžní odměny za splnění menších online úkonů jako je např. vyplnění dotazníku či nákup na konkrétním e-shopu. V uživatelské skupině tak očekáváme především uživatele mladší až střední dospělosti. Při útoku na službu unikla hesla ve dvou formátech: přibližně polovinu tvořila hesla v plaintextu, druhou polovinu pak MD5 haše, které chránily hesla novějších uživatelů. Jediné předzpracování, které pro tento dataset bylo nutné provést, tak bylo odstranění hašů, což bylo zajištěno omezením délky analyzovaných hesel na max. 30 znaků (MD5 haš je jakožto sekvence tvořen 32 znaky).

***

Pro získání prvotního obecného náhledu na nejčastější hesla a to, kolik uživatelů celkem je sdílí, uvádíme zde tabulku dvaceti nejfrekventovanějších hesel pro tři ze zkoumaných datasetů (viz *_tab_datasety_freq*). Můžeme zde sledovat podobu hesla, počet uživatelů, kteří dané heslo sdílí a zároveň také jeho délku. V této tabulce je vynechán dataset RockYou, jelikož vzhledem k tomu, že se jedná pouze o seznam hesel, nelze v jeho případě zkoumat frekvence jednotlivých tokenů. Na prvních příčkách se potvrzují standardně uváděné volby hesel jako jsou jednoduché číselné postupky, <password> či <heslo>, dále jednoduše zapamatovatelná slova jako např. <baseball> nebo fráze jako vtipná upomínka <changeme>. Obecně je zde zajímavé zastoupení deminutiv, popř. slov/frází s pozitivním příznakem. V českém datasetu Alpaka tak vidíme hesla jako <slunicko>, <beruska> nebo <milacek>, v anglofonních pak <sunshine>, <princess> nebo <iloveyou>.

Co se týká síly těchto hesel, lze konstatovat následující. Tabulka zobrazuje jak znění samotných hesel a jejich četnost, tak i jejich délku, která ani v jednom z případů nepřesahuje osm znaků. Uvážíme-li diskuzi k síle hesla uvedené v předcházející části, pozorujeme zde, že žádné z těchto hesel není *silné*, a to z několika důvodů. Jedním z nich je jejich délka, jelikož právě osm je ta maximální, na kterou se potenciálně vyplatí útočit hrubou silou; jinak řečeno: je v našich výpočetních možnostech tento útok provést (což platí pro některé typy hašů, např. MD5; oproti tomu pro jiné typy, jako je např. bcrypt, by i tato délka byla za aktuálních podmínek výpočetně náročnější). Tento typ útoku je také, jak víme, ovlivněn množinou znaků, které tato hesla užívají. Vidíme, že se v uvedených heslech nevyskytují velká písmena ani symboly, užity jsou zde pouze dvě sady znaků: malá písmena abecedy, tedy množina o 26 prvcích, a čísla, tedy množina o 10 prvcích. Např. pro prolomení hesla <123456> by tak útočník musel hrubou silou vyzkoušet „pouhých“ 10^6 kombinací, pro prolomení hesla <password> pak 26^8 kombinací[4]. Ani jedno z nich při dnešní výpočetní síle není vhodným adeptem pro silné heslo, jelikož takováto hesla jsou pak prolomitelná v řádu minut až hodin (záleží samozřejmě na

[4] To platí za předpokladu, že by útočník znal původní délku hesla – pokud ne, postupoval by dle počtu znaků a výpočet množství kombinací by pro heslo <123456> vypadal následovně: $10^1 + 10^2 + 10^3 + \ldots + 10^6 = 1\,111\,110$ kombinací.

typu útoku a typu haše, každopádně se může jednat až o desítky milionů kombinací za vteřinu – ne-li více). To samozřejmě v případě, kdy bychom tato hesla prolamovali hrubou silou – ve chvíli, kdy vezmeme v potaz slovníkový útok, je možné, že by tato hesla byla odhalena rychleji vzhledem k tomu, že slovníky jsou často designované tak, aby zahrnovaly právě ony nejčastější uživatelské volby hesel.

Nejfrekventovanější hesla							
Rank	**Heslo**	**Frekvence**	**Délka [znak]**	**Rank**	**Heslo**	**Frekvence**	**Délka [znak]**
Alpaka							
1	123456	3 498	6	11	*(adresa platformy)*	1 054	7
2	12345	2 917	5	12	martina	1 030	7
3	heslo	2 005	5	13	eliska	1 000	6
4	*(název platformy)*	1 894	6	14	michal	969	6
5	martin	1 796	6	15	terezka	969	7
6	slunicko	1 414	8	16	milacek	926	7
7	korunka	1 267	7	17	tereza	922	6
8	beruska	1 226	7	18	sparta	904	6
9	veronika	1 159	8	19	lenka	901	5
10	monika	1 087	6	20	maminka	897	7
Kosatka							
1	password	4 846	8	11	*(název platformy)*	421	7
2	bluebird	1 778	8	12	redcar	405	6
3	12345678	1 570	8	13	bluefish	384	8
4	changeme	1 245	8	14	catdog	377	6
5	baseball	1 206	8	15	ocean	302	5
6	sunshine	1 100	8	16	blue	288	4
7	password1	677	9	17	123456	285	6
8	football	649	8	18	princess	273	8
9	1234	554	4	19	bluesky	252	7
10	*(název plat. a č. 1)*	489	8	20	samantha	240	8
Krajta							
1	123456	18 332	6	11	12345	1 341	5
2	password	5 513	8	12	1234567	1 315	7
3	123456789	3 987	9	13	123123	1 308	6
4	*(název platformy)*	3 164	9	14	princess	1 266	8
5	money	2 396	5	15	monkey	1 138	6
6	qwerty	2 328	6	16	password1	1 001	9
7	12345678	2 006	8	17	sunshine	985	8
8	abc123	1 939	6	18	michael	982	7
9	iloveyou	1 458	8	19	soccer	942	6
10	111111	1 449	6	20	shadow	917	6

_tab_datasety_freq

— Základní kvantitativní indexy datasetů

V následující části budeme věnovat pozornost základním kvantitativním indexům, které jsou využity pro popis zkoumaných datasetů. Pro každý z indexů budou uvedeny výsledky měření pro každý z daných datasetů, přičemž závěrem kapitoly budou dané hodnoty vyobrazeny souhrnně v tabulce pro každý jednotlivý dataset – a to jak pro tokeny, tak i typy. Výjimkou je, stejně jako v předchozím případě, RockYou, u něhož se zaměřujeme pouze na typy, jelikož tokeny zde nelze zkoumat. Kromě hesel jako takových bylo zkoumáno také zastoupení jednotlivých množin znaků v rámci datasetů. V následující části se na tyto indexy zaměříme, přičemž

jejich pořadí nebude nutně odpovídat pořadí v daných tabulkách. S těmito indexy budeme pracovat i na dalších stranách této práce, přičemž zde bude poskytnut základní výklad toho, *čeho je daný index popisem*. Pro každý z těchto indexů bude uvedena náhledová tabulka výsledků zkoumaných datasetů, přičemž souhrnná tabulka výsledků bude poskytnuta v závěru části.

1) Počet typů, počet tokenů, TTR

Jak bylo popsáno v části dedikované terminologii, termínem *token* se zde míní každé jednotlivé heslo uvnitř daného zkoumaného datasetu (nebo také každý *uživatel*), přičemž termínem *typ* se zde míní každý tvar, či každá podoba, hesla. TTR (dle anglického *type–token ratio*, tedy poměr typů a tokenů) je poté jedním z indexů kvantitativní lingvistiky, kde se standardně využívá pro definici bohatství slovníku. V základní podobě[5] jej vypočítáme jako podíl typů a tokenů:

$$TTR = \frac{Počet\ typů}{Počet\ tokenů}$$

a jeho výsledná hodnota se pohybuje v rozmezí 0 a 1 (včetně). Čím vyšší hodnota TTR, tím více různých slov autor textu užívá: vyšší TTR značí vysoký poměr tokenů vůči typům, a tedy větší slovní bohatství autora. Maximální hodnota, TTR = 1, pak značí, že žádné ze slov se v textu neopakuje (v takovém případě si lze představit např. seznam různých položek, kdy každá se vyskytuje právě jednou).

V našem případě TTR vyčísluje míru repetice hesel v rámci datasetu – jinými slovy, nakolik se hesla opakují. Popišme si tuto problematiku na případu datasetu Kosatka. Původní soubor 590 298 hesel obsahuje 413 959 typů, což udává hodnotu TTR 0,70, což značí, že hesla se zde poměrně málo opakují; jinými slovy také to, že až 70 % datasetu jsou typy.

Ve zde analyzovaných datasetech se hodnota TTR pohybuje mezi 0,50 a 0,70, přičemž nejnižší hodnota náleží českému datasetu Alpaka, u nějž očekáváme potenciálně zkreslené výsledky. Kvůli malému srovnání však zde nelze učit, zda jeho hodnota TTR je zde anomálií nebo je ještě v normě. Uvážíme-li však, že se jedná o poměrně malý dataset (srov. Krajta nebo RockYou, které mají až miliony typů), můžeme o TTR uvažovat jako o první vlaštovce anomálie: z původních ~766,5 tisíců hesel je polovina typů, zatímco ve druhém nejmenším datasetu, Kosatka, je hodnota značně vyšší.

---

[5] Existují další modifikace daného výpočtu. Korpusová lingvistika totiž počítá s velkým objemem jazykových dat, která jsou více heterogenní než námi zkoumané datasety, a tak pro ni základní výpočet TTR není vždy dostačující. Velikost korpusu a TTR jsou v nepřímé úměře: čím větší korpus, tím nižší hodnota TTR. Tento jev je způsoben ustálenými (povětšinou gramatickými) slovy v jazyce (např. v angličtině člen *the*, v češtině spojka *a* nebo zájmeno *ten*), které se budou v korpusu objevovat pravidelně; oproti nim se budou objevovat slova tematická s razantně nižší frekvencí. Tento problém se řeší např. **standardizovaným TTR** (ang. *standardised* nebo také *mean TTR*), které TTR korpusu počítá po úsecích o dané délce; z naměřených hodnot se dále spočítá průměr, který udává reprezentativnější náhled na data. (Baker, 2006) Pro naše potřeby je však základní výpočet TTR dostačující.

Hodnota TTR			
	Alpaka	**Kosatka**	**Krajta**
Počet tokenů	766 421	590 298	2 234 829
Počet typů	381 908	413 959	1 388 099
TTR	0,50	0,70	0,62

_tab_TTR

2) Množství hapaxů

Množství hapax legomen udává, kolik z hesel v datasetu je unikátní, tedy kolik z nich se objevuje pouze jednou[6]. Tento index je sledována v poměru jak k tokenům, tak k typům. V případě tokenů udává množství hesel celého datasetu, které mají pouze jeden výskyt – je tak zaměřen na uživatelskou zvyklost. V případě typů popisuje to, kolik z forem hesel je unikátních – zaměřuje se tedy na unikátnost různých podob hesel.

Množství hapaxů			
	Alpaka	**Kosatka**	**Krajta**
Tokeny [%]	43,48	60,48	53,20
Typy [%]	87,26	86,25	85,65

_tab_hapaxy

3) Pareto 20/x

Tento index vychází z tzv. Paretova principu, označován také jako *pravidlo 80/20*, který obecně tvrdí, že až *80 % důsledků je způsobeno 20 % příčin*. Tento princip nachází uplatnění v nejrůznějších oblastech; příkladem uveďme kriminalistiku, kde se tvrdí, že přibližně 80 % zločinů je spácháno 20 % kriminálníků, nebo ekonomiku, kde 80 % výdělku společnosti pochází od 20 % zákazníků atd. (Koch, 1999). Nutno podotknout, že se nejedná o přesně definovaný zákon, ale empiricky pozorovaný princip. Pro potřeby této práce jsme jej označili jako 20/x, jelikož zkoumáme, kolik % datasetu tvoří 20 % jeho nejfrekventovanějších hesel, přičemž dle Paretova principu očekáváme právě 80. Jak lze pozorovat v tabulce *_tab_pareto*, ani v jednom z případů nelze hovořit o platnosti tohoto principu, neboť se pohybujeme mezi ~43 a ~60 procenty. Tento index taktéž dokládá poměrně velkou míru heterogenity, různorodosti, hesel.

Pareto 20/x [%]	
dataset	**x**
Alpaka	60,14
Kosatka	43,90
Krajta	50,31

_tab_pareto

[6] Slovo *unikátní* jako takové není zcela jednoznačné: může se jednat buď o unikátnost z hlediska formy (tedy o typy), nebo o unikátnost z hlediska výskytu. Zde a na následujících stránkách se (stejně jako v tomto případě) vždy přidržíme druhé definice, tedy unikátnosti výskytu. Pokud je zde heslo označeno za unikátní, jeho frekvence je rovna jedné (není-li uvedeno jinak).

4) Zastoupení znaků v rámci souboru

Tématu zastoupení znaků v rámci datasetu bude dopodrobna věnovaná jedna část této práce, nebudeme zde tedy provádět její vyčerpávající popis – zde se pro uvedení do kontextu zaměříme na zastoupení jednotlivých množin znaků. Zároveň, z důvodu lingvistického zaměření této práce, zde při rozdělení alfabetických znaků pracujeme především s konsonanty a vokály, zatímco běžně při provádění analýz je zaměření směrování spíše vůči rozdílu užití malých a velkých písmen. To zde však také nechceme opominout – blíže se s ním setkáme v dalších částech této práce. Co se dále týká uvedených počtů, existuje výrazný nepoměr užití daných skupin znaků. Je znatelné, že uživatelé výrazně preferují užití alfabetických znaků, což může být způsobeno vícero faktory: (I) může to být z důvodu výrazného lexikálního zastoupení v datasetech (čímž se vracíme k argumentu zapamatovatelnosti), (II) v případě generování hesel jsou alfabetické znaky poněkud lepší volbou, jelikož jejich množina je oproti např. číslům více než dvojnásobná (26 + 26 znaků malých + velkých písmen oproti 10 číslicím). Překvapivé zde však je nízké zastoupení speciálních symbolů – ve všech případech se jedná o zlomky procent. To si lze nejspíše částečně vysvětlit uživatelským zvykem: alfanumerické znaky jsou lidem přeci jen bližší než např. interpunkce, jelikož jsou z nich tvořena slova, data (jako např. letopočty) atp. Zároveň je nutné uvážit roky úniku dat z jednotlivých platforem (nejnovější pochází z r. 2016) – s narůstajícím povědomím o kyberbezpečnosti a tvorbě co možná nejlepšího hesla je možné (avšak ne nutně pravděpodobné), že novější data by v tomto ohledu poskytla jiné výsledky, které by kupříkladu ukazovaly vyšší zastoupení speciálních symbolů, vyšší délky hesel atp.

Znaky v rámci datasetů				
	Alpaka		**Kosatka**	
	Tokeny	**Typy**	**Tokeny**	**Typy**
Celkový počet	5 011 617	2 466 479	4 881 293	3 512 765
Konsonanty	53,03 %	52,70 %	49,22 %	47,13 %
Vokály	30,17 %	22,77 %	28,40 %	27,00 %
Čísla	16,71 %	24,43 %	25,70 %	22,25 %
Spec. znaky	0,08 %	0,09 %	0,13 %	0,16 %
	Krajta		**RockYou**	
	Tokeny	**Typy**	**Typy**	
Celkový počet	17 783 822	11 633 176	125 240 372	
Konsonanty	45,37 %	43,39 %	39,42 %	
Vokály	28,63 %	26,87 %	25,97 %	
Čísla	25,57 %	29,17 %	33,40 %	
Spec. znaky	0,42 %	0,57 %	1,20 %	

_tab_znaky

5) Délka hesla

Zajímavým indexem, kterou lze pozorovat v tabulce *_tab_delky-hesel*, jsou délky hesel v jednotlivých datasetech – tedy to, z kolika znaků se daná hesla skládají. Jak lze pozorovat, pro tři ze čtyř analyzovaných datasetů se průměrná délka typů i tokenů pohybuje okolo osmi znaků, zatímco průměrná délka hesla v datasetu Alpaka se pohybuje okolo šesti znaků.

Co se týká anglofonních datasetů (Kosatka, Krajta, RockYou), délka se ve všech případech pohybuje blízko osmi znaků, a to jak v případě typů i tokenů: spadá sem

průměr i **medián**. To obecně napovídá tomu, že právě osm znaků je nejčastější délkou hesel obsažených v těchto datasetech. Zaměříme-li se na **95% interval délek**, který je odvozený na základě 2,5 a 97,5 kvantilu, zjišťujeme, že 95 % všech hesel spadá svou délkou do rozmezí, které se pro každý dataset mírně liší. Zajímavé zde je, že čím větší dataset, tím delší heslo na maximální hranici intervalu: pro nejmenší Kosatku je to 12 znaků, pro větší Krajtu je to 13 znaků, pro největší RockYou je to až 15 znaků. Tento úkaz lze vysvětlit možností přímé úměry velikosti datasetu a pravděpodobnosti výskytu různých, a tedy potenciálně i delších, hesel[7].

V případě českého datasetu (Alpaka) sledujeme hodnoty pohybující se okolo čísla šest, a to jak v případě průměru, tak i mediánu; 95% interval délek je taktéž nižších hodnot. Zde se potenciálně projevuje nepřesnost tohoto datasetu: obecně nižší hodnoty zde mohou být způsobeny tím, že je to dataset získaný z hašů. Jak víme, bylo na něj uplatněno vícero metod prolamování – je zde např. možné, že útočník zde zkombinoval útok hrubou silou, kterým rozkryl hesla o malých délkách, a slovníkový útok, kterým rozkryl další, a především delší, obsažená hesla. V takovém případě lze předpokládat, že komplexnější a delší hesla zůstala nerozkryta, což by pak mělo efekt, který zde můžeme pozorovat. Samozřejmě je možné, že obecně nižší hodnoty zde mohou být způsobeny i jiným faktorem, který jsme nevzali v potaz nebo nám zůstává skryt (např. uživatelská preference českých uživatelů, vliv českého jazyka, pravidla nastavená platformou jako např. maximální délka hesla atp.). Definitivně však tento dataset nemůžeme označit za směrodatný, a to právě z důvodu panující nejistoty ohledně jeho původu.

