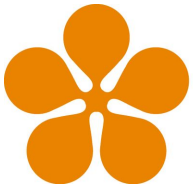


Pedagogická fakulta
Faculty of Education

Jihočeská univerzita v Českých Budějovicích
University of South Bohemia in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

Využití databází hacknutých uživatelských hesel jako nástroje pro zlepšení zabezpečení uživatelských účtů

The use of databases of hacked user passwords as a tool for improving protection of user accounts

Bakalářská práce

Vypracoval: Petr Samec

Vedoucí práce: Mgr. Václav Šimandl Ph.D.

České Budějovice 2023

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Petr SAMEC**
Osobní číslo:	**P20498**
Studijní program:	**B7507 Specializace v pedagogice**
Studijní obor:	**Informační technologie a e-learning**
Téma práce:	**Využití databází hacknutých uživatelských hesel jako nástroje pro zlepšení zabezpečení uživatelských účtů**
Zadávající katedra:	**Katedra informatiky**

Zásady pro vypracování

Cílem bakalářské práce je zmapovat využitelnost databází hacknutých uživatelských hesel ke zlepšení zabezpečení uživatelských účtů na webu. Student upraví moduly pro registraci a přihlášení běžného uživatele do služby ve webové aplikaci Bobříka informatiky, přičemž zajistí jejich propojení s databází českých a slovenských hacknutých přihlašovacích údajů (tj. uživatelských hesel a e-mailů popř. uživatelských jmen). Informace o hacknutých uživatelských údajích bude student čerpat převážně z databází Collection #1 – #5. Na základě realizovaných úprav bude student schopen zjistit, zda dané přihlašovací údaje (uživatelské jméno + heslo i emailová adresa + heslo) nebyly v rámci internetu hacknuty. V případě, že hacknuty byly, bude uživateli doporučena změna uživatelského hesla. Student funkčnost vytvořeného řešení ověří v praxi v období kolem školního kola soutěže a následně zanalyzuje, u kolika uživatelů byly zjištěny hacknuté údaje a kolik z nich přistoupilo k doporučované změně hesla. Analýza se zaměří na to, za jak dlouho si uživatel změnil hacknuté heslo za nové a zda je toto nové heslo bezpečnější než to, které doposud používal. V teoretické části práce student popíše způsoby, jakými lze zjistit bez vědomí uživatele jeho přihlašovací údaje. Zabývat se bude především technikami, jako jsou útok hrubou silou, útok prostřednictvím tzv. duhových tabulek (v případě použití hashovacích algoritmů MD5 a SHA1), sociální inženýrství a phishing.

Rozsah pracovní zprávy:	**40**
Rozsah grafických prací:	**–**
Forma zpracování bakalářské práce:	**tištěná**

Seznam doporučené literatury:

1. HADNAGY, Christopher. Social Engineering: The Science of Human Hacking. 2nd edition. Crosspoint Boulevard: Wiley, 2018. ISBN 978-1119433385.
2. HADNAGY, Christopher a Michele FINCHER. Phishing Dark Waters: The Offensive and Defensive Sides of Malicious Emails. Crosspoint Boulevard: Wiley, 2015. ISBN 978-1118958476.
3. POSTON, Howard. 10 most popular password cracking tools. Infosec Resources [online]. Infosec Institute, 2020. Dostupné z: https://resources.infosecinstitute.com/topic/10-popular-password-cracking-tools/
4. WELLING, Luke a Laura THOMSON. Mistrovství PHP a MySQL. Přeložil Ondřej BAŠE. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.
5. Defuse Security Research and Development [online]. Dostupné z: https://defuse.ca/
6. Hack crack [online]. Netmux, 2017. ISBN 978-1975924584. Dostupné z: https://digtvbg.com/files/books-for-hacking/
7. Online Safety – CyberInsureOne. Cyber Security Insurance – CyberInsureOne [online]. Cyber Insure One, 2019. Dostupné z: https://cyberinsureone.com/online-safety/

Vedoucí bakalářské práce: **Mgr. Václav Šimandl, Ph.D.**
Katedra informatiky

Datum zadání bakalářské práce: **11. dubna 2022**
Termín odevzdání bakalářské práce: **30. dubna 2023**

L.S.

doc. RNDr. Helena Koldová, Ph.D.
děkanka

doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 11. dubna 2022

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

V Českých Budějovicích dne 26. června 2023.