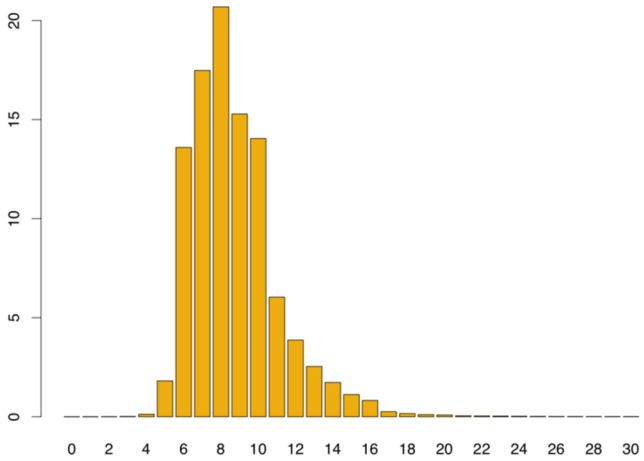
Indexy délky hesel				
	Alpaka		**Kosatka**	
	Tokeny	**Typy**	**Tokeny**	**Typy**
Průměrná délka	6,54	6,46	8,27	8,49
Medián délek	6	6	8	8
95% interval délek	5–10	5–10	5–12	5–12
	Krajta		**RockYou**	
	Tokeny	**Typy**	**Typy**	
Průměrná délka	7,96	8,38	8,73	
Medián délek	8	8	8	
95% interval délek	5–13	6–13	6–15	

_tab_delky-hesel

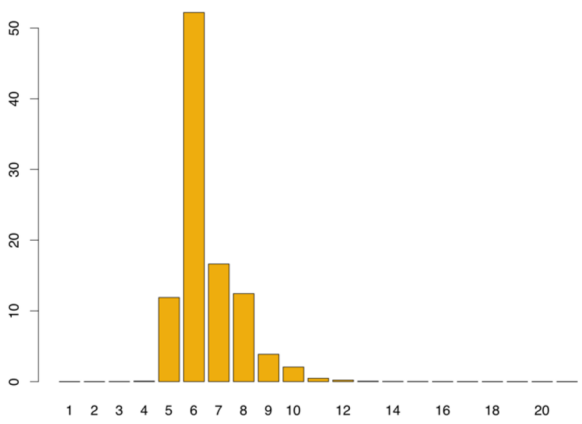
Následující soubor grafů shrnuje rozdíly v zastoupení délek mezi tokeny (nazvané *_graf_[název datasetu]_tokeny*) a typy (nazvané *_graf_[název datasetu]_typy*). Grafy tokenů poskytují informaci o celých datasetech – které délky jsou obecně v datasetu nejčastější bez ohledu na podobu hesla. Grafy typů oproti tomu vypovídají o nejčastějších délkách jednotlivých tvarů hesel. Výjimkou je zde (opět) dataset RockYou, jehož zastoupení délek typů uvádíme níže (*_graf_RockYou_typy*). Obecně jsou tyto ilustrace délek přínosné svým vyobrazením rozdílů užití různých

[7] Na druhou stranu je nutné vzít v úvahu, že vzorek zde zkoumaných datasetů je poměrně malý na to, abychom z něj vyvozovali obecné závěry. Maximální délka může být v některých případech ovlivněna i např. povahou dané platformy či restrikcemi, které jsou kladeny na uživatele při tvorbě hesla.

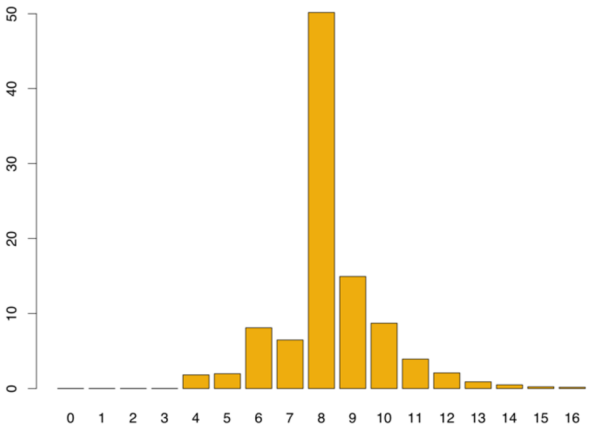
délek typů a jejich zastoupení v datasetu. Nejlépe je tento rozdíl pozorovatelný v datasetu Alpaka, kde můžeme pozorovat, že ačkoli z hlediska zastoupení typy hesel o délkách 7 a 8 prakticky stejné, reálné užití převažuje ve prospěch délky 7. Jinými slovy tedy existují *vícekrát zastoupené tvary hesel* o délce 7.



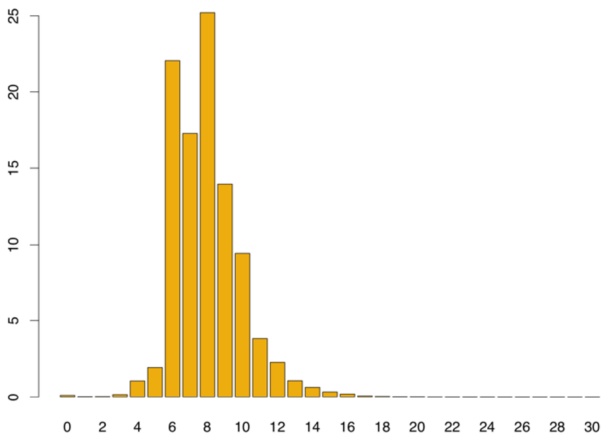
_graf_RockYou_typy



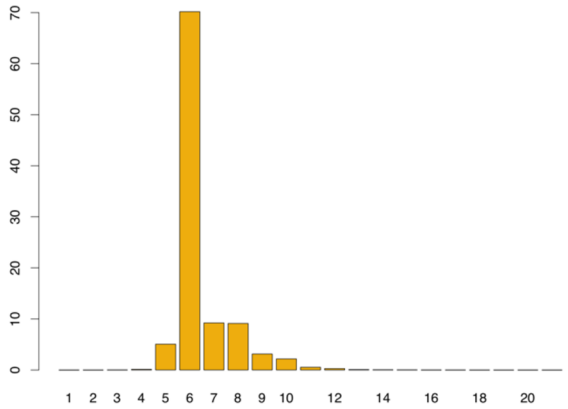
graf_Alpaka_tokeny



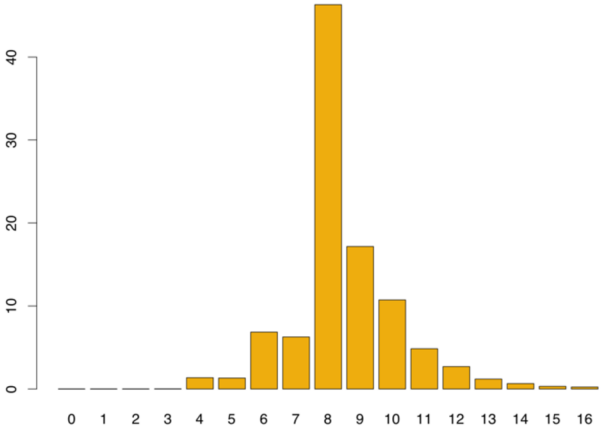
graf_Kosatka_tokeny



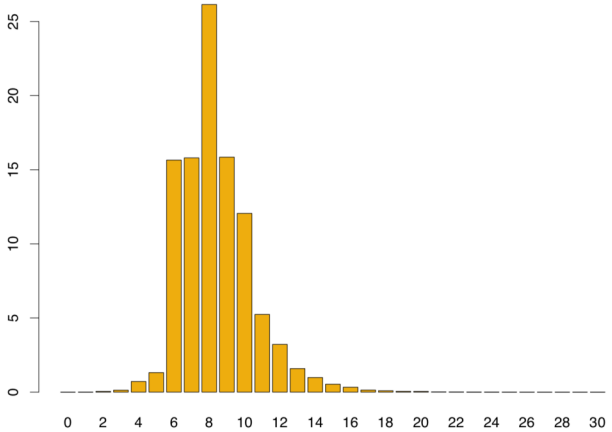
graf_Krajta_tokeny



graf_Alpaka_typy



graf_Kosatka_typy



graf_Krajta_typy

***

Výše uvedené tabulky[8] a indexy se snaží o co možná nejlepší vystižení obecných vlastností datasetů a v nich vedoucích trendů: můžeme zde sledovat nejčastější frekvence, délky, znaky atp. Co nám tato data mimo jiné poskytují, je náhled na běžnou uživatelskou praxi – přestože, jak již bylo zmíněno výše, z důvodu poměrně malého vzorku datasetů, nelze výsledky výrazně zobecňovat, získáváme takto vhled do alespoň několika souborů dat z reálného prostředí a můžeme tak pozorovat nejčastější vlastnosti uživatelsky tvořených hesel. Obecně se v těchto přehledech potvrdil předpoklad předeslaný v úvodní části, a tedy to, že lexikální složka je přítomna v nemalých částech všech datasetů hesel. Toto zjištění otevírá otázku toho, *jaká slova* si uživatelé nejčastěji tvoří jako hesla. V ukázce nejčastějších hesel můžeme pozorovat zastoupení vlastních jmen, sportů a dalších. Následující část této práce se tedy zaměří na možnost identifikace konkrétních témat, a to analýzou sémantiky v datasetech.

[8] Souhrnně je lze nalézt v **příloze 1**

Sémantika v datasetech

V ilustrační tabulce nejčastějších hesel (*_tab_datasety_freq*) uvedené výše v této práci jsme mohli pozorovat dvacet nejfrekventovanějších podob hesel – tedy dvacet hesel, která si uživatelé vytvořili na dané online službě nejčastěji. Převládajícím rysem, který se v těchto heslech objevuje, je jejich lexikální povaha. Tento rys je logický, uvážíme-li, že do tvorby takových hesel nevstupuje náhoda; lexikální hesla odrážejí mimojazykovou realitu, která je do jisté míry společná všem uživatelům, a tedy je pravděpodobné, že pokud využijí k tvorbě hesel lexika, z jisté části dojde ke shodě. Nyní tedy obraťme naši pozornost k tomu, jaká je významová charakteristika nejčastějších hesel: zaměřme se na sémantickou stránku datasetů.

— Výběr hesel a metoda zpracování
Pro analýzu sémantiky bylo ze tří datasetů (Alpaka, Kosatka, Krajta) vybráno 150 nejčastějších hesel. Dataset RockYou byl z této analýzy vynechán, jelikož v jeho případě nelze identifikovat nejčastější výskyty, a tedy nejčastěji se vyskytující lexikum[9]. Počet 150 hesel byl zvolen z praktických důvodů s ohledem na prezentaci daných výsledků: je to dostatečně velký počet pro získání představy o převládajících trendech z hlediska sémantiky a zároveň je tento počet přehledně prezentovatelný ve 2D grafu, který bude níže využita pro ilustraci výsledků. Námi zvolených 150 hesel bylo dále upraveno. Vzhledem k tomu, že se zde zaměřujeme na čistě lexikální stránku věci, byla nejprve z hesel odstraněna veškerá čísla a speciální znaky; např. z hesla typu <heslo1!> byla extrahována pouze jeho lexikální část, a tedy do sémantické analýzy vstoupilo jako <heslo>. Dále byla veškerá hesla převedena na malá písmena abecedy. Velikost písmen povětšinou v jazyce nemá vliv na význam slova (výjimku lze najít např. u příjmení či jiných pojmenovaní, avšak na tyto nuance nebyl brán zřetel), a tedy hesla jako např. <Heslo> byla převedena na <heslo>, a to proto, aby byla interpretována správně jako jedna významová jednotka. Zaměřme se nyní na technickou stránku věci.

Sémantická analýza byla provedena za pomoci tzv. **sémantických embeddingů**, které fungují na principu odhalení latentní sémantiky slov v korpusech textů. Myšlenka, která za nimi stojí, je taková, že pokud se určitá slova objevují ve stejných kontextech, budou si nejspíše sémanticky blízká. Sémantické embeddingy stojí na metodách strojového učení, kdy na objemném korpusu textů je natrénován model (dnes již téměř výhradně) neuronové sítě, který má za úkol identifikaci právě těchto kontextů a odvození latentních témat. Dle těchto témat je dále schopen převést předložená slova na vektory, tedy seznamy čísel o daném počtu komponentů, pomocí kterých umisťuje slova do prostoru, přičemž platí, že sémanticky podobná slova jsou si v daném prostoru blízká. Metod existuje vícero, nejznámější z nich jsou Word2Vec (Mikolov, 2013) a FastText (Bojanowski et al. 2017) – druhý

[9] Vybíráme-li hesla z datasetu náhodným výběrem, narazíme na vysokou míru heterogenity co do náhodnosti/lexika a obecně tvarů hesel. Vzhledem k tomu, že se v datasetech vyskytuje velké množství hapaxů, i velká část náhodně vybraných hesel obsahuje, vedle lexika, mnoho hesel, u kterých nelze smýšlet o sémantice. Tento fakt diskvalifikuje RockYou ze zde prováděné analýzy.

z uvedených byl použit pro námi prováděnou analýzu. Metoda FastText je dnes široce užívanou technikou, která užívá 300 komponentů pro umístění slova v daném prostoru; její nespornou výhodou je také to, že dokáže do jisté míry řešit flexi. Blízkost dvojice vektorů slov je dále spočítána kosinovou podobností, která je formálně definována jako

$$podobnost = \cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} \boldsymbol{A}_i \times \boldsymbol{B}_i}{\sqrt{\sum_{i=1}^{n} (\boldsymbol{A}_i)^2} \times \sqrt{\sum_{i=1}^{n} (\boldsymbol{B}_i)^2}}$$

kde $\boldsymbol{A}$ a $\boldsymbol{B}$ jsou jednotlivé vektory a n je počet komponent; kosinová podobnost takto počítá velikost úhlu daných vektorů. Zde dále pro vizualizaci výsledků užíváme metody MDS (*Multidimensional Scaling*, česky vícerozměrné škálování), a tedy – spíše než kosinovu *podobnost* – je třeba určit kosinovou *vzdálenost*, kterou vypočítáme jako

$$vzdálenost(\boldsymbol{A}, \boldsymbol{B}) = 1 - |podobnost(\boldsymbol{A}, \boldsymbol{B})|$$

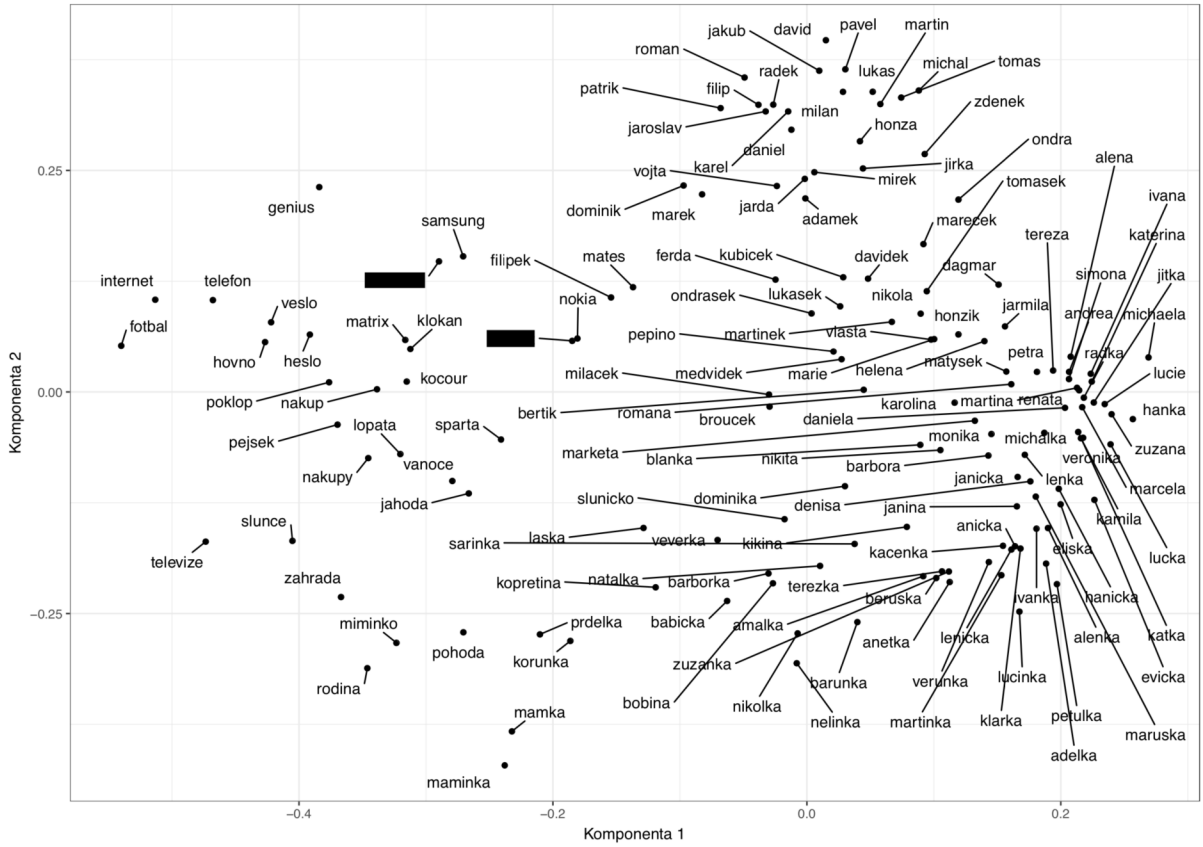
tedy jako rozdíl absolutní hodnoty podobnosti a jedné, kdy nula je shoda a jedna je maximální podobnost. Touto metodou jsme tedy číselně vyjádřili (ne)podobnost jednotlivých vektorů. Zaměřme se dále na metodu užitou k vizualizaci výsledků, tedy již zmíněné MDS.

Vícerozměrné škálování či **MDS** (jak bude dále označováno) (Torgerson, 1952) je technika využívající vypočítaných vzdáleností objektů ve formě bodů pro jejich vizualizaci ve 2D (někdy i 3D) prostoru charakteru mapy, kdy platí, že body, které jsou si blízké, mají podobné vlastnosti a naopak body, které mají nejméně společných vlastností, jsou i na mapě vyobrazeny daleko od sebe. Vlastnosti, které MDS pro vizualizaci užívá, nejsou explicitně dané, jedná se o kombinace latentních vlastností, a tedy i nelze přesně pojmenovat, co popisují osy výsledných grafů. Výsledkem MDS je pak v ideálním případě mapa, na které lze sledovat shluky sobě podobných objektů, určit, které objekty jsou si blízké, a případně identifikovat skrytá témata. Výstupem takovéto analýzy v našem případě ideálně očekáváme identifikaci několika skupin, na základě kterých budeme moci určit, jaký je charakter 150 nejfrekventovanějších hesel z hlediska jejich významu.

V následující části se budeme věnovat výstupům analýz jednotlivých datasetů. Jak již bylo zmíněno výše, vzhledem k povaze MDS nelze přesně pojmenovat osy grafů (ty jsou výsledkem kombinací latentních vlastností), mají zde tak názvy *Komponenta 1* a *Komponenta 2*. Jednotlivá hesla jsou reprezentována bodem v dvourozměrném prostoru; vzhledem k hustotě výskytu v určitých místech bylo nutné vést spojnici mezi zněním hesla a jeho umístěním v prostoru. Dále, pro zachování anonymity, byla hesla obsahující název platformy překryta černým obdélníkem.

— Výsledky analýzy

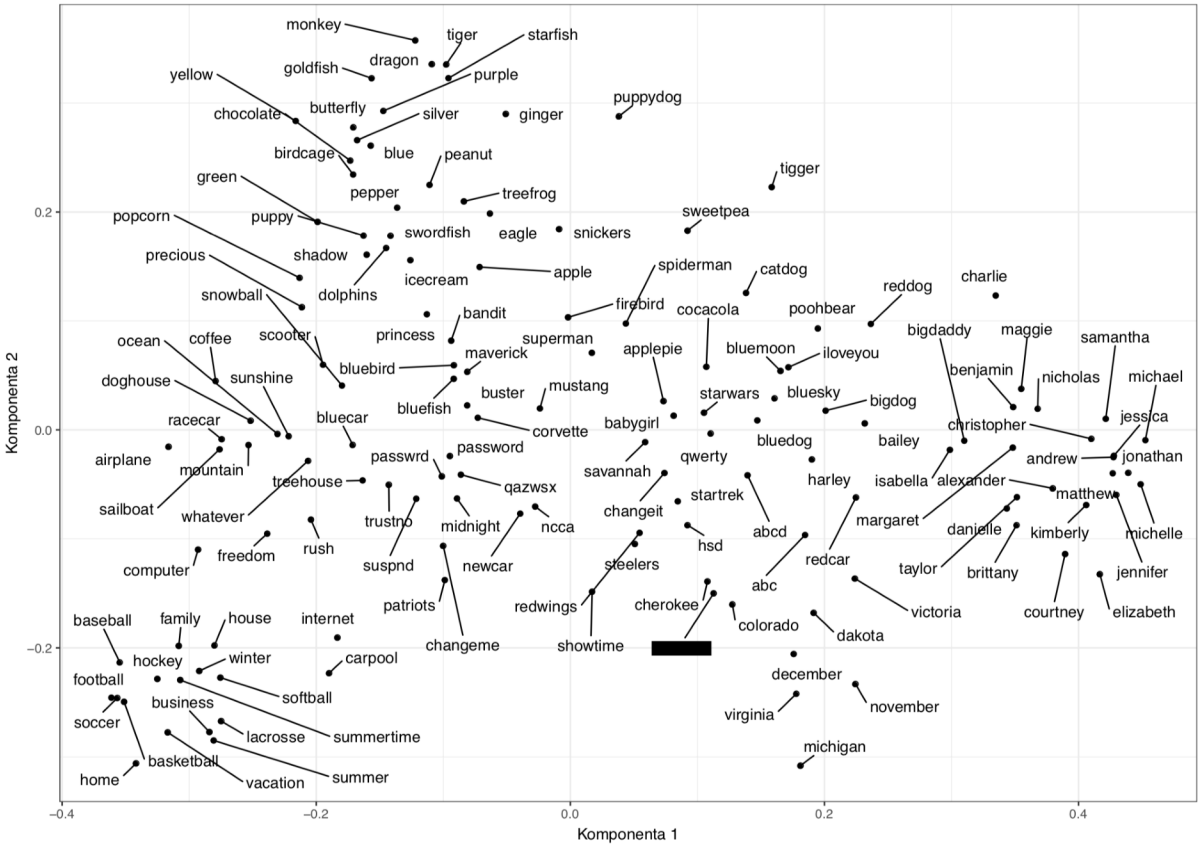
Zaměřme se nejprve na obsahovou stránku datasetu Alpaka. Výstupem analýzy je níže vyobrazený graf *_graf_semantika_Alpaka*, kde můžeme pozorovat několik prominentních uskupení.



_graf_semantika_Alpaka

Nejvýraznější seskupení můžeme pozorovat na pravé polovině mapy, kde se nachází shluk jmen. Ta jsou horizontálně rozdělena na ženská a mužská; v dolní části vidíme hustší zastoupení jmen ženských. Co je v tomto ohledu zajímavé, je to, že krom rozdělení jmen na mužská a ženská lze také identifikovat subkategorie deminutiv, která se vyskytují pod příslušnou, mužskou či ženskou, částí jmen. Dále stojí za povšimnutí to, že univerzální jména jako <nikola> nebo <vlasta> se správně vyskytují na předělu mezi mužskými a ženskými jmény. V levé polovině mapy pak vidíme uskupení, u kterého je již těžší témata přesně identifikovat. Ve spodní části se nachází slova, která mohou potenciálně souviset s tématem léta, víkendu či jiných kratochvil – vidíme zde např. slova jako <pohoda>, <zahrada> nebo <rodina>. Avšak shluk nacházející se nad tímto je co do vyskytujících se témat již zdánlivě náhodný: vidíme zde slova jako <poklop>, <nakup>, <heslo>, <kocour> nebo <vanoce>. V horní části lze pak identifikovat slova technologického rázu (např. <internet>, <telefon>); jak však spolu souvisí jim blízká slova <matrix> a <klokan>, to lze jen stěží interpretovat. Obecně však můžeme říci, že vedoucím trendem zde sledovaných hesel je užití vlastních jmen: mužských i ženských, neutrálních i deminutiv.

Dalším z analyzovaných datasetů je Kosatka, vizualizaci jejíž sémantiky lze sledovat níže v grafu *_graf_semantika_Kosatka*.



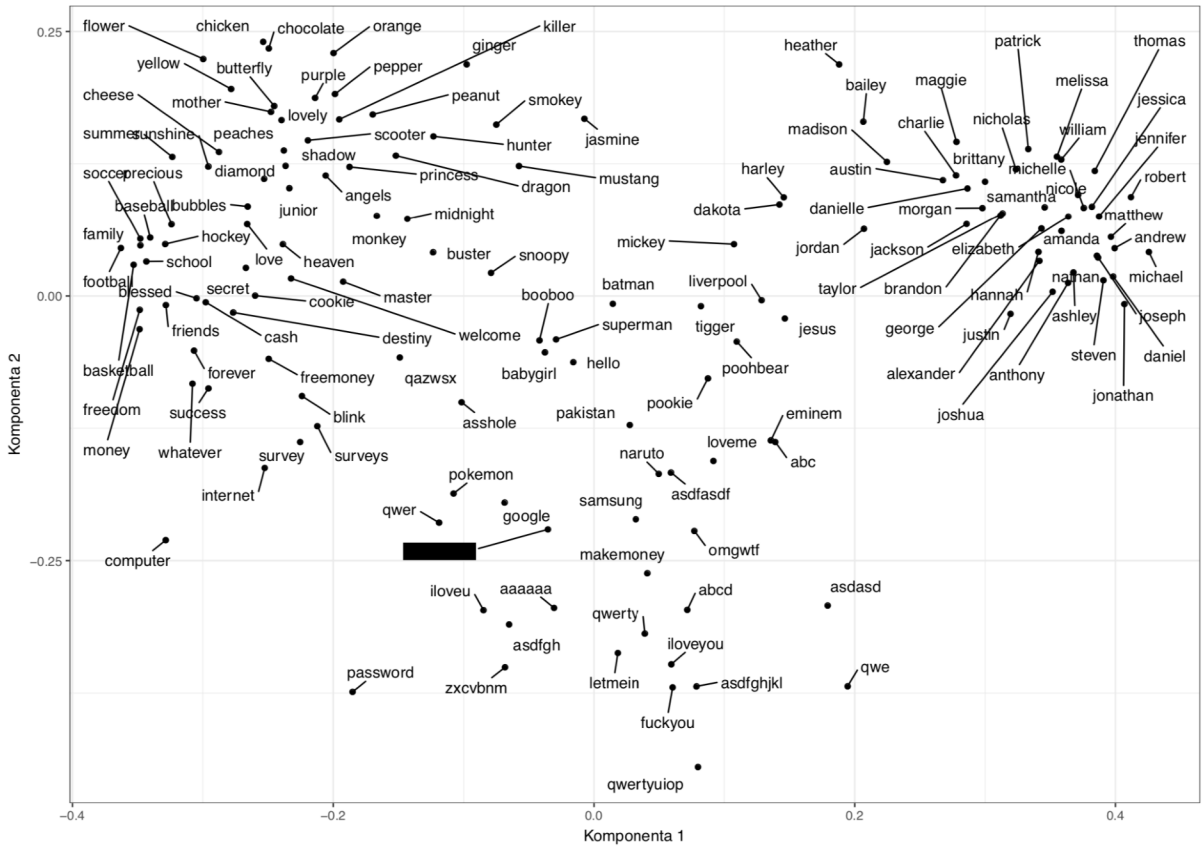
_graf_semantika_Kosatka

V tomto grafu vidíme vyšší míru heterogenity, avšak i z ní lze extrahovat jisté tematické celky. Pravá část je opět obsazena primárně jmény (výjimkou je pouze heslo <bigdaddy>), přičemž zde není tak jasné rozdělení mezi mužskými a ženskými jmény jako je tomu u datasetu Alpaka – to je však v případě anglofonního datasetu očekávatelné, jelikož angličtina nemá výraznou polaritu mužských a ženských jmen (srov. např. <taylor>) a ani gramaticky nerozlišuje maskulinum a femininum – z kontextu tedy nelze vždy jednoznačně určit, zda se v textu jedná o muže či ženu. Další tematicky ucelený shluk můžeme pozorovat v levém dolním rohu, kde se nachází slova týkající se sportu a s ním souvisejícími slovy; můžeme zde tak pozorovat hesla jako <soccer>, <hockey> aj. doplněné o <winter> nebo <vacation>. V levé horní části mapy pak převažují zvířata a barvy, což sémanticky dává smysl, uvážíme-li, že barvy jsou často kvality užité k deskripci zvířat.

Zaměříme-li se na střed mapy, zjišťujeme, že se zde nachází primárně sousloví. Otázka, která zde vyvstává, je, jakým způsobem proběhl sémantický embedding pro tato hesla. Vzhledem k tomu, že se tato hesla vyskytují v blízkosti sekvencí jako <qwerty>, <qazwsx> nebo <abcd>, lze předpokládat, že v jejich případě nelze hovořit o analýze sémantiky, jelikož z důvodu, že se jedná o sousloví neoddělená mezerou, nebylo možné je správně interpretovat. Je to poměrně pravděpodobná odpověď na otázku, proč se v blízkosti sebe vyskytují hesla jako např. <starwars>, <babygirl>, <applepie> a <qwerty>. Tato hesla jsou tím pádem více rozprostřená

v prostoru, netvoří ucelený shluk a významově jsou taktéž rozkolísaná. U tohoto datasetu také není možné jednoznačně určit vedoucí trend jako tomu bylo v případě Alpaky – ani jedna z vyobrazených skupin výrazně nepřevažuje v zastoupení.

Přesuňme se nyní ke třetímu a poslednímu analyzovanému datasetu: Krajta. Graf *_graf_semantika_Krajta* je vizualizací sémantiky v něm se vyskytujících hesel.



_graf_semantika_Krajta

Oproti předešlému datasetu Kosatka je zde možné sledovat lépe tematicky ucelené shluky a celkově jej tak rozdělit na tři primární skupiny. Stejně jako u předešlých dvou datasetů se v pravé části vyskytují jména, která jsou přibližně rozdělena na mužská a ženská – opět se jedná o anglofonní dataset, a tedy se hranice mezi mužskými a ženskými jmény částečně stírá. Levá část mapy je, analogicky k datasetu Kosatka, obsazena v horní části převážně zvířaty a barvami, zde však doplněná o jídlo (<cheese>, <peaches>) a i jiné přírodní motivy (<shadow>, <diamond>, <peanuts>); nalevo uprostřed pak nacházíme opět názvy sportů a s nimi související lexikum. Taktéž analogicky k přechozímu datasetu se ve středu mapy nacházejí zejména hesla tvořená souslovími společně s různými nenáhodnými sekvencemi znaků (jako např. „klávesnicové“ postupky); nacházíme zde tak blízko sebe hesla jako <makemoney>, <omgwtf>, <abcd> nebo <iloveyou>. Vzhledem k tomu, že v případě tohoto datasetu můžeme lépe sledovat, že takováto hesla tvoří méně koherentní shluky a významově jsou si vzdálená (srov. např. <iloveu>, <aaaaaa> a <letmein>), můžeme se o to více přiklonit k teorii navržené u podobných hesel v datasetu Kosatka, a tedy že sémantika zde nebyla správně interpretována.

Tato sémantická analýza jednotlivých datasetů přináší několik zajímavých pozorování ohledně významové složky lexikálních hesel. V prvním, nejjednodušším, případě ukazuje, jaké jsou obecné „tematické trendy“ 150 nejpoužívanějších hesel, přičemž zjišťujeme, že lze najít jistou shodu napříč všemi datasety. Nejvíce prominentní složkou jsou zde vlastní jména, která tvoří poměrně ucelené shluky a nacházíme je jak v datasetu ze striktně českého prostředí, tak ve zbylých dvou převážně amerických datasetech. Kromě jmen zde nacházíme názvy sportů, sportovních týmů či zvířat. Zároveň zde vidíme, že poměrně velké množství hesel je zde tvořeno více slovy, což pravděpodobně brání jejich správné sémantické interpretaci. Otázkou zůstává, nakolik by se výsledky změnily, pokud by byla tato hesla rozdělena na příslušné lexikální jednotky. V tomto případě by bylo však nutné dbát na zachování významu jednotlivých sousloví: uvážíme-li např. heslo <starwars>, bylo by nutné jej interpretovat jako název populární sci-fi filmové série, ne jako individuální slova <star> a <wars>. Takováto analýza by pak mohla poskytnout poněkud jasnější náhled na menší a potenciálně ucelenější tematická uskupení vyskytující se v datasetech.

Výsledky sémantické analýzy také ukazují na to, že výše diskutovaná zapamatovatelnost hesla je jedním z vedoucích faktorů při tvorbě „populárních“ hesel. Často se jedná o věci, o kterých lze uvažovat jako o koníčcích uživatele (např. názvy sportů), jmen buď přímo uživatelů nebo jejich rodinných příslušníků, oblíbených filmů či filmových postav atp. Kromě tohoto aspektu orientovaného *na uživatele* zde také můžeme sledovat aspekt orientace *na povahu platformy či poskytovaných služeb* – např. v datasetu Alpaka se jedná o heslo <nakup>, u Krajty <makemoney>, u Kosatky <showtime> a jim podobné. Takto jsou hesla jednodušeji zapamatovatelná, a tedy nekladou vysoké nároky na uživatele co do paměti a vynaloženého úsilí v tomto ohledu.

***

Doposavad jsme se v této práci zajímali o analýzy a indexy, které popisují dané datasety „odshora dolů“ a nabízejí tak na zkoumané datasety souhrnný náhled. Byly popsány nejčastější výskyty, délky hesel a další kvantitativní indexy, krom toho také sémantika populárních hesel. Jak jsme však mohli vidět pomocí indexu *množství hapaxů*, v datasetech je nemalé zastoupení hesel, která jsou uživatelsky unikátní. Tohoto faktu jsme si mohli povšimnout již dříve v tabulce *_tab_datasety_freq*, kde se i u více než milionového datasetu (Krajta) na dvacáté příčce pohybujeme ve stovkách výskytů. V následující části se tedy zaměříme právě na problematiku *unikátnosti* hesla a pokusíme se aplikovat jednoduché deskriptivní metody na soubory hapaxů jednotlivých datasetů.

Unikátní hesla

Oproti předcházejícím částem této práce, které nám poskytly exaktní úhel pohledu na situaci v datasetech co do vybraných kvantitativních vlastností, se nyní obrátíme ke specifické skupině hesel, a to k heslům unikátním. Indexem *množství hapaxů* jsme zjistili, že poměrně velkou část všech datasetů (přibližně polovinu) zabírají tzv. hapax legomena, tedy unikátní hesla (názvy *hapax legomena*, popř. *hapaxy*, a *unikátní hesla* jsou zde užívány synonymně). Odkloňme se nyní od zkoumání toho, co je *časté*, k tomu, co je *unikátní* – tedy k tomu, jak vypadají hesla, která nejsou v rámci datasetu nikým sdílena. Budeme se tak dále soustředit pouze na tři z vybraných datasetů: Alpaka, Kosatka a Krajta. RockYou je z těchto analýz diskvalifikován pro povahu seznamu, tedy že četnost výskytu každého hesla v něm je rovna jedné a hapaxy zde není možné identifikovat.

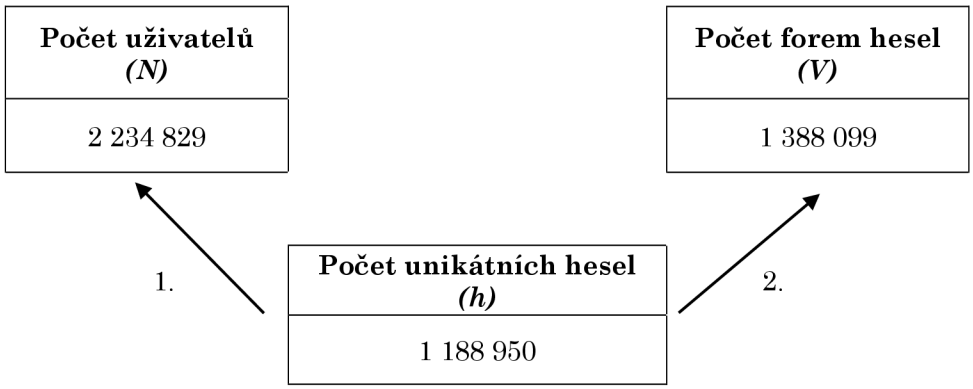
Než přejdeme k analýzám samotných hapaxů, podívejme se do detailu na to, jak velká část datasetu je unikátními hesly tvořena; jinými slovy: kolik lidí zakládajících si účet na jednotlivých online službách si vytvořili takové heslo, které nesdílejí s jiným uživatelem.

— Množství hapaxů v jednotlivých datasetech
Téma množství hapaxů v datasetech bylo již stručně nastíněno v části *Datasety*, která se věnuje základním kvantitativním indexům užitých k popisu vlastností zkoumaných datasetů. Pro ukotvení přesného množství hapaxů v datasetech se zde zaměříme na exaktnější ilustraci dané problematiky. Pro explikaci se zaměřme na případ datasetu Krajta.

Počet tokenů zde budeme nahlížet jako počet jednotlivých uživatelů zakládajících si účet na dané platformě; počet uživatelů (a tedy hesel) nechť je označen N. Počet typů zde reprezentuje počet jednotlivých forem hesel, která si uživatelé vybírají; jejich počet nechť je označen V. Počet hapaxů udává počet lidí, kteří si vytvořili unikátní heslo; tento počet nechť je označen h. Mezi uvedenými počty (N, V, h) platí vztah:

$$h \leq V \leq N$$

tedy že počet hapaxů je vždy menší nebo roven počtu typů, který je vždy menší nebo roven celkovému počtu hesel. Vizualizujme si nyní vztah hapaxů vůči tokenům a vůči typům (viz *_diag_typ-token_hapaxy*):



_diag_typ-token_hapaxy

Vztáhneme-li počet unikátních hesel h k celkovému počtu uživatelů N (viz případ 1. v *_diag_typ-token_hapaxy*):

$$N - h = 2\,234\,829 - 1\,188\,950 = 1\,045\,879$$

zjišťujeme, že celkem 1 045 879 uživatelů má heslo s frekvencí ≥ 2, tedy že takové množství hesel je sdíleno alespoň dvěma uživateli. Dále, vztáhneme-li počet unikátních hesel k typům hesel (viz případ 2. v *_diag_typ-token_hapaxy*):

$$V - h = 1\,388\,099 - 1\,188\,950 = 199\,149$$

zjišťujeme, že mezi nimi existuje 199 149 takových, která jsou sdílena více uživateli. Znamená to tedy, že celkem 1 045 879 tokenů hesel je realizováno „pouze" 199 149 různými formami (typy).

Díky těmto poznatkům můžeme o datasetu Krajta vyvodit následující zjištění. Z tabulky v předešlé části (*Základní kvantitativní indexy datasetů*: *_tab_hapaxy*) víme, že procentuální zastoupení hapaxů mezi všemi uživateli je 53,20 %. To značí, že zbylých 46,8 % uživatelů sdílí heslo s alespoň jedním dalším uživatelem. Výše provedeným výpočtem jsme zjistili, že existuje výrazný nepoměr mezi množinou typů hesel a množinou uživatelů – vydělíme-li počet hesel uživatelů, jejichž heslo má frekvenci ≥ 2, počtem neunikátních forem hesel, tedy:

$$\frac{N - h}{V - h} = \frac{1\,045\,879}{199\,149} \approx 5{,}25$$

zjišťujeme, že na jedno neunikátní heslo připadá v průměru ~5 uživatelů. Jinými slovy lze říci, že každý uživatel, který sdílí heslo s někým jiným, jej v průměru sdílí s přibližně čtyřmi dalšími lidmi. Tímto je dataset v podstatě rozdělen na dvě poloviny, kdy jedna polovina je zcela unikátní, kdežto ta druhá je reprezentována nevelikým počtem opakujících se typů. Podívejme se nyní na situaci ve zbylých dvou datasetech (viz *_tab_vypocet-prumeru).*

Alpaka	Kosatka
$N = 766\,421$ $V = 381\,908$ $h = 333\,247$ $\frac{N-h}{V-h} = \frac{766\,421 - 333\,247}{381\,908 - 333\,247} \approx 8{,}90$	$N = 590\,298$ $V = 413\,972$ $h = 357\,046$ $\frac{N-h}{V-h} = \frac{590\,298 - 357\,046}{413\,972 - 357\,046} \approx 4{,}10$

_tab_vypocet-prumeru

V případě Kosatky se jedná o ~4 uživatele na jedno neunikátní heslo, což je počet poměrně blízký Krajtě, která má ~5 uživatelů/neunikátní heslo – o to víc uvážíme-li, že Krajta je větší, a tedy je tam i vyšší pravděpodobnost výskytu neunikátních hesel: každý nový uživatel má s vyšším počtem již existujících hesel možnost se trefit do hesla, které si již vytvořil jiný uživatel. Zajímavá je v tomto ohledu situace datasetu Alpaka, kde připadá na jedno neunikátní heslo bezmála 9 uživatelů. Jedná se přibližně o dvojnásobek Kosatky, která mu je nejbližší co do množství tokenů. V tomto případě se nejspíše jedná o manifestaci toho, že Alpaka je dataset prolámaný z hašů[10]. Pokud tomu tak je, byl by pro nás počet uživatelů připadajících na jedno neunikátní heslo zajímavým indexem.