Petr Samec

Abstrakt

V teoretické části práce jsou popsány techniky prolomení a odcizení hesel. V praktické části se práce zabývá zabezpečením webové stránky Bobříka informatiky pomocí tabulek odcizených údajů a analýzou uživatelských hesel. Tabulky odcizených údajů jsou importované do databáze Bobříka informatiky, pomocí kterých se kontroluje bezpečnost uživatelských údajů při přihlášení do webové stránky. Ze získaných uživatelských záznamů vznikla analýza hacknutých uživatelů, u kterých jsme sledovali, do jaké míry si uživatel zvolil složitější heslo a za jak dlouho, dále analýza závislosti síly hesla na pohlaví, analýza závislosti síly hesla učitele na škole, ve které vyučuje a analýza síly hesel uživatelů webové stránky Bobříka informatiky. Hacknutých uživatelů se zaznamenalo pouze 9, z čehož 4 přistoupili na změnu hesla. Současně jsme zjistili, že síla hesla nezávisí na pohlaví učitele ani na typu školy, ve které vyučuje. Zároveň se ukázalo, že polovina uživatelů Bobříka informatiky používá slabá hesla, jedna čtvrtina používá obstojná hesla a druhá čtvrtina používá relativně silná hesla.

Klíčová slova

prolomení hesel, zapezpečení webové stránky, tabulky odcizených údajů

Abstract

The theoretical part of the thesis describes techniques for password cracking and stealing passwords. The practical part of the thesis deals with the safety of the Bobřík infromatiky website using tables of stolen dates and analysis of user passwords. Database tables of stolen data are imported into Bobřík informatiky database, which are used to check the security of user data when logging into the website. From the obtained user records, we created an analysis of the hacked users, where we investigated the complexity of the user's password change and how long it took. Furthermore, analysis of the dependence of password strength on gender, analysis of the dependence of password strength of a teacher at the school which he teaches in and analysis of password strength of users of the Bobřík informatiky website. There were only 9 hacked users. Only 4 agreed to change their passwords. At the same time, we found that the strength of the password does not depend on the gender or the school of the teacher. It turns out that half of Bobřík informatiky users use weak passwords, one quarter use fair passwords and the other quarter use relatively strong passwords.

Keywords

password cracking, web security, credential stuffing

Poděkování

Chtěl bych poděkovat Mgr. Václavu Šimandlovi Ph.D. za odbornné vedení, cenné rady a trpělivost při zpracovávání bakalářské práce.

Obsah

1 Úvod

Většina z nás si nedokáže představit, jaký by byl náš život bez počítače a internetu. Každý uživatel určitě používá pár svých vytvořených hesel pro různé internetové účty, kde je potřeba mít své vlastní přihlašovací údaje a heslo. Proto bychom měli dbát na jejich bezpečnost, neboť nemusejí být tak bezpečná, jak si myslíme. Často se stává, že používáme pořád stejná hesla, která jsme si vytvořili již dříve, abychom si je dobře pamatovali. Bohužel tímto naším nedůsledným chováním se nám snadno může stát, že používané přihlašovací údaje, které zadáváme do všech našich účtů, nám hackne hacker. Hacknuté údaje se pak můžou jednoduše vyskytnout v hacknutých kolekcích přihlašovacích údajů na dark webu. Odcizené přihlašovací údaje poté může někdo použít na zneužití našich účtů, například údajů do bankovnictví, sociálních účtů a mnoho dalších. Proto bychom měli dbát na bezpečnost hesel. V této práci si popíšeme různé techniky, které se používají k prolomení hesel. Na internetu jsou dostupné miliardy přihlašovacích údajů ke stažení, které se dají zneužít proti nám, ale i využít k lepšímu zabezpečení. Proto některé z těchto přihlašovacích údajů použijeme v této práci k lepšímu zabezpečení webu Bobříka informatiky a ke kontrole bezpečnosti hesel jeho uživatelů.