Počet uživatelů, kteří průměrně sdílejí jedno heslo může potenciálně být indikátorem toho, zda se jedná o dataset, který unikl v plaintextu, nebo o dataset, který byl lámaný z hašů. Jedno z možných vysvětlení je to, že u datasetů, které byly „prolámány“ z hašů, je možné, že při prolamování bylo možné získat především jednodušší hesla a hesla, která jsou sdílena více uživateli, a to např. pomocí slovníkového útoku, kterému nahrává do karet konvence užití určitých hesel. Oproti tomu uživatelé, kteří si vytvořili silnější heslo (unikátní, dostatečně dlouhé a užívající co největší množiny symbolů) zůstali ochráněni, a tedy jejich hesla nejsou obsažena v seznamu prolomených hesel. Tímto by byla zapříčiněna převaha konvenčních (a mimo to také jednodušších) hesel v datasetu, a tedy může být i ukazatelem toho, že dataset byl prolomen z hašů. Zda tomu tak však je, nelze usuzovat ze zde provedené analýzy, a to z důvodu příliš malého vzorku – pro potvrzení by bylo třeba provést extenzivní výzkum, kde by byly reprezentativně zastoupeny jak datasety prolomené, tak i ty plaintextové, na základě kterých by bylo potenciálně možné určit hranici, která by indikovala pravděpodobnost toho, že daný dataset neunikl v plaintextu, ale byl prolomen z hašů. Zde lze o této možnosti pouze spekulovat.

Skutečnost, že až polovina uživatelů zvolí stejné heslo jako jiní uživatelé, nese jisté implikace i pro útočníka, který by se snažil o převedení seznamu hašů na plaintextová hesla jejich prolomením. Výše jsme mohli vidět, že existuje výrazný nepoměr mezi počtem uživatelů s neunikátními hesly ($N - h$) a forem hesel, kteří takoví uživatelé volí ($V - h$). Lze tak uvažovat o přímé úměře mezi popularitou hesla a

[10] Pro nedostatečně velký vzorek zkoumaných datasetů (především těch prolomených, které jsou zde reprezentovány pouze Alpakou) nelze tuto skutečnost tvrdit s naprostou jistotou.

pravděpodobností toho, že se takové heslo vyskytuje ve slovníku, který může být útočníkem použit pro prolomení daného seznamu hašů (vzpomeňme, že takové slovníky obsahují miliony, ne-li více, hesel), a tedy můžeme předpokládat, že až polovina zde analyzovaných datasetů by byla prolomitelná slovníkovým útokem. Pro hlubší náhled na málo frekventovaná hesla a to, kolik procent datasetů tvoří viz **přílohu 2**, kde nabízíme odpověď na otázku *kolik procent datasetu zabírají málo frekventovaná hesla*. Konkrétněji se zde zabýváme hesly s frekvencí 1–10, která, dle provedených analýz, náleží ~61–87 % uživatelů jednotlivých služeb a zabírají ~98–99 % *podob* hesel.

Nyní tedy víme, jaké je množství hapaxů v jednotlivých datasetech a shrnuli jsme i základní implikace tohoto počtu. Vyvstává zde však další otázka, kterou se budeme v této části dále primárně zabývat, a to *co činí unikátní hesla unikátními*.

— Délky hapaxů

Než se přesuneme ke specificky orientovaným analýzám, zaměřme se na nejjednodušší deskriptivní index, a to na délku unikátních hesel, tedy počet znaků, kterými jsou individuální hesla tvořena. Podobně jako v případě celých datasetů se zaměříme na průměr, medián a 95% interval počtu znaků v rámci hesla. Obecně je cílem pozorovat, zda, a popřípadě jak, se hapaxy v tomto ohledu odchylují od trendů v datasetu. Potenciálně zde lze očekávat, že unikátní hesla budou delší, a to na základě prosté úvahy: čím delší sekvenci znaků vytvoříme, tím spíše se na dané sekvenci neshodneme s někým jiným.

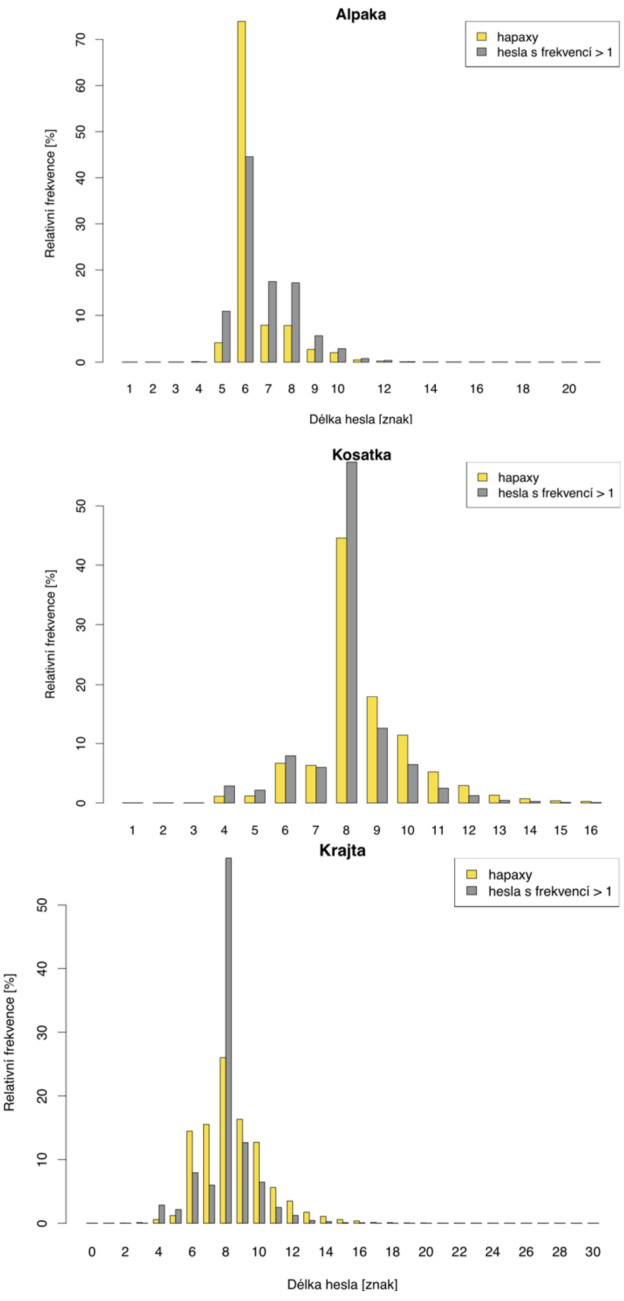
Souhrnný náhled na výsledky měření je poskytnut v tabulce *_tab_delky-hapaxy*; pro srovnání jsou v závorkách uvedeny výsledky pro celý dataset (z tabulky *_tab_datasety_kvant* uvedené výše v této práci).

Délky hapaxů [znak]			
Dataset	**Průměr**	**Medián**	**95% interval**
Alpaka	6,41 (6,54)	6 (6)	5–10 (5–10)
Kosatka	8,56 (8,27)	8 (8)	6–13 (5–12)
Krajta	8,48 (7,96)	8 (8)	6–13 (5–13)

_tab_delky-hapaxy (v závorce uvedeny hodnoty pro celý dataset)

V této tabulce vidíme, že zatímco medián zůstává ve všech případech stejný, průměry a intervaly se mírně liší. Nejméně se oproti zbytku datasetu liší hapaxy Alpaky, kde se pouze snížil průměr o několik desetin znaku, přičemž 95% kvantil zůstává stejný – tedy 95 % všech délek stále spadá do rozmezí 5 a 10 znaků, stejně jako v případě celého datasetu. Oproti tomu pro Kosatku i Krajtu je průměr vyšší (avšak opět pouze o desetiny znaku) a spolu s ním i kvantil, kdy 95 % délek spadá do o něco vyšších rozmezí. Uvedené rozdíly ukazují na to, že unikátní hesla jsou v porovnání se zbytkem datasetu o něco delší (s výjimkou Alpaky), avšak nelze tvrdit, že by se jednalo o markantní rozdíl. Zaměřme se nyní tedy na srovnání *unikátního* s *neunikátním*.

Pro exaktnější náhled na to, jak se unikátní hesla z hlediska délky liší od těch neunikátních, byly datasety typů[11] rozděleny na dvě části: *hapaxy* a *hesla s frekvencí* ≥ *2*. Cílem je změřit délku pro obě tyto části separátně, přičemž očekáváme, že unikátní hesla budou mít četnější zastoupení ve vyšších délkách. Teno předpoklad je založen na výše nastíněné úvaze, že s narůstající délkou hesla narůstají i možnosti toho, jaké kombinace (popř. i jaké množiny znaků) se zde mohou vyskytovat, a tedy klesá pravděpodobnost, že se na jednom hesle shodne více uživatelů. Výsledky měření pro každý z datasetů byly vizualizovány pomocí sloupcových grafů.



_grafy_ne-hapaxy

Grafy vyobrazené v souboru grafů *_grafy_ne-hapaxy* nám poskytují srovnání délek unikátních a neunikátních hesel. Důležitá vlastnost, kterou zde můžeme pozorovat, je nejčastější délka hesla a její zastoupení. Zjišťujeme zde, že v případě Kosatky a Krajty (tedy plaintextových datasetů) je nejčastější délka z největší části zastoupena neunikátními hesly. To značí, že pokud si uživatel na daných službách vytvořil heslo o nejčastější délce, většinou se trefil do hesla, které má i jiný uživatel. Oproti tomu v případě datasetu Alpaka (který byl prolomen z hašů) vidíme, že nejčastěji zastoupená délka 6 je tvořena primárně hapaxy: tedy pokud si uživatel vytvořil heslo o nejčastější délce, nejspíše vytvořil heslo unikátní. V tomto ohledu můžeme opět uvažovat o manifestaci anomálie Alpaky. Jak bylo zmíněno v kapitole *Síla hesla*, kratší hesla jsou obecně náchylnější k prolomení (ačkoli do prolamování vstupuje vícero faktorů) – tím je možné vysvětlit nejčastější délku právě 6. Při kvalitativním prozkoumání tohoto datasetu zjišťujeme, že se v této délce převážně jedná o zdánlivě náhodné sekvence jako např. <rPTFPL>, <hTKCjh> nebo <RhlHGS>. Zaměříme-li se obecně na hesla o délce 6 v datasetu Alpaka, zjišťujeme, že zabírají celkem 70 % celého datasetu typů a zároveň, že velká část z nich jsou náhodné sekvence kombinující malá a velká písmena

[11] S typy zde pracujeme, jelikož ve středu zájmu nyní nejsou uživatelé, ale různé podoby hesel

s čísly – což jsou právě zde vyobrazené hapaxy. Takovéto výskyty lze interpretovat jako hesla generovaná buď přímo platformou, aplikací, nebo jiným manažerem hesel[12]. Každopádně zde vidíme, že i takováto náhodná hesla nejsou dostatečně bezpečná, jelikož byla prolomena.

Zaměříme-li se na rozdělení délek unikátních a neunikátních hesel, to bylo testováno pomocí χ^2 (simulovaná p-hodnota, B = 5 000); posoudíme-li výsledky:

- Alpaka: χ^2 = 18 343; p-hodnota = 0,0002***;
- Kosatka: χ^2 = 7 176,2; p-hodnota = 0,0002***;
- Krajta: χ^2 = 34 322; p-hodnota = 0,0002***

zjišťujeme, že rozdělení délek neunikátních a unikátních hesel daných datasetů jsou odlišné. Jinými slovy zde existuje systém řídící délku neunikátních hesel a, separátně, systém řídící délku unikátních hesel.

— Unikátnost hapaxů

Hlavní otázkou, kterou s sebou přináší zjištění, že až polovina hesel, které si uživatelé na různých platformách vytvoří, je unikátní, je to, *do jaké míry* se tato hesla od sebe odlišují. Obraťme nyní naši pozornost k tomu, kolik hesel je unikátní v rámci datasetu a kolik z nich je unikátní obecně – tedy nejsou sdílena napříč zde zkoumanými datasety.

Pro tři zkoumané datasety byla vytvořena matice zobrazující průniky hapaxů, tedy to, kolik hapaxů je mezi danými dvojicemi datasetů sdíleno – viz *_tab_pruniky*.

Matice průniků			
	Alpaka	**Kosatka**	**Krajta**
Alpaka	–		
Kosatka	2 413	–	
Krajta	6 879	18 388	–
Všechny	467		

_tab_pruniky

Tato matice vyobrazuje počet sdílených hapaxů mezi jednotlivými dvojicemi datasetů a zároveň i mezi všemi datasety zároveň. Vidíme zde, že nejvíce společných unikátních hesel se nachází ve dvojici Kosatka–Krajta. To je poměrně očekávatelné, uvážíme-li, že tyto datasety mají nejvíce společných vlastností, a to především to, že jsou plaintextové povahy a pochází z podobného demografického prostředí. Oproti tomu průniky s Alpakou jsou obecně méně zastoupeny, což je s nejvyšší pravděpodobností způsobeno tím, že hapaxy v Alpace jsou poměrně specifické povahy (viz přecházející část *Délky hapaxů*). Anomálie Alpaky nejspíše také způsobuje to, že průnik hapaxů všech datasetů obsahuje „pouhých" 467 hesel. Nejzajímavějším poznatkem tak zůstává průnik mezi Kosatkou a Krajtou, jelikož právě tyto dva datasety jsou v rámci našeho výzkumu reprezentativní (tedy jsou

[12] Vzhledem k tomu, že se zde jedná o nemalý počet hesel, přikláníme se spíše k možnosti hesel generovaných platformou či jiným způsobem společným takovému množství uživatelů.

plaintextové povahy). Podíváme-li se však na procento, které jimi sdílené hapaxy zastupují v rámci všech hapaxů daných datasetů, jedná se o pouhé jednotky: v případě Krajty je to 1,55 % a v případě Kosatky 5,52 %; v rámci celých datasetů je to pak ještě méně: 3,12 % v Kosatce a 0,82 % v Krajtě – jinými slovy tedy pouze jednotky procent hapaxů jsou sdíleny napříč datasety. Obecně díky tomuto měření vidíme, že napříč platformami existuje pouze malá shoda v unikátních heslech. Tyto výsledky samozřejmě nelze zobecnit, popř. z nich vyvozovat konkrétní závěry, avšak minimálně v případě zde zkoumaných datasetů platí to, že pokud si uživatelé těchto služeb založili heslo, jen v jednotkách procent případů jej sdíleli s někým jiným z jiné platformy.

Tyto poznatky nás však nedostávají o moc blíže k odpovědi na to, jak moc jsou hapaxy unikátní. Zaměřme se tedy na rozdíly, které mohou být způsobeny užitými znaky, resp. užitými množinami znaků; dále budeme zkoumat to, jakým způsobem se množství hapaxů v datasetu změní, odstraníme-li jednu z množin znaků. Předpoklad, ze kterého zde vycházíme, je možnost toho, že by se hesla lišila pouze např. užitím velkých písmen nebo přidáním sekvence číslic či speciálního symbolu na konec (popř. začátek) hesla. Pokud bychom takto zkoumali např. hesla: <Password>, <password123> a <password!>, vidíme, že se ve všech případech jedná o modifikace jednoho a toho samého základu.

Zaměřme se nyní tedy na tyto rozdíly: pro každý z datasetů byly provedeny výše popsané změny. Postup srovnání byl následovný: ze souboru hapaxů byl vždy odstraněn jeden z výše uvedených rysů (tj. velká písmena, číslice, speciální znaky), byl spočítán nový počet unikátních hesel a rozdíl mezi původním a novým počtem hapaxů byl vyjádřen procentuálně. Pro výsledky těchto měření viz souhrnnou tabulku *_tab_rozdily-odstraneni.*

Rozdíly před a po odstranění množin znaků							
	Původní počet hapaxů	**Po převodu na malá písmena**	**Rozdíl [%]**	**Po odstranění číslic**	**Rozdíl [%]**	**Po odstranění speciálních symbolů**	**Rozdíl [%]**
		Heslo123! → heslo123!		*Heslo123! → Heslo!*		*Heslo123! → Heslo123*	
Krajta	1 188 950	1 179 648	0,78	683 452	42,52	1 185 339	0,30
Kosatka	357 046	354 894	0,60	238 793	33,12	356 890	0,04
Alpaka	333 247	331 696	0,47	276 542	17,2	333 119	0,04

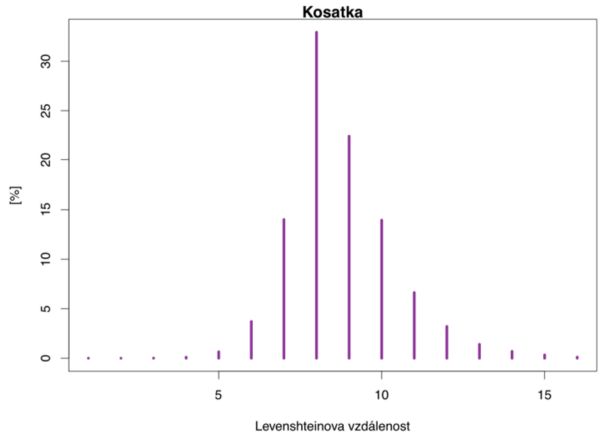
_tab_rozdily-odstraneni

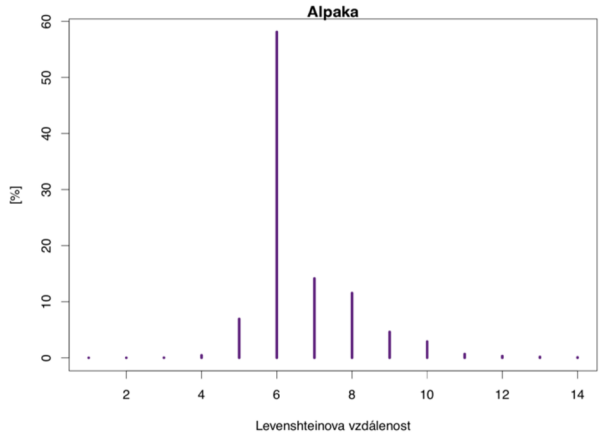
V této tabulce vidíme nejprve původní počet hapaxů – tedy takový, jaký se v datasetech vyskytuje před modifikací. Ten je zde uveden primárně pro srovnání, jelikož v dalších sloupcích můžeme sledovat, jak se počet hapaxů mění při odstranění každého z daných rysů. Vidíme zde, že zatímco smazání rozdílu mezi malými a velkými písmeny a odstranění speciálních symbolů má na unikátnost hapaxů téměř zanedbatelný vliv (pohybujeme se zde ve zlomcích procent), odstranění číslic má za následek snížení počtu unikátních hesel až na téměř polovinu (viz dataset Krajta, kde je naměřena hodnota 42,52 %). To tedy znamená, že pokud si uživatel vytvoří unikátní heslo, poměrně často bude toto heslo unikátní právě užitím číslic.

U tématu rozdílnosti užitých znaků ještě dále setrváme. Díky výše uvedenému měření jsme získali náhled na to, jaké množiny znaků nejčastěji činí unikátní hesla unikátními. Zaměřme se dále na obecnější rozdíl, který částečně vychází z předešlé analýzy, a to na otázku *nakolik se od sebe unikátní hesla liší*. Pro tuto analýzu využijeme Levenshteinovu vzdálenost.

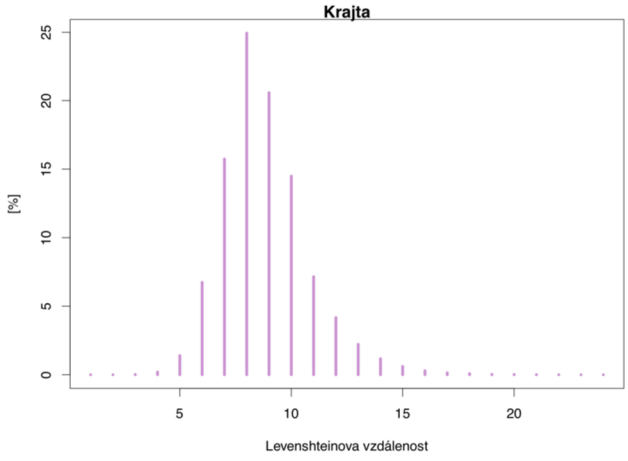
Levenshteinova vzdálenost (známá také jako editační vzdálenost) byla navržena Vladimirem Levenshteinem (1966); tato vzdálenost bere v potaz dva řetězce znaků a vyjadřuje, kolik změn je potřeba provést, aby dané dva řetězce byly totožné. Změny, které jsou brány v potaz, jsou odstranění či přidání znaků a substituce; pokud bychom tedy srovnávali řetězce <Heslo> a <heslo123>, jejich vzdálenost by byla rovna čtyřem, jelikož je nutno (I) změnit velké písmeno na malé, (II) přidat tři číslice. Pomocí této vzdálenosti lze v našem případě zjistit, o kolik *změn* se od sebe unikátní hesla nejčastěji liší; resp. pokud si uživatel vytvoří unikátní heslo na dané službě, nakolik je toto heslo odlišné od ostatních unikátních hesel. Vzhledem k tomu, že víme, že unikátní hesla se často liší v užití čísel, na výstupu této analýzy očekáváme rozdíly v hodnotách nižších, než je průměrná délka hesla daného datasetu, a to jelikož předpokládáme, že rozdíly jsou často dány právě přidáním číslic.

Postup provedení této analýzy v datasetech byl následující. Nejprve bylo z každého datasetu náhodně vybráno 10 000 hapaxů, které vstoupily do analýzy. Celé soubory hapaxů nebyly použity z důvodu kapacity paměti počítače – byl tak zvolen reprezentativní počet, pro který bylo možno výpočet vzdálenosti provést. Následně byla vypočtena vzdálenost jednotlivých hapaxů na principu „každý s každým“, a to za pomoci matice, do které se postupně ukládaly vzdálenosti jednotlivých dvojic, přičemž vzdálenost pro každou dvojici byla počítána pouze jednou (tedy pokud se již v matici vyskytoval výsledek pro dvojici *AB*, *BA* bylo ignorováno). Hodnoty z výsledných matic pro jednotlivé datasety jsou zde vizualizované v souboru grafů *_graf_levenshtein*.





Na ose *x* je v grafech vyjádřena Levenshteinova vzdálenost; na ose *y* je její procentuální zastoupení. Zajímavé zjištění, které nám tyto grafy zprostředkovávají, je fakt, že nejčastější vzdálenost hapaxů je ve všech případech rovna jejich průměrné délce: v případě Alpaky 6, v případě Kosatky a Krajty 8. Jinak řečeno, abychom z jednoho hapaxu vytvořili jiný,



_graf_levenshtein

nejčastěji musíme změnit osm znaků. Nejspíše se tedy nejčastěji nejedná o pouhé přidání čísla, substituci znaku za symbol založené na grafické podobnosti (např. použití zavináče místo hlásky *a* nebo použití nuly místo *o*) a podobné jednoduché změny, jak jsme původně očekávali – vzhledem k tomu, že osm je i průměrná délka unikátních hesel, lze konstatovat, že nejčastěji je nutné změnit celý řetězec. Tyto výsledky vyvrací naše očekávání, že unikátní hesla jsou nejčastěji unikátní díky substitucím či jednoduchým změnám a poukazují na to, že unikátnost hesel je komplexnější záležitost.

***

Souhrnně lze říci, že unikátnost hapaxů je záležitost více komplexní, než jsme očekávali. Základní údaje o délkách hapaxů uvedené na začátku této části odhalily to, že délka není hlavním důvodem unikátnosti hapaxů, jelikož se v tomto ohledu příliš neliší od zbytku datasetu. Přestože jsou zde naměřené hodnoty o něco vyšší než pro dataset jako celek, nejsou tyto rozdíly markantní. Způsobeny jsou nejspíše efektem toho, že s narůstající délkou hesla narůstá i četnost možností znaků, které se dále mohou vyskytnout, úměrně k čemuž klesá pravděpodobnost toho, že se dva uživatelé (popř. i více) na dané sekvenci znaků shodnou. Tento fakt lze ostatně podpořit i naměřeným 95% kvantilem délek, jehož spodní hranice je v případech obou reprezentativních datasetů (Kosatka, Krajta) posunuta o jeden znak výše.

Co se pak týká unikátnosti hapaxů – bylo zjištěno, že po odstranění čísel klesne počet hapaxů až o více než třetinu (v případě Krajty je to ~42 %). Tento počet však může být zavádějící. Na jednu stranu vede k úvaze, že hesla tvořená sekvencí alfabetických znaků se od sebe odlišují pouze užitím specifické numerické sekvence (např. *heslo1234* a *heslo789*). Ačkoli tato úvaha není zcestná, nelze zapomenout na to, že se v datasetu objevuje poměrně velké množství čísel jako takových a některá hesla jsou tvořena výhradně jimi – odstraníme-li tedy z datasetu veškerá čísla, nevyhnutelně přijdeme i o některá hesla jako celek, a tedy naroste i procentuální rozdíl množství hapaxů po odstranění číslic. Tento rozdíl pak nenastává v případě velkých písmen, jelikož ta jsou substituována na písmena malá, ani v případě speciálních znaků, jelikož jen výjimečně jsou hesla tvořena výhradně jimi. Naměřená Levenshteinova vzdálenost hapaxů pak tuto úvahu v jistém ohledu podporuje, jelikož vidíme, že nejčastější vzdálenost je právě průměrná délka hapaxů – není nižší, jak jsme před provedením měření očekávali, a jak by tomu nejspíše bylo v případě, kdy by se unikátní hesla lišila pouze přidáním specifických čísel. Výsledky těchto měření vedou k závěru, že pokud je heslo hapaxem, je ve větším množství případů zcela unikátní; jinak řečeno: hapaxy nejspíše nejsou z velké části zastoupeny modifikacemi několika opakujících se typů hesel.

Lze takto dojít k závěru, že to, co činí heslo unikátním, je specifické rozložení znaků. V následující části tedy nahlédneme hesla právě z této stránky, avšak než abychom se vydávali cestou jejich grafické povahy (tedy malá a velká písmena), vydáme se cestou motivovanou lingvisticky: pokusíme se hesla popsat z hlediska jejich „sylabické struktury“, respektive z hlediska rozložení konsonantů a vokálů. Tento přístup má oporu i ve výsledcích měření provedených v části *Datasety*, kde můžeme pozorovat, že konsonanty a vokály jsou prominentní skupiny znaků tvořící hesla (viz *_tab_datasety_kvant*), a tedy lze očekávat, že jejich užití se bude řídit určitými pravidly – ať už tato pravidla vyvstávají z lexikálních vlastností hesel, nebo jiných (nepopsaných) aspektů majících vliv na výslednou podobu hesla.

Sylabická struktura hesel

V této části práce vycházíme (I) z poznatku, že unikátnost hesla nespočívá pouze v modifikaci (tedy že unikátní hesla nejčastěji nejsou obměnou častějších typů), ale že se nejspíše jedná o obecně unikátní sekvence znaků; (II) z předpokladu, že nemalá část datasetu je tvořena lexikem (ať už se jedná o jedno slovo, sousloví či frázi). Pokusíme se zde tak nahlédnout hesla z hlediska jejich kombinatoriky, přičemž sledované množiny jsou motivovány jazykově: budeme sledovat to, co bychom z hlediska jazyka popsali jako sylabické (či slabičné) struktury; zde se jedná obecněji o rozložení konsonantů a vokálů[13]. Uvažme kupříkladu hesla <fisher>, <milton> a <redcat> ze souboru Kosatka – ačkoli jsou na první pohled odlišná (kromě své délky a sady užitých znaků – malých písmen), sdílí pozice konsonantů (C) a vokálů (V) a jsou tedy vyjádřitelná stejnou strukturou:

fisher
milton } CVCCVC
redcat

Vidíme zde tedy, že i zdánlivě unikátní heslo může potenciálně sdílet jisté vlastnosti s jinými hesly v datasetu. V následující části prozkoumáme soubory hesel z hlediska jejich sylabických struktur, a to s cílem získat nový úhel pohledu na jejich unikátnost a popřípadě se přiblížit odpovědi na otázku *co činí heslo unikátním*. Pro získání obecného náhledu na situaci v datasetech co do sylabických struktur zde budeme dále pracovat s celými datasety, nikoli pouze s unikátními hesly.

— Zpracování dat

V následující části se budeme zajímat primárně o strukturu výstavby hesel, a tedy budeme pracovat s jejich typy, nikoli s tokeny. Z každého souboru dat byly nejprve vytvořeny typy (tj. každé heslo má nyní pouze jeden výskyt – toto nebylo nutné provést pro soubor RockYou, který má sám o sobě povahu seznamu), v nichž jsou provedeny následující změny: veškeré konsonanty jsou nahrazeny C, vokály V, čísla D a speciální znaky S; není zde bráno v potaz rozlišení malých a velkých písmen. Jako problematický se při této substituci jeví ypsilon. V češtině se ve většině případů vyskytuje v roli vokálu; výslovnostní rozdíl mezi ypsilonem a měkkým i dnes již prakticky neexistuje (až na jistá nářeční specifika). Výjimkou jsou zde pouze přejatá slova jako například yorkshirský, yard, yosemitský, yetti atp., kde ypsilon zůstává věrný původnímu jazyku a vyslovujeme jej jako konsonant [j]. Takovéto tvary jsou však zřídkavé, a tak na ně při substituování hesel z českého prostředí nebylo nutné brát zřetel. Poněkud jiná situace je však v anglickém jazyce, který je prominentní ve třech zkoumaných souborech hesel.

[13] O slabikách zde lze hovořit pouze s jistou opatrností – slabiky a priori počítají s lexikem, avšak ne všechna hesla jsou lexikální povahy. I zde však očekáváme typické struktury. Pokud zde tedy budeme hovořit o sylabických strukturách, myslíme je v přeneseném slova smyslu jako kombinatoriku konsonantů a vokálů.

Povaha ypsilonu je vázána na jeho bezprostřední kontext, kde za některých okolností funguje jako vokál, za jiných jako konsonant. Obecně lze definovat tři typické situace, kdy je ypsilon vokálem (adaptováno ze slovníku Merriam-Webster.com, 2018 [cit. 17-08-2021]):

1) pokud se ve slově nenachází žádný jiný vokál (např. *my*, *gym*),
2) pokud se nachází na konci slova nebo slabiky (např. *candy*, *bicycle*) kromě případů, kdy se vyskytuje na konci slova za vokálem – v takovém případě je součástí diftongu (např. *play*, *day*),
3) pokud je uprostřed slabiky (např. *system*).

Tyto základní principy nám poskytují jistý základ pro definování pravidel substituce v anglickém jazyce. Přestože je možné, že se nám nepodaří zohlednit veškeré specifické případy a kontexty, ve kterých se ypsilon v rámci souborů hesel může vyskytnout, pokusíme se tímto ideálnímu stavu co nejvíce přiblížit. Tabulka *_tab_pravidla-Y* shrnuje pravidla, která byla implementována pro substituci ypsilonu.

Aplikovaná pravidla pro ypsilon v angličtině[14]			
místo výskytu ypsilonu	**náhrada**	**schéma**	**příklad**
mezi konsonanty	vokál	CyC → CVC	*gym*
mezi konsonantem a číslem	vokál	CyD → CVD	*kelly123*
mezi vokálem a číslem	vokál (=> diftong)	VyD → VVD	*play99*
na začátku hesla před vokálem	konsonant	yV → CV	*yesterday*
na začátku hesla před konsonantem	vokál	yC → VC	*ylem*
na konci hesla za konsonantem	vokál	Cy → CV	*kelly*
na konci hesla za vokálem	vokál (=> diftong)	Vy → VV	*play*
veškeré ostatní výskyty	konsonant	y → C	–

_tab_pravidla-Y

Jsou zde zohledněny kontexty nejen s vokály a konsonanty, ale zároveň i s čísly, která se mohou v rámci hesla vyskytovat. V nezachycených případech, jako např. v sekvencích, které nemají lexikální povahu, je poté ypsilon substituován jako konsonant, jelikož do této kategorie formálně spadá. Díky implementaci těchto pravidel poté získáváme reálnější náhled na schémata sylabických struktur hesel, která pocházejí z anglofonního prostředí.

— Struktury v datasetech

Po převedení datasetů na sylabické struktury zde potenciálně očekáváme nalezení nejčastějších kombinací, kterými jsou hesla tvořena. Nemusí se zde nutně jednat o nalezení lexikálních struktur – i v případě náhodných sekvencí bychom zde měli získat náhled na nejčastěji využívané skupiny a jejich kombinace. Nejprve po substituci byla provedena frekvenční analýza – 20 nejčastějších struktur lze sledovat v souhrnné tabulce *_tab_frekvence-struktur*.

[14] Substituce nebere v potaz výskyty ypsilon vedle speciálních symbolů (S). Takovéto výskyty jsou v souborech hesel natolik vzácné, že je nebylo nutné zohlednit: procentuální zastoupení speciálních symbolů je v souborech dat v rámci jednotek. V případě práce s daty z platformy, která by explicitně vyžadovala užití speciálního symbolu v hesle, a tedy by zastoupení S bylo mnohonásobně vyšší, by bylo nutné pro symboly dodat stejná substituční pravidla jako pro čísla.

Nejčastější struktury					
#	**Struktura**	**Frekvence**	**#**	**Struktura**	**Frekvence**
Alpaka					
1	DDDDDD	18 353	11	CCCCVC	4 733
2	CCCCCC	16 479	12	CVCVCVC	4 655
3	DDDDDDDD	8 241	13	CCCCCV	4 633
4	CVCCVC	7 081	14	CCCCDC	4 063
5	CVCVCV	6 125	15	CVCVCCV	4 027
6	CCCVCC	4 891	16	CCCCCD	3 996
7	CCVCCC	4 874	17	CCCDCC	3 923
8	CVCCCC	4 751	18	CCDCCC	3 880
9	VCCCCC	4 739	19	DCCCCC	3 867
10	CVCVC	4 734	20	CDCCCC	3 779
Kosatka					
1	DDDDDDDD	9 485	11	CVCCVCVC	2 958
2	CVCCVCDD	9 369	12	CVCVCVCC	2 943
3	CVCVDDDD	5 033	13	CVCVCVDD	2 856
4	CVCCCVCC	4 761	14	CVCCVDDDD	2 789
5	CVCCDDDD	4 740	15	CVCCVVDD	2 751
6	DDDDDD	3 913	16	CCDDDDDD	2 401
7	CVCVCVCV	3 615	17	CVCVCDDD	2 363
8	DDDD	3 357	18	CVCCCVCV	2 357
9	CVCCVDDD	3 193	19	CVCCVCDDDD	2 282
10	CVCCVC	3 160	20	CVCVCCVC	2 269
Krajta					
1	DDDDDD	50 240	11	CVCVCVCV	8 058
2	DDDDDDDD	31 193	12	CVCVCVDD	7 905
3	DDDDDDD	24 338	13	CVCCDDDD	7 838
4	CVCCVCDD	19 564	14	CCDDDD	7 667
5	CVCCVC	15 599	15	CVCVCDD	7 660
6	DDDDDDDDDD	12 442	16	CVCCVDDD	7 198
7	CVCVDDDD	12 152	17	DDDDDDDDD	6 873
8	CVCCVDD	10 127	18	CVCCVDDDD	6 800
9	CVCVDD	9 474	19	CVCCVVDD	6 597
10	CVCVCV	9 026	20	CVCCVCD	6 412
RockYou					
1	DDDDDDD	487 429	11	CVCCVDD	88 372
2	DDDDDDDDDD	478 196	12	CVCVDD	87 330
3	DDDDDDDD	428 296	13	CVCVCVCV	85 638
4	DDDDDD	390 529	14	CCDDDD	74 374
5	DDDDDDDDD	307 532	15	CVCVCVC	69 313
6	CVCCVC	112 971	16	CVCVCDD	65 970
7	CVCCVCDD	108 557	17	CVCVCVDD	64 484
8	DDDDDDDDDDD	107 862	18	CCCDDDD	64 236
9	CVCVDDDD	100 894	19	CVCDDDD	62 528
10	CVCVCV	88 777	20	CVCCVCV	60 288

_tab_frekvence-struktur

Zaměřme se nyní na výše uvedenou tabulku. Jak bylo předpokládáno, každá z nalezených struktur má nemalé zastoupení v rámci datasetu a je zde znatelné, že některé kombinace se obecně vyskytují častěji než jiné. Kromě číselných sekvencí jsou zde například struktury <CVCCVC> nebo <CVCVCV> v různých obměnách, které lze sledovat jak v souborech z anglofonního prostředí, tak i z českého (ačkoli ten má jisté nedostatky, jak již bylo popsáno v kapitole *Datasety*). Co je však poněkud anomální, je kombinace <CCCCCC>, tedy šest bezprostředně za sebou jdoucích konsonantů, na druhé pozici v datasetu Alpaka – vzpomeneme-li však na analýzu hapaxů provedenou v předcházející části, jedná se zde nejspíše právě o ona unikátní hesla, která byla s nejvyšší pravděpodobností generovaná platformou (popř. jiným způsobem společným pro velké množství uživatelů). Doklad toho, že kombinatorika konsonantů a vokálů je pro reprezentativní datasety společná, lze nalézt v **příloze 3** této práce, která se věnuje analýze *n-gramů* těchto struktur. Zjistili jsme zde, že pro datasety Kosatka, Krajta a RockYou jsou užité struktury velmi podobné: Kendallova korelace zde pro jednotlivé dvojice vychází mezi 88 a 95 %, přičemž Alpaka v kombinaci s žádných z datasetů nepřesáhne 37 %.

***

Souhrnný náhled na soubory je poskytnut souhrnu tabulek *_tab_struktury-kvant*, které vyobrazuje základní kvantitativní indexy, a to jak pro typy, tak pro tokeny struktur. Tokeny jsou zde původní typy hesel převedené na sylabickou strukturu, typy jsou poté jednotlivé struktury v sylabickém datasetu. Ve všech případech je tedy vidět, že veškeré typy hesel jsou realizovány pomocí nevelikého počtu struktur. To dokládá také hodnota TTR, která např. v případě Kosatky je 0,08, jinak také 8 %. Z původního počtu tokenů struktur je pouze 8 % typů. Díky tomu, že se po převedení tokenů na typy se výrazně zmenšila sledovaná množina struktur, poté u typů můžeme sledovat, že oproti tokenům se navýšil průměr, medián a 95% interval délek hesel. Vidíme zde také malé zastoupení hapaxů, tedy unikátních struktur: v případě celých datasetů se jedná o jednotky procent. Celkově tak tyto výsledky ukazují, že ačkoli hesel je v datasetech mnoho, užívají podobnou kombinatoriku konsonantů, vokálů, čísel a spec. znaků.

Alpaka		
Celé struktury	**Tokeny**	**Typy**
Celkový počet	381 908	6 161
TTR	0,02	
Průměrná délka	6,46	8,96
Medián délek	6	9
95% interval délek	5–10	5–14
Množství hapaxů	0,78 %	48,27 %
Znaky v rámci souboru		
Celkový počet znaků	2 466 479	55 222
Konsonanty	52,70 %	45,99 %
Vokály	22,77 %	31,3 %
Čísla	24,43 %	20,6 %
Speciální znaky	0,09 %	2,12 %

Krajta		
Celé struktury	**Tokeny**	**Typy**
Celkový počet	1 388 099	75 702
TTR	0,06	
Průměrná délka	8,38	11,65
Medián délek	8	11
95% interval délek	6–13	7–19
Množství hapaxů	3,4 %	62,39 %
Znaky v rámci souboru		
Celkový počet znaků	11 633 176	881 862
Konsonanty	43,39 %	46,70 %
Vokály	26,87 %	28,32 %
Čísla	29,17 %	21,09 %
Speciální znaky	0,57 %	3,89 %

Kosatka		
Celé struktury	**Tokeny**	**Typy**
Celkový počet	413 959	31 089
TTR	0,08	
Průměrná délka	8,49	10,50
Medián délek	8	10
95% interval délek	5–12	6–16
Množství hapaxů	4,30 %	57,21 %
Znaky v rámci souboru		
Celkový počet znaků	3 512 765	326 532
Konsonanty	47,13 %	48,06 %
Vokály	27 %	27,95 %
Čísla	25,7 %	22,74 %
Speciální znaky	0,16 %	1,25 %

RockYou		
Celé struktury	**Tokeny**	**Typy**
Celkový počet	14 341 227	436 160
TTR	0,03	
Průměrná délka	8,73	13,85
Medián délek	8	14
95% interval délek	6–15	8–24
Množství hapaxů	1,97 %	64,84 %
Znaky v rámci souboru		
Celkový počet znaků	125 240 372	6 041 265
Konsonanty	39,42 %	45,44 %
Vokály	25,97 %	30,4 %
Čísla	33,40 %	16,95 %
Speciální znaky	1,20 %	7,21 %

_tab_struktury-kvant

V neposlední řadě se ve spojení s tématem sylabických struktur zaměřme na počet speciálních znaků (S) a čísel (D). Spec. znaků je ve všech souborech jednoznačně nejméně: procentuální zastoupení je v rámci jednotek, ne-li méně. To ukazuje na fakt, že jen malé množství uživatelů volí užití speciálních symbolů. Mnohem častěji se zde objevují čísla – jejich procentuální zastoupení se ve všech případech blíží procentuálnímu zastoupení vokálů. To nás vede k otázce, čím je tento stav způsoben. Odpovědí se nabízí více, lze uvažovat např. o těchto: (I) je to efekt způsobený pravidly platformy, (II) jedná se o obecnou uživatelskou preferenci, (III) existuje poměrně velké množství hesel, které jsou tvořeny pouze čísly, popř. v nich čísla převažují. V případě (I) počítáme s určitými restrikcemi, které platforma klade na uživatele při tvorbě hesla (např. *heslo musí obsahovat alespoň jedno číslo*), a díky kterým jsou čísla natolik prominentní v celých souborech. V takovémto případě bychom však museli počítat s touto restrikcí v případě všech souborů, jelikož jejich hodnoty jsou podobné. Tato možnost se jeví jako nepravděpodobná vzhledem k příkladům struktur uvedených v tabulce *_tab_frekvence-struktur* – ne všechna hesla obsahují číslo. Případ (II) počítá s možnou uživatelskou preferencí čísel oproti speciálním znakům. Tato možnost se již jeví jakožto pravděpodobnější, a to hlavně když uvážíme hledisko zapamatovatelnosti či vázanosti k socio-kulturnímu kontextu: s čísly se v běžném životě setkáváme denně, mohou to být kupříkladu data narození (popř. jiných událostí, viz Veras et al., 2012), mohou být kulturně vázaná (např. 007 odkazující k agentu Bondovi, 68 k Jaromíru Jágrovi nebo 42 jako odpověď na základní otázku života, vesmíru a vůbec). Případ (III) v podstatě vychází z případu předchozího, počítáme zde však s hesly, která jsou tvořena čísly primárně či zcela. Jak vidíme v tabulce *_tab_frekvence-struktur*, tento typ hesla se drží na prvních pozicích ve všech souborech, v RockYou dokonce obsazuje prvních pět příček. Při náhledu do souborů zjišťujeme, že se jedná o sekvence nejrůznějších *složitostí*: od opakování jednoho čísla či střídání dvou a více čísel (např. <80808080>), přes postupky (např. <123456>) až po zdánlivě náhodné sekvence (např. <42137777>), v některých případech doplněné o další znaky (např. <01081964an>). Obecně lze tak problematiku čísel uzavřít přikloněním se k bodům (II) a (III). Čísla jsou dále rozebírána v **příloze 4** této práce, kde se do detailu zaobíráme numerickými sufixy (např. <heslo1900> nebo <abc123>), které jsou připojovány za hesla: uvádíme jejich nejčastější délky a podoby – pro více viz příslušnou přílohu.

***

Obecně tato část přináší zajímavé poznatky, co se týká unikátnosti hesel. Mohli jsme zde pozorovat, že když nahlédneme jednotlivé datasety z hlediska kombinatoriky konsonantů a vokálů, vyvstává neveliké množství struktur, jimiž jsou jednotlivá hesla v datasetu realizována. Tato vlastnost ukazuje na to, že ačkoli si uživatel vytvoří unikátní heslo, je pravděpodobné, že jej z hlediska takovéto struktury bude sdílet s jiným uživatelem (či jinými uživateli) – což potenciálně nese implikace pro bezpečnost takového hesla. Další část této práce se zaměřuje právě na tuto slabinu a pokusí se jí zneužít k prolamování hašů hesel dvou ze zde analyzovaných datasetů.

Sylabické struktury a prolamování hašů hesel

Jak jsme mohli sledovat v předcházející části, hesla všech platforem jsou si do určité míry podobná co do rozložení konsonantů, vokálů, čísel a spec. znaků, které jsou užity v jejich formaci. Tato kombinatorika částečně přirozeně vychází z lexikální povahy značného množství hesel, částečně však lze hovořit o jisté uživatelské preferenci umisťovat konkrétní typy znaků na konkrétní místa. Tato emergentní vlastnost je pak potenciální slabinou (uživatelsky volených) hesel, jelikož, vzhledem k nemalému zastoupení každé z těchto struktur, jí lze využít k prolamování hašů hesel.

Podívejme se nyní do důsledku na to, kolik *hesel* a kolik *uživatelů* tyto struktury v průměru sdílí. Situaci si ilustrujeme na příkladu datasetu Kosatka. Počet uživatelů je zde označen N, počet typů hesel V; počet různých struktur je dále označen V_{CV}. Mezi nimi platí vztah

$$N \geq V \geq V_{CV}$$

tedy že počet uživatelů je vyšší nebo roven počtu typů hesel, který je vyšší nebo roven počtu typů různých struktur hesel. Zaměřme se nejprve na to, kolik hesel v průměru sdílí strukturu, formálně:

$$\frac{V}{V_{CV}} = \frac{413\,972}{31\,121} \approx 13{,}30$$

tedy v průměru ~13 různých hesel sdílí sylabickou strukturu. Toto je však případ seznamu hesel, kdy se každé z nich vyskytuje pouze jednou – vztáhneme-li počet struktur k počtu uživatelů, formálně:

$$\frac{N}{V_{CV}} = \frac{590\,298}{31\,121} \approx 18{,}97$$

zjišťujeme, že ~19 uživatelů si vytvořilo heslo se stejnou sylabickou strukturou (ať už bylo identické, nebo „pouze“ sdílelo rozložení konsonantů, vokálů, čísel, spec. znaků). Toto měření bylo provedeno pro všechny datasety (opět s výjimkou RockYou); přehled výsledků je poskytnut v tabulce *_tab_prumer-poctu-struktur*.

Sdílení struktur					
	N	V	V_{CV}	V/V_{CV}	N/V_{CV}
Alpaka	766 421	381 908	6 161	61,99	124,40
Kosatka	590 298	413 972	31 121	13,30	18,97
Krajta	2 234 829	1 388 099	75 756	18,32	29,50

_tab_prumer-poctu-struktur

Výše uvedená tabulka poskytuje náhled na to, kolik hesel a kolik uživatelů celkem sdílí sylabické struktury v rámci jednotlivých datasetů. Zaměříme-li se zde na

poslední sloupec, tedy počet toho, kolik uživatelů celkem strukturu sdílí, zjišťujeme, že se v případech reprezentativních datasetů (Kosatka, Krajta) jedná až o ~19 a ~30 lidí. Vybočení datsetu Alpaka již v tomto bodě práce není překvapující, jelikož anomální hodnoty jsme mohli pozorovat v naprosté většině zde provedených analýz.

Hlavním zjištěním je zde to, že počet uživatelů se stejnou strukturou není zanedbatelný, což nás vede k myšlence zda tyto vlastnosti jsou nebo mohou být útočníky využity k efektivnějšímu prolamování souborů hašů. V části *Typy útoků na databáze* byly popsány možnosti, které má útočník, zaměřuje-li se na soubor hašů hesel. Vedle slovníkového útoku a útoku hrubou silou existuje v programu *hashcat* také možnost využít tzv. masky, tedy předem definované struktury hesel, dle kterých se generují kandidáti na prolomení daného haše. Tyto masky lze tvořit buď předdefinovanými skupinami znaků: malá/velká písmena, číslice, speciální znaky, nebo lze definovat skupiny vlastní. Maskami tak lze definovat heslo se strukturou např. *libovolná sekvence malých písmen o délce šest* – takováto struktura však nereflektuje možnost využití potenciálu lexikální povahy hesel, nýbrž je orientována čistě na jeho grafickou podobu. Přístup k prolamování hašů pomocí masek využívající konsonanty a vokály prozatím nebyl odborně zpracován a nebyla provedena komparace výkonnosti těchto dvou typů masek – alespoň dle našeho nejlepšího vědomí. Jediné nalezené reference odkazují k online fórům[15], kde se uživatelé touto možností zabývají povrchně či pouze na teoretické rovině. V následující části se tedy pokusíme využít právě této emergentní vlastnosti hesel k prolamování hašů s tím, že její efektivita bude srovnána s výkonností masek generovaných dle pozic malých a velkých písmen.

— Příprava masek a souborů pro komparaci
Pro experimentální komparaci výkonnosti masek využívající malá a velká písmena (dále LUDS masky) a masek využívajících konsonanty a vokály (dále CVDS masky) byly využity pouze dva datasety: Krajta a Kosatka. Takovéto rozhodnutí bylo učiněno z toho důvodu, že Alpaka není dostatečně reprezentativní pro uživatelskou praxi – jak jsme ostatně mohli pozorovat v rámci této práce; RockYou nebyl prolamován z podobného důvodu (jeho povaha seznamu je taktéž nereprezentativní), ale byl využit pro tvorbu masek, jelikož díky tomu, že původně unikl v plaintextu (na rozdíl od „prolámané“ Alpaky), obsahuje veškeré možnosti podob uživatelských hesel.

Prvním provedeným krokem byla aplikace hašovací funkce MD5 na datasety Kosatka a Krajta. Krom toho, že se jedná o funkci dříve typicky užívanou k ochraně hesel v databázích, byla MD5 zvolena primárně z důvodu časové úspornosti – na GPU, kterou máme pro potřeby této práce k dispozici, je možné spočítat až 30 milionů MD5 hašů za vteřinu. Hlavním cílem bylo simulovat podmínky tak, aby výsledky tohoto experimentu byly v dohledné době; pro získání výsledků pro výpočetně náročnější hašovací funkce lze výsledné hodnoty násobit příslušným koeficientem, a tedy získat náhled na to, jak by si masky vedly při jejich prolamování.

[15] Např.: https://hashcat.net/forum/thread-8280.html [cit. 14-08-2021]

Dalším krokem byla tvorba masek. Obecnými principy, které provázely proces jejich tvorby (a obecně designování podmínek prolamování), byla férovost šancí obou typů masek a zároveň co možná nejvěrnější simulace praxe prolamování souboru hašů. Postup generování masek si přiblížíme na popisu kroků při přípravě na prolamování datasetu Kosatka.

1) **Průnik**
   Nejprve byl vytvořen průnik hesel, která se objevují v datasetech Krajta a RockYou, a to ze dvou důvodů: (I) jedná se tak o hesla, která nejsou unikátní pouze pro jeden dataset, (II) užitím pouze Krajty a RockYou je zajištěno to, že se masky negenerují z datasetu, který se jimi bude prolamovat – v opačném případě by došlo ke zvýhodnění masek, jelikož by bylo jisté, že generované struktury jsou v datasetu obsaženy.
2) **Substituce**
   Znaky v heslech průniku byly substituovány příslušnými skupinami zástupných znaků: *l/u/d/s* (malá a velká písmena, číslice, speciální znaky) a *c/v/d/s* (konsonanty, vokály, číslice, speciální znaky).
3) **Seřazení dle frekvence**
   Ze dvou výsledných seznamů tokenů masek byly dále vytvořeny frekvenční seznamy, v nichž nejpopulárnější struktury obsadily první příčky.
4) **Zkrácení seznamů na stejnou délku**
   Oba výsledné seznamy byly zkráceny na stejnou délku 5 000 nejpopulárnějších masek. Tento krok byl proveden primárně z důvodu striktního zachování stejných podmínek pro tyto typologicky odlišné masky. Jsme si každopádně vědomi toho, že ani v jednom z případů by těmito zkrácenými seznamy nebylo prolomeno takové množství hesel, jako je plný potenciál generovaných masek, a to jelikož zde nyní nejsou obsaženy všechny *možné* struktury. Implikace, které s sebou nese nezkrácení seznamů na stejnou délku pak sahají nad rámec této práce, avšak jsou potenciálním tématem pro budoucí výzkum.

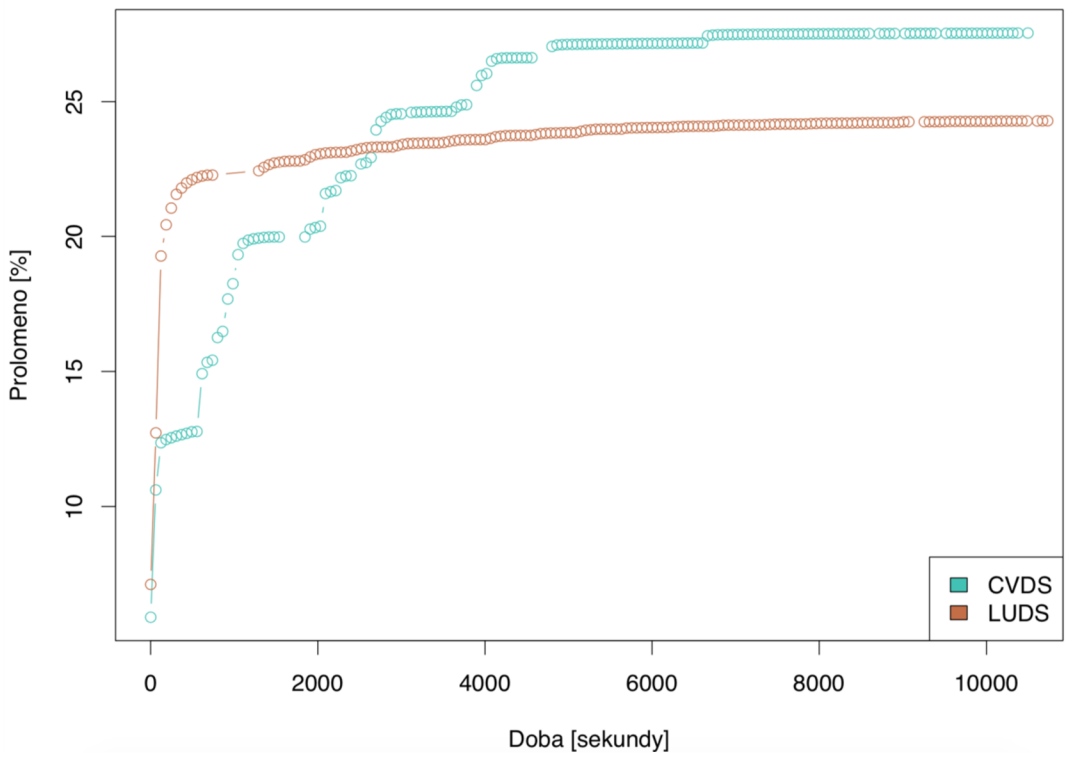
Výsledkem výše popsaných kroků jsou dva separátní soubory masek, kdy jedna z nich využívá čistě grafických vlastností znaků (LUDS), druhá je oproti tomu orientovaná na lexikální stránku hesel (CVDS). Stejný postup byl aplikován při tvorbě masek pro prolamování Krajty, avšak pro jejich tvorbu byl využit průnik hesel Kosatky a RockYou z důvodů popsaných v kroku 1.

Soubory MD5 hašů Kosatky a Krajty byly následně podrobeny experimentální komparaci efektivity prolamování pomocí těchto dvou typů masek. Otázka, která zde vyvstává, je ta, zda se masky CVDS efektivitou vyrovnají zavedeným maskám typu LUDS, jelikož rozložení konsonantů a vokálů v rámci hesla stojí na myšlence, že hesla jsou z velké části lexikální povahy. Na druhou stranu je však možné, že to, co stojí za typickým rozložením konsonantů a vokálů v heslech, není lexikální povaha, ale obecně jistá uživatelská preference umisťovat jisté skupiny znaků na konkrétní místa. V případě LUDS lze předpokládat, že velká písmena budou primárně na začátku hesel, a to analogicky k jazyku, kdy první písmena jmen, názvů atp. jsou právě velká. V případě CVDS pak lze uvažovat o jisté preferenci užívat např. více konsonanty než vokály čistě z toho důvodu, že je to větší skupina a poskytuje tak větší variabilitu. Co se pak týká rozmístění konsonantů a vokálů

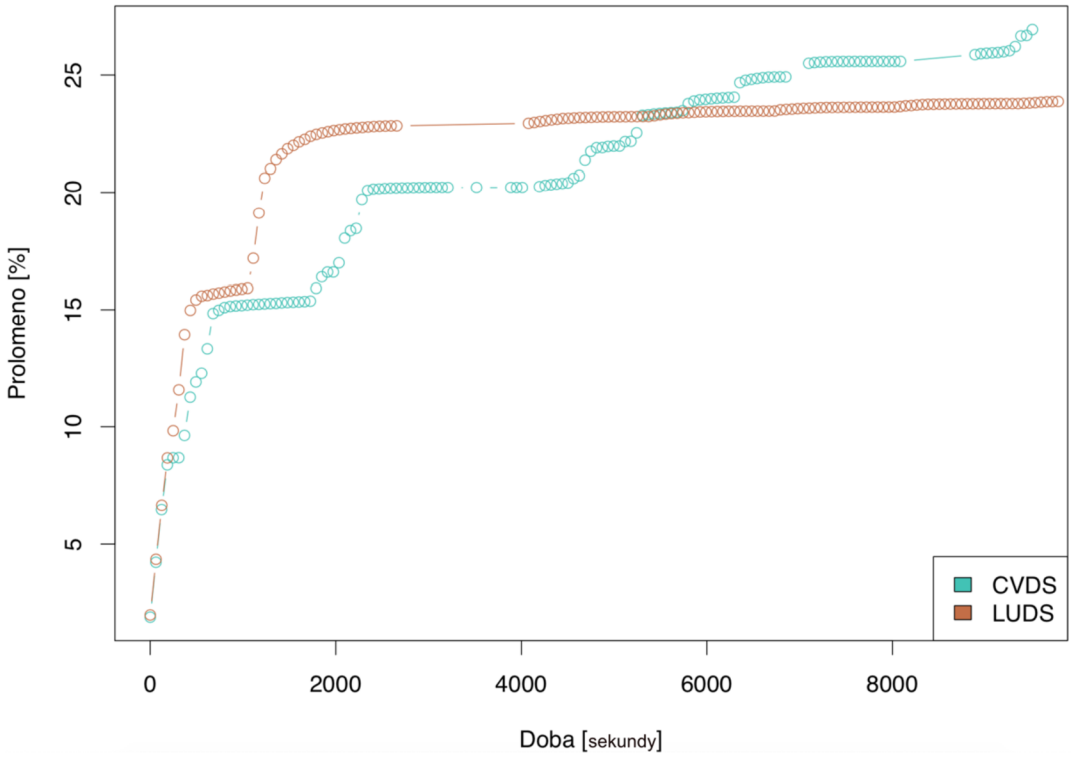
v heslech, lze případně uvažovat taktéž o latentních rysech jazyka. Přesuňme se tedy nyní k popisu průběhu komparace.

— Průběh a výsledky experimentální komparace
Každý ze souborů hašů (Kosatka a Krajta) byl prolamován dvakrát: jednou pomocí masek LUDS, podruhé maskami CVDS. Samotné měření probíhalo pomocí skriptu v jazyce *R*, který pro zadaný seznam masek spustil program *hashcat* a zároveň zahájil měření času a kontrolu souboru, do kterého se prolomená hesla průběžně ukládala. Kontrola souboru probíhala každou minutu: pokud došlo v souboru ke změně, tj. přibyla prolomená hesla, byl zaznamenán čas změny (v podobě časové značky) a aktuální počet hesel v souboru. Problémem, který zde nastal, bylo to, že časování bylo vždy mírně zpožděno, jelikož každá kontrola souboru zabrala přibližně vteřinu. Měření času zde tak bylo s každou přibývající kontrolou mírně posunuto – tato drobná vada však nenese implikace pro správnost výsledků. Prolamování samotné bylo omezeno na tři hodiny, po kterých byly výsledky uloženy a pro každou dvojici masek vizualizovány v grafu.

Podívejme se nyní na výsledky komparace. Grafy *_graf_kompar-Kosatka* a *_graf_kompar-Krajta* vyobrazují to, jak si jednotlivé typy masek vedly při prolamování daných souborů hašů.



_graf_kompar-Kosatka



_graf_kompar-Krajta

Osa *x* znázorňuje dobu od spuštění programu vyjádřenou v sekundách; osa *y* znázorňuje to, kolik procent datasetu bylo prolomeno. V grafu pak můžeme sledovat průběh prolamování daných datasetů, který je pro oba případy analogický. LUDS masky mají v obou případech rychlejší nástup – tedy že krátce po spuštění prolomí více hašů než masky CVDS. Tato situace se mění mezi 20 a 25 % datasetu, kdy LUDS masky začínají stagnovat. CVDS masky mají oproti tomu pomalejší nástup, avšak v průběhu prolamování dělají skoky, kterými postupně předčí LUDS masky a výsledně tak za tři hodiny překonají hranici 25 % datasetu. Jak můžeme sledovat v případě Kosatky, nad 25 % datasetu již stagnují i CVDS masky. Obecně je však vidět, že za takto krátkou dobu mají vyšší efektivitu prolamování než LUDS masky.

Co se týká těchto výsledků, je třeba mít na paměti následující aspekty. Prvním z nich je doba, po kterou prolamování probíhalo – pouhé tři hodiny. Je zde otázkou, jak by se situace vyvíjela, nechali-li bychom prolamování pokračovat po delší dobu (kupříkladu 24 hodin a více). Dalším aspektem, který vstupuje do prolamování, je design procesu tvorby masek. Zde pracujeme s datasety, které jsou zaměřené na anglofonní prostředí, a tedy je možné, že jejich CVDS struktura je díky tomu analogická. V případě prolamování hašů pomocí takovýchto masek je pak možné, že je nutné brát v potaz prostředí, ze kterého daný soubor hašů pochází. Oproti tomu jsou LUDS masky univerzálnější – netěží totiž z lexikálních vlastností hesel, nýbrž z vlastností grafických. Vedle *tvorby* jednotlivých masek je pak potřeba brát v úvahu jejich *řazení*: program *hashcat* prochází masky postupně. Zde jsme zvolili řazení masek od nejpopulárnější po nejméně populární – existuje však úskalí toho, že pokud *hashcat* narazí na příliš dlouhou masku, bude tvořit všechny možné kombinace, a tedy prolamování začne nevyhnutelně stagnovat – jak se tomu stalo u

LUDS masek v obou případech. Je tak nutno zvážit nalezení rovnováhy mezi délkou masky a popularitou hesel, která jsou touto maskou zastoupena. Zda by tyto aspekty měly větší dopad na efektivity CVDS masek v porovnání s LUDS maskami lze v této chvíli pouze spekulovat.

Vedle separátního užití LUDS a CVDS masek lze také uvažovat o jejich kombinaci – tedy masky, které by braly v potaz malá a velká písmena a zároveň konsonanty a vokály: množiny by se tak rozdělily na (I) malé vokály, (II) velké vokály, (III) malé konsonanty, (IV) velké konsonanty. Pokud k těmto skupinám připočteme množinu číslic a speciálních znaků, jednalo by se o masky tvořené šesti specifickými skupinami – což může být na jednu stranu výhoda, ale zároveň se tento fakt může stát kamenem úrazu. Takovéto masky by na jednu stranu mohly být rychlejší, jelikož množiny, které jsou jednotlivými znaky zastoupeny, jsou menší, než v případě CVDS i LUDS masek – výpočet tak bude probíhat kratší dobu. Na druhou stranu mohou takovéto masky být přespříliš specifické a nesou riziko toho, že neobsáhnou tolik potenciálních hesel jako o něco obecnější LUDS a CVDS masky – tato otázka však také zůstává otevřena budoucímu zájmu.

***

Jak jsme v této části mohli pozorovat, prolamování hašů pomocí rozložení konsonantů a vokálů v rámci hesla přináší slibné výsledky. Oproti zavedené metodě, kdy se k prolamování využívají pozice velkých a malých písmen, se nám zde podařilo v obou měřených případech pomocí těchto masek prolomit vetší množství hašů za danou dobu a dostat se až nad čtvrtinu prolomeného datasetu. Přestože tato metoda není bez úskalí (jakým může být např. počet hesel, které tyto masky celkově obsáhnou v porovnání s LUDS maskami), jedná se o potenciálně zajímavé téma pro budoucí výzkum. Jejich efektivitu je pak nutno změřit na větším vzorku datasetů, a to ideálně z různých geografických prostředí s cílem zjistit, zda jazyk uživatelů je aspektem, který je nutné brát v potaz. Do srovnání rychlosti a efektivity by dále bylo vhodné zařadit více typů útoků – zařadit zde vedle útoku maskami také útok hrubou silou a slovníkový útok, aby bylo možné posoudit efektivitu takovýchto masek v kontextu nejen v kontextu užitých množin, ale v kontextu rozsáhlejší praxe prolamování hašů. Celkově se však jedná o podnět, který by mohl nést hlubší implikace nejen pro útoky na soubory hašů, ale pro uživatele volící konkrétní hesla a metodiku tvorby a posuzování síly hesel obecně.

Shrnutí a závěr

Na stránkách této práce jsme se zabývali vybranými aspekty uživatelských hesel, přičemž jsme se pohybovali na pomezí kvantitativní lingvistiky a kybernetické bezpečnosti. Práce byla orientována primárně prakticky – teorie byla omezena na nezbytný kontext týkající se hesel z hlediska kybernetické bezpečnosti. Byly tak popsány aspekty tvorby hesla a obecná doporučení či restrikce, kterými platformy regulují podobu a sílu uživatelsky tvořených hesel; dále jsme se věnovali ochraně hesel v databázích, kdy hlavním zájmem pro nás byly hašovací funkce (jejich vlastnosti, výhody, ale také slabiny), jelikož v této práci byly využity i v jedné z provedených analýz.

Po shrnutí výše uvedeného kontextu jsme se již přesunuli k měřením a analýzám, jimž byly věnované zbylé části. Ty lze obecně rozdělit dle toho, co bylo objektem jejich zájmu: nejprve to bylo *to, co je časté* a sledovali jsme obecné trendy a nejčetnější jevy, poté jsme přesunuli pozornost k *tomu, co je unikátní* a zaměřili se tak na specifika unikátních hesel. V první řadě byl proveden popis datasetů, které byly v rámci celé práce užívány k popisu nejrůznějších vlastností hesel. Tato část se na jedné straně věnovala rovině metainformací, kdy byly jednotlivé datasety zasazeny do kontextu z hlediska jejich geografického a demografického původu; na straně druhé se tato část věnovala základním kvantitativním indexům hesel, jako je jejich délka, TTR, množství hapaxů atp. Byla také provedena sémantická analýza nejfrekventovanějších hesel, kdy jsme se pomocí sémantických embeddingů a MDS pokusili o identifikaci tematických skupin, přičemž bylo zjištěno, že v tomto ohledu lze do jisté míry najít shodu napříč všemi analyzovanými datasety.

Poté jsme naši pozornost obrátili k unikátním heslům – tedy heslům, která mají v rámci každého datasetu pouze jednoho uživatele. Vycházeli jsme z předpokladu, že unikátnost hesel je dána tím, že k častým typům jsou přidány specifické sekvence čísel či speciálních symbolů. Tento předpoklad byl však vyvrácen výpočtem Levenshteinovy vzdálenosti, která ukazuje na to, že unikátní hesla jsou nejčastěji sama o sobě unikátní sekvence znaků a nejedná se tak o modifikace několika populárních typů. K otázce unikátnosti hesel jsme poté přistoupili z jiného úhlu pohledu, a to analýzou rozložení lingvisticky motivovaných skupin znaků v heslech: vedle čísel a speciálních znaků jsme sledovali typické struktury, které v heslech vyvstávají z dichotomie konsonanty/vokály. Zjistili jsme tak, že v rámci celých datasetů existuje v tomto ohledu nemalá shoda – tento poznatek dále motivoval závěrečnou část této práce. Ta se věnovala možnosti prolamování hašů hesel na základě této vlastnosti pomocí tzv. maskového útoku. Byla tak provedena experimentální komparace zavedeného typu masek (využívající velká/malá písmena, LUDS masky) se zde navrženými maskami (CVDS masky). Ačkoli prezentované výsledky nejsou bez úskalí, CVDS masky prokázaly vyšší účinnost při prolamování MD5 hašů, kdy za uplynulé tři hodiny dokázaly prolomit více než 25 % obou zkoumaných datasetů, čímž předčily LUDS masky. Prolamování hašů maskami

typu CVDS je (dle našeho nejlepšího vědomí) stále neprobádanou oblastí, a tedy zde prezentované výsledky potenciálně staví základy pro nový výzkum.

Obecným cílem této práce, jak bylo zmíněno v úvodu, nebylo prozkoumat jednu konkrétní problematiku do hloubky, ale spíše zmapovat pole působení kybernetické bezpečnosti, ve kterém by poznatky komputační a kvantitativní lingvistiky mohly poskytnout nový úhel pohledu a potenciálně nové poznatky. Vzhledem k charakteru zde prezentovaných výsledků a otevření otázek pro potenciální budoucí výzkum (primárně co se CVDS struktur týká) tak lze označit cíl práce za splněný.

Literatura a zdroje

Baker, P. (2006). *Glossary of corpus linguistics*. Edinburgh University Press.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146.

Burr, W. E. et al. (2013). *Electronic Authentication Guideline. NIST Special Publication 800-63-2*. Gaithersburg: NIST.

Čermák, F. (2017). *Korpus a korpusová lingvistika*. Univerzita Karlova: Karolinum Press.

de Carné de Carnavalet, X., & Mannan, M. (2014). From very weak to very strong: Analyzing password-strength meters. In *Network and Distributed System Security Symposium (NDSS 2014)*. Internet Society.

Devillers, M. M. (2010). Analyzing password strength. *Radboud University Nijmegen, Tech. Rep*, *2*.

Hranický, R., Zobal, L., Večeřa, V., Múčka, M. (2018). *Distribuce výpočtů pro nástroj hashcat*. Vysoké učení technické v Brně.

Kim, Peter (2015). *Hacking: praktický průvodce penetračním testováním*. Zoner Press.

Klíma, V. (2005). *Hašovací funkce, principy, příklady a kolize*. Dostupné z http://crypto-world.info/klima/2005/cryptofest_2005.htm [cit. 17-08-2021].

Koch, R. (1999). *The 80/20 Principle: The Secret of Achieving More with Less*. London: Nicholas Brealey Publishing.

Krámský, J. (1959). Teorie sdělné promluvy. In *Slovo a Slovesnost* 20: 55–66.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* 10(8): 707–710.

mask_attack [hashcat wiki]. *hashcat: advanced password recovery*. https://hashcat.net/wiki/doku.php?id=mask_attack [cit. 17-08-2021]

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

NordPass. (2021). *Top 200 Most Common Passwords of 2020*. https://nordpass.com/most-common-passwords-list/ [cit. 10-08-2021]

NÚKIB (2019). *Bezpečný pohyb v kybersvětě*. https://www.nukib.cz/cs/infoservis/doporuceni/1508-doporuceni-pro-bezpecny-pohyb-v-kybersvete/ [cit. 02-08-2021].

Pavlíček, L., Sedláček, J. (2020). *Bezpečnost informačních systémů: materiály ke cvičením 1*. Vysoká škola ekonomická v Praze, Nakladatelství Oeconomica.

R Core Team (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell system technical journal*, *30*(1): 50–64.

Siegler, M. G. (2009, December 15). One Of The 32 Million With A RockYou Account? You May Want To Change All Your Passwords. Like Now. *TechCrunch*. https://techcrunch.com/2009/12/14/rockyou-hacked/?guccounter=2 [cit. 10-07-2021].

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika 17*(4): 401–419.

Veras, R., Thorpe, J., & Collins, C. (říjen 2012). Visualizing semantics in passwords: The role of dates. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security* (pp. 88–95).

When Is "Y" a Vowel or Consonant?. The Merriam-Webster.Com Dictionary. https://www.merriam-webster.com/words-at-play/why-y-is-sometimes-a-vowel-usage [cit. 10-07-2021]

Příloha 1

Základní kvantitativní indexy datasetů: souhrnné tabulky

Alpaka		
Celá hesla	**Tokeny**	**Typy**
Celkový počet	766 421	381 908
TTR	0,50	
Průměrná délka	6,54	6,46
Medián délek	6	6
95% interval délek	5–10	5–10
Množství hapaxů	43,48 %	87,26 %
Pareto 20/x	60,14 %	–
Znaky v rámci souboru		
Celkový počet znaků	5 011 617	2 466 479
Konsonanty [%]	53,03	52,70
Vokály [%]	30,17	22,77
Čísla [%]	16,71	24,43
Speciální znaky [%]	0,08	0,09

Kosatka		
Celá hesla	**Tokeny**	**Typy**
Celkový počet	590 298	413 959
TTR	0,70	
Průměrná délka	8,27	8,49
Medián délek	8	8
95% interval délek	5–12	5–12
Množství hapaxů	60,48 %	86,25 %
Pareto 20/x	43,90 %	–
Znaky v rámci souboru		
Celkový počet znaků	4 881 293	3 512 765
Konsonanty [%]	49,22	47,13
Vokály [%]	28,40	27,00
Čísla [%]	25,70	22,25
Speciální znaky [%]	0,13	0,16

Krajta		
Celá hesla	**Tokeny**	**Typy**
Celkový počet	2 234 829	1 388 099
TTR	0,62	
Průměrná délka	7,96	8,38
Medián délek	8	8
95% interval délek	5–13	6–13
Množství hapaxů	53,20 %	85,65 %
Pareto 20/x	50,31 %	–
Znaky v rámci souboru		
Celkový počet znaků	17 783 822	11 633 176
Konsonanty [%]	45,37	43,39
Vokály [%]	28,63	26,87
Čísla [%]	25,57	29,17
Speciální znaky [%]	0,42	0,57

RockYou	
Celá hesla	
Celkový počet	14 341 227
Průměrná délka	8,73
Medián délek	8
95% interval délek	6–15
Znaky v rámci souboru	
Celkový počet	125 240 372
Konsonanty [%]	39,42
Vokály [%]	25,97
Čísla [%]	33,40
Speciální znaky [%]	1,20

Příloha 2

Kumulativní součet datasetu dle frekvencí hesel

Pro tři ze zde analyzovaných datasetů (Alpaka, Kosatka, Krajta) jsme se pokusili odpovědět na otázku *kolik procent datasetu zabírají málo frekventovaná hesla*. Sledovali jsme tedy hesla s frekvencí 1–10, přičemž pro každou z délek bylo zjištěno její procentuální zastoupení. Výsledky lze pozorovat v souhrnu tabulek *_tab_kumsoucet*.

Alpaka				
Frekvence	**Tokeny**		**Typy**	
	[%]	Kumulativní součet [%]	[%]	Kumulativní součet [%]
1	43,48	43,48	87,26	87,26
2	5,89	49,37	5,91	93,17
3	2,81	52,18	1,88	95,05
4	2,08	54,26	1,04	96,09
5	1,64	55,90	0,66	96,75
6	1,42	57,32	0,47	97,22
7	1,33	58,65	0,38	97,60
8	1,20	59,85	0,30	97,90
9	1,03	60,88	0,23	98,13
10	0,93	61,81	0,19	98,32

Kosatka				
Frekvence	**Tokeny**		**Typy**	
	[%]	Kumulativní součet [%]	[%]	Kumulativní součet [%]
1	60,48	60,48	86,25	86,25
2	10,96	71,44	7,81	94,06
3	5,27	76,71	2,50	96,56
4	3,45	80,16	1,23	97,79
5	2,43	82,59	0,69	98,48
6	1,69	84,28	0,40	98,88
7	1,24	85,52	0,25	99,13
8	0,80	86,32	0,14	99,27
9	0,66	86,98	0,10	99,37
10	0,54	87,52	0,08	99,45

Krajta				
Frekvence	Tokeny		Typy	
	[%]	Kumulativní součet [%]	[%]	Kumulativní součet [%]
1	53,20	53,20	85,65	85,65
2	10,44	63,64	8,41	94,06
3	4,42	68,06	2,37	96,43
4	2,65	70,71	1,07	97,50
5	1,80	72,51	0,58	98,08
6	0,23	73,88	0,37	98,45
7	1,10	74,98	0,25	98,70
8	0,90	75,88	0,18	98,88
9	0,78	76,66	0,14	99,02
10	0,68	77,34	0,11	99,13

_tab_kum-soucet

Tabulky *_tab_kum-soucet* vyobrazují, kolik procent jednotlivých datasetů zabírají nízko frekventovaná hesla – do této analýzy vstoupila hesla s frekvencemi 1–10. Zaměřme se nejprve na tokeny, tedy uživatele, kteří daná hesla vytvořili. Pro každou z frekvencí bylo zjištěno, kolik procent celého datasetu tvoří hesla s danou frekvencí – např. v datasetu Krajta zabírají unikátní hesla 53,2 % datasetu, dále hesla s dvojím výskytem 10,44 % datasetu, hesla objevující se třikrát tvoří 4,42 % datasetu; třetí sloupec tabulky vyobrazuje kumulativní součet zjištěných %, přičemž na posledním řádku poskytuje informaci, že celkem hesla s frekvencemi 1–10 zabírají ~61–87 % celých datasetů. Nahlédneme-li na tento počet z druhé strany, zjišťujeme, že ~13–39 % jednotlivých datasetů je tvořeno hesly, která sdílí více než 10 uživatelů.

V případě typů byl postup analogický, akorát zde bereme v potaz pouze *podoby* (nikoli jednotlivé uživatele) hesel a ptáme se, kolik procent těchto podob je zastoupeno daným počtem (frekvencí). Zůstaňme u příkladu datasetu Krajta. Unikátní hesla v případě typů zabírají 85,65 % datasetu typů, typy s dvojím výskytem 8,41 % datasetu typů atd. U frekvence 10 v kumulativním součtu vidíme, že se v případě všech datasetů pohybujeme mezi ~98–99 % datasetu. To značí, že hesla s frekvencemi 1–10 obsahují téměř všechny *podoby* hesel.

Příloha 3

Kombinatorika konsonantů, vokálů, čísel a symbolů

V následující části se vrátíme k myšlence sledování nejčastějších kombinací. Jak bylo možno vidět již v tabulce *_tab_frekvence-struktur*, některé struktury se (v obměnách) vyskytují častěji než jiné – jako tomu bylo například u <CVCCVC>. Abychom se mohli lépe zaměřit na tuto problematiku, provedli jsme pro takto substituované datasety analýzu tzv. n-gramů. Na výstupu takovéto analýzy očekáváme náhled na situaci v datasetech co do jejich kombinatoriky konsonantů a vokálů, mimo jiné s cílem zjistit, zda jsou si v tomto ohledu datasety podobné.

N-gramy jsou pojmem korpusové lingvistiky; jedná se o spojení *n* po sobě jdoucích jednotek, kdy *n* může být libovolné číslo – vznikají tak bigramy, trigramy, tetragramy atd. Sledováním četností výskytu konkrétních spojení je pak možné ve zkoumaném textu (popř. obecněji v jazyce) odhalit časté kolokace, ustálená slovní spojení či obecně přítomné struktury. Zde užijeme analýzu n-gramů pro získání náhledu na nejčastějších kombinací konsonantů, vokálů, čísel a speciálních symbolů v jednotlivých datasetech.

Následující postup byl proveden pro každý dataset typů struktur zvlášť. Každý dataset byl převeden na n-gramy znaků s tím, že mezera byla brána v potaz (tedy znamenala předěl, přes který jsme n-gramy netvořili), a to kvůli tomu, že hesla jako taková nejsou souvislý text. Při tvorbě n-gramů z textu bychom běžně mezeru v potaz nebrali a postupovali následovně:

analyzovaný text: bezpečné heslo
3-gramy: bez, ezp, zpe, peč, ečn, čné, néh, éhe, hes, esl, slo

Avšak vzhledem k tomu, že datasety mají povahu seznamů, mezery budou brány v potaz a kombinace na jejich hranicích se nebudou tvořit. Ilustrujme si toto zpracování na příkladu:

analyzované struktury: <CVCCVC>, <CVCVD>
3-gramy: <CVC>, <VCC>, <CCV>, <CVC>, *(vynecháme <VCC> a <CCV> na předělu)*, <CVC>, <VCV>, <CVD>.

Tímto postupem zabráníme tomu, aby jisté kombinace (v příkladu výše se jedná o <VCC> a <CCV>) neprávem nabíraly hodnotu frekvence a tím potenciálně zkreslily výsledky.

Z každého datasetu takto byly postupně vytvořeny seznamy 2, 3, 4, 5 až 6-gramů. Maximální číslo $n = 6$ bylo zvoleno na základě empirie: jedná se o nižší hranici průměrné délky hesel pro některé z datasetů. Vytvořené seznamy byly spojeny dohromady a následně z nich byla vytvořena frekvenční tabulka, která vyobrazuje, které n-kombinace se v datasetu vyskytují nejčastěji. Pro ilustraci viz tabulku *_tab_n-gramy-struktur*.

Nejčastější n-gramy struktur (n = 2–6)					
Rank	**n-gram**	**Frekvence**	**Rank**	**n-gram**	**Frekvence**
Alpaka					
1	CC	566 247	11	CD	137 787
2	CV	412 301	12	VCV	134 459
3	VC	367 891	13	CCCC	119 535
4	DD	332 550	14	DC	117 830
5	CVC	275 056	15	DDDDD	114 657
6	CCC	253 321	16	CVCV	109 582
7	DDD	228 230	17	CVCC	95 800
8	DDDD	166 845	18	VCVC	85 004
9	CCV	147 176	19	CCVC	83 135
10	VCC	141 170	20	DDDDDD	75 272
Kosatka					
1	CV	771 301	11	DDDD	215 463
2	VC	683 556	12	CCVC	205 294
3	DD	580 402	13	CVCV	186 656
4	CC	573 146	14	CD	180 212
5	CVC	556 952	15	VCCV	165 918
6	DDD	354 949	16	CCC	145 367
7	VCC	305 367	17	VCVC	144 976
8	CCV	293 139	18	CVCCV	136 660
9	CVCC	252 484	19	CDD	118 450
10	VCV	231 576	20	VV	116 209
Krajta					
1	CV	2 534 176	11	CVCC	751 773
2	DD	2 329 594	12	CVCV	697 383
3	VC	2 221 699	13	DDDDD	641 009
4	CVC	1 803 893	14	CCVC	615 675
5	CC	1 638 360	15	VCVC	534 449
6	DDD	1 530 725	16	VCCV	534 251
7	DDDD	1 020 517	17	CD	517 709
8	VCC	916 055	18	CVCCV	439 031
9	CCV	897 131	19	DDDDDD	424 538
10	VCV	864 968	20	CVCVC	422 527
RockYou					
1	DD	31 433 156	11	VCC	8 869 479
2	CV	26 330 765	12	CCV	8 702 260
3	DDD	23 079 641	13	CVCV	7 983 091
4	VC	22 899 810	14	CVCC	7 242 826
5	CVC	18 491 343	15	VCVC	6 143 933
6	DDDD	17 231 097	16	CCVC	5 895 831
7	CC	15 170 563	17	VCCV	5 375 292
8	DDDDD	12 460 468	18	CVCVC	4 848 241
9	VCV	9 955 857	19	CVCCV	4 391 466
10	DDDDDD	9 133 082	20	VV	4 113 472

_tab_n-gramy-struktur

V tabulce *_tab_n-gramy-struktur* vidíme vyobrazených dvacet nejčastějších n-gramů pro každý z jednotlivých datasetů. Již při pohledu na první příčky si lze všimnout, že minimálně v případě Kosatky, Krajty a RockYou jsou pořadí struktur podobné. Tuto podobnost dále ověříme pomocí Kendallovy korelace. Ta bere v potaz rank jednotlivých položek v seznamu a vyjádří jejich podobnost na škále 0–1, kde 0 značí nulovou korelaci a 1 absolutní korelaci – tedy že položky seznamu jsou seřazeny identicky.

Kendallova korelace				
	Kosatka	Alpaka	RockYou	Krajta
Kosatka	1			
Alpaka	0,37	1		
RockYou	0,88	0,31	1	
Krajta	0,90	0,33	0,95	1

_tab_kendall-ngram

Tabulka *_tab_kendall-ngram* je matice vyobrazující Kendallovu korelaci pro jednotlivé páry datasetů. Jak můžeme vidět, v případě kombinací anglofonních (reprezentativních) datasetů vyšla korelace ve vysokých hodnotách: pohybujeme se zde mezi 0,88 a 0,95. Oproti tomu veškeré kombinace s Alpakou nepřesahují hranici 0,37. Vyvstává zde otázka, proč tomu tak je, ačkoli odpověď na ni je v tomto bodě práce již poměrně snadná. S nejvyšší pravděpodobností je to další manifestace nereprezentativnosti tohoto datasetu z důvodu, že je prolámaný z hašů. Popřípadě se může jednat o rozdíl v jazyce, ve kterém jsou lexikální hesla tvořena, avšak pro ověření takovéto úvahy by bylo nutné provést další srovnání s více českými datasety, které by byly reprezentativní povahy. Zde se tedy přikláníme k první možnosti, a to primárně proto, že anomálie Alpaky můžeme sledovat v rámci celé této práce, a tedy ani zde pro nás není překvapující.

Příloha 4

Numerické sufixy

Vidíme zde také, že struktury tvořené symboly <C> a <V> jsou často na konci doplněny o numerický sufix: srov. např. <CVCCVCDD> zastupující např. <secret77> nebo <CVCCDDDD> zastupující např. <word1111>. Pokud by byly prominentní v rámci celých datasetů, mohli bychom hovořit o jisté uživatelské preferenci umisťovat čísla na konec hesla. V následující části se na tuto problematiku zaměříme. Nejprve, pro zjištění, kolik hesel takové struktury v datasetech je, byly pomocí regulárního výrazu extrahovány veškeré kombinace <C> a/nebo <V> končící libovolným počtem <D>.

Výraz	**Vyhledá**
^[CV]+(D){1,}$	vše začínající libovolnou kombinací a počtem <C> a/nebo <V>, končící libovolným počtem čísel <D>

Tento regulární výraz se neomezuje pouze na hesla kombinující vokály a konsonanty, ale vyhledá i sekvence konsonantů a/nebo vokálů doplněné číslem na konci – jinými slovy tedy veškeré struktury, které končí sekvencí čísel, i když jsou to pouze vokály nebo konsonanty. Ačkoli se ne ve všech případech může jednat o heslo lexikální povahy, přistoupili jsme k tomuto kroku z následujících důvodů: (I) ne všechna slova nutně obsahují vokál, jádrem slabiky může být i konsonant (srov. např. v češtině sonory *r* nebo *m* ve slovech krk nebo sedm); (II) ne všechna slova nutně obsahují konsonant (srov. např. francouzské *eau* [*voda*]); (III) sekvence pouze konsonantů/vokálů může mít povahu zkratky či iniciál. Srovnáme-li dále množství takto extrahovaných struktur, zjišťujeme, že tímto způsobem je tvořena nemalá část typů hesel.

Zastoupení struktur s numerickým sufixem				
Dataset	Kosatka	Krajta	Alpaka	RockYou
Zastoupení	43 %	46 %	16 %	37 %

_tab_zastoupení-struktur

V tabulce *_tab_zastoupení-struktur* vidíme, že zastoupení hesel s numerickým sufixem se v reprezentativních datasetech typů pohybuje mezi 37 a 46 procenty. Jinými slovy to znamená, že pokud nahlédneme na to, jaká různá hesla uživatelé na platformách tvoří, až takovéto množství je sekvence písmen doplněná o číslo či čísla. Nízká hodnota v případě Alpaky je nejspíše způsobená jeho nereprezentativní povahou (pro podrobný popis viz dedikovaný oddíl v části *Datasety*). Dalším možným faktorem je jeho orientace na české prostředí, jelikož to je druhá vlastnost, kterou se od ostatních datasetů liší. To však nelze prokázat; resp. nelze tvrdit, že Češi mají jiné zvyklosti tvorby hesel než jiné národnosti, zakládáme-li toto tvrzení na datasetu, který byl prolomen z hašů, a tedy obsahuje jednodušší hesla.

Další otázka, která vyvstává u tématu numerických sufixů, je to, jaký počet čísel za heslem je nejběžnější či *kolik čísel v sufixu uživatel nejspíše užije, tvoří-li nové heslo*. Pro tyto sufixy očekáváme, že budou spíše krátké a primárně se bude jednat

o délky 1–4, a to s odkazem na příkladovou frekvenční tabulku v části *Datasety*, kde se objevují sufixy o délkách 1 a 3; délka 4 se očekává z důvodu předpokládaného výskytu letopočtů či dat (např. <password1990>). Pro tuto analýzu byla data zpracována následujícím způsobem: ze struktur vyhledaných regulárním výrazem byly odstraněny veškeré výskyty <CV>, což ponechalo pouze sekvence <D> různých délek; z těchto byly vytvořeny frekvenční tabulky. Níže můžeme sledovat nejčastější numerické sufixy v datasetech.

Nejčastější délky numerických sufixů					
#	**Sufix**	**Freq**	**#**	**Sufix**	**Freq**
Kosatka			**Krajta**		
1	DD	57 138	1	DD	225 505
2	DDDD	46 061	2	DDDD	132 610
3	D	32 490	3	D	127 720
4	DDD	25 869	4	DDD	102 359
5	DDDDDD	7 527	5	DDDDDD	24 849
6	DDDDD	6 646	6	DDDDD	18 078
7	DDDDDDD	1 727	7	DDDDDDD	6 139
8	DDDDDDDD	1 035	8	DDDDDDDD	4 514
9	DDDDDDDDD	166	9	DDDDDDDDD	1 316
10	DDDDDDDDDD	79	10	DDDDDDDDDD	695
Alpaka			**RockYou**		
1	D	22 455	1	DD	1 832 857
2	DD	21 865	2	DDDD	1 130 009
3	DDDD	10 802	3	D	988 218
4	DDD	5 900	4	DDD	766 154
5	DDDDDD	529	5	DDDDDD	268 562
6	DDDDD	422	6	DDDDD	174 495
7	DDDDDDDD	80	7	DDDDDDD	74 865
8	DDDDDDD	78	8	DDDDDDDD	59 905
9	DDDDDDDDD	26	9	DDDDDDDDD	21 940
10	DDDDDDDDDD	18	10	DDDDDDDDDD	15 844

_tab_délka-sufixů

Jak lze vidět v tabulce *_tab_délka-sufixů*, očekávání nejčastějších délek je potvrzeno: délky 1–4 se objevují na prvních čtyřech pozicích, ve třech datasetech dokonce ve stejném pořadí. Jediná anomálie je, nepřekvapivě, u datasetu Alpaka, přičemž ji lze vysvětlit opět tím, že se jedná o prolomený dataset: jedno číslo na konci hesla je nejjednodušší možnost kombinace, a tedy je možné, že právě proto se zde vyskytuje nejčastěji. I v případě Alpaky však délky 1–4 zabírají první čtyři pozice.

Pro získání souhrnného náhledu na to, jak takové sufixy nejčastěji vypadají, uvádíme zde níže tabulku *_tab_sufixy-freq*, která vyobrazuje deset nejčastějších sufixů obecně, tedy bez ohledu na jejich délku (v prvním sloupci), dále pak pro nejčastější délky: 1–4 (druhý až pátý sloupec). Vidíme zde, že i v rámci konkrétních užitých sufixů lze najít nemalou míru shody napříč všemi datasety. Zároveň zde můžeme sledovat trend týkající se hesel obecně: tedy vázanost k socio-kulturnímu kontextu – ta je nejvíce viditelná v sufixech o délce 3, kde vidíme sekvence jako 007 nebo 666, a 4, jejíž příčky jsou, vedle nejčastější postupky, obsazeny výhradně letopočty.

Nejčastější numerické sufixy hesel										
#	Vše	Freq	Délka 1	Freq	Délka 2	Freq	Délka 3	Freq	Délka 4	Freq
Alpaka										
1	**1**	6 333	**1**	6 333	**11**	1 085	**123**	1 790	**1234**	366
2	**2**	2 143	**2**	2 143	**12**	987	**007**	467	**2010**	244
3	**7**	1 916	**7**	1 916	**01**	978	**111**	235	**2000**	176
4	**5**	1 896	**5**	1 896	**22**	622	**666**	173	**2009**	157
5	**3**	1 851	**3**	1 851	**77**	603	**001**	131	**2008**	152
6	**123**	1 790	**4**	1 744	**13**	599	**321**	87	**2011**	138
7	**4**	1 744	**8**	1 698	**10**	505	**777**	81	**2007**	124
8	**8**	1 698	**6**	1 683	**23**	439	**456**	57	**2006**	103
9	**6**	1 683	**9**	1 638	**21**	402	**333**	55	**2005**	100
10	**9**	1 638	**0**	1 551	**99**	362	**159**	54	**1980**	98
Kosatka										
1	**1**	18 034	**1**	18 034	**01**	3 732	**123**	4 271	**1234**	1 283
2	**123**	4 271	**2**	3 972	**12**	3 209	**111**	787	**2000**	722
3	**2**	3 972	**3**	2 065	**11**	3 018	**001**	574	**2007**	616
4	**01**	3 732	**7**	1 755	**99**	1 959	**777**	491	**2005**	516
5	**12**	3 209	**5**	1 468	**13**	1 534	**007**	489	**2006**	507
6	**11**	3 018	**4**	1 269	**69**	1 517	**101**	446	**2004**	484
7	**3**	2 065	**9**	1 157	**22**	1 509	**100**	435	**2003**	435
8	**99**	1 959	**8**	1 146	**10**	1 287	**666**	355	**2002**	430
9	**7**	1 755	**6**	983	**00**	1 238	**333**	307	**2001**	333
10	**13**	1 534	**0**	641	**23**	993	**999**	301	**2008**	262
Krajta										
1	**1**	62 810	**1**	62 810	**12**	14 046	**123**	22 033	**1234**	4 529
2	**123**	22 033	**2**	16 149	**11**	8 593	**101**	3 081	**2008**	2 395
3	**2**	16 149	**3**	10 101	**13**	7 633	**007**	2 110	**2009**	2 207
4	**12**	14 046	**7**	8 596	**01**	6 972	**777**	1 764	**2007**	1 842
5	**3**	10 101	**5**	6 997	**22**	6 714	**666**	1 684	**2010**	1 721
6	**7**	8 596	**4**	6 288	**23**	6 153	**111**	1 678	**2006**	1 449
7	**11**	8 593	**8**	4 789	**21**	5 826	**321**	1 551	**2000**	1 343
8	**13**	7 633	**9**	4 691	**10**	5 433	**420**	1 465	**2005**	1 220
9	**5**	6 997	**6**	4 642	**09**	5 408	**143**	1 403	**2004**	997
10	**01**	6 972	**0**	2 656	**08**	5 357	**100**	1 355	**1994**	938
RockYou										
1	**1**	439 492	**1**	439 492	**12**	85 812	**123**	103 040	**1234**	22 309
2	**2**	125 786	**2**	125 786	**13**	65 453	**101**	21 919	**2007**	19 511
3	**123**	103 040	**3**	83 332	**11**	54 810	**666**	14 090	**2006**	17 473
4	**12**	85 812	**7**	71 932	**22**	50 014	**143**	13 996	**2008**	16 489
5	**3**	83 332	**5**	61 382	**23**	49 860	**007**	12 964	**2005**	11 551
6	**7**	71 932	**4**	54 494	**07**	49 781	**100**	9 926	**1994**	9 873
7	**13**	65 453	**8**	45 954	**01**	48 441	**777**	8 931	**1992**	9 598
8	**5**	61 382	**6**	42 570	**21**	47 838	**420**	8 613	**1993**	9 478
9	**11**	54 810	**9**	40 573	**14**	45 216	**111**	8 419	**1995**	9 018
10	**4**	54 494	**0**	22 675	**10**	44 505	**321**	7 786	**1991**	8 920

_tab_sufixy-freq

Obecně tyto poznatky tedy ukazují na to, že nemalé množství uživatelů následuje podobný vzor tvorby hesla, a tedy to, že heslo je sekvencí alfabetických znaků doplněných o numerický sufix. Zároveň vidíme, že i mezi těmito sufixy lze najít jistou konvenci napříč uživateli, a to co do jejich délky, tak i do nejčastěji volených konkrétních sufixů.