1.1 Cíle práce

Cílem práce je využití hacknutých hesel z databází ke zlepšení bezpečnosti internetových účtů. Na internetu existuje velké množství hacknutých hesel, které se dají snadno zneužít k prolomení uživatelských účtů. Tato hesla ale můžeme využít i k jejich lepšímu zabezpečení tím, že uživatelům oznámíme, že jejich heslo není bezpečné a bylo by potřeba si jej změnit na nové unikátní heslo. Praktická část je rozdělena na dvě části, v první se zabýváme implementací modulu login a změny hesla na webové stránce Bobříka informatiky, který bude upozorňovat uživatele na nebezpečnost jejich hesel a doporučí změnu hesla. V druhé části práce analyzujeme, kolik uživatelů mělo hacknuté heslo a kolik jich přistoupilo na změnu hesla. Budeme zkoumat, kolik uživatelů bylo na doporučení implementovaného modulu ochotno si změnit heslo, za jak dlouho dobu si uživatel změnil heslo a složitost nově zvoleného hesla oproti starému. Jelikož se nepodařilo nasbírat dostatek dat pro konstruktivní analýzu hacknutých uživatelů, vznikl nový cíl práce, analýza zastoupení hesel různé síly a závislost síly hesla učitele na pohlaví a typu školy, ve které vyučuje. V teoretické části upozorníme na potenciální nebezpečí, jak snadno lze bez vědomí uživatele zjistit jeho přihlašovací údaje.

1.2 Metoda práce

Modul pro kontrolu bezpečnosti a měření síly hesla budeme programovat v Joomle. Hacknutá hesla budou uložena v databázi phpMyAdmin. Při vyplnění přihlašovacího formuláře se každému uživateli vyhodnotí síla hesla, která se vloží do databázové tabulky pro analýzu hesel. Při vyplnění přihlašovacího formuláře modul bude také porovnávat emailovovu adresu a heslo s databází uniklých údajů. Modul vyhodnotí, zda heslo bylo již dříve hacknuto. Pokud se najde shoda s databází, uživatel dostane upozornění, že jeho heslo není bezpečné a bude mu doporučena změna. Podle datumu nalezení shody hacknutého hesla a změny hesla se analyzuje, za jak dlouhou dobu

uživatel přistoupil na změnu svého hesla. Zjistíme, kolik hacknutých uživatelů se nalezlo v systému, kolik jich přistoupilo na změnu hesla a do jaké míry složitější heslo uživatel při změně použil.

Při nalezení shodného emailu a hesla s databází, se do tabulky hacknutých uživatelů vloží záznam, který bude obsahovat sílu hesla a čas, kdy byla nalezena shoda. Při změně hesla hacknutého uživatele se do stejného řádku tabulky hacknutých uživatelů vloží čas změny hesla a síla hesla. Podle těchto záznamů budeme zkoumat, jak silné si uživatel zvolil heslo a po jaké době uživatel přistoupil na jeho změnu od obdržení výstražné zprávy o nebezpečnosti uživatelských údajů. K provedení další analýzy hesel použijeme nasbírané obodovaná hesla učitelů. Z těchto záznamů provedeme analýzu, zastoupení hesel různé síly a závislost síly hesla učitele na pohlaví a typu školy, ve které vyučuje. Analýzu závislosti síly hesla učitele na pohlaví a na typu školy, ve které vyučuje provedeme pomocí statistických testů. Crackování hesel budeme čerpat z knihy Password Cracking Techniques [-PUNISHER-].

2 Techniky prolamování hesel

2.1 Heslo

Heslo můžeme považovat za tajný přístupový kód. V nejjednoduší formě je heslo pouze tajné slovo nebo fráze, které se používá pro ověření a určení uživatele, za kterého se vydává. V dnešní době, když si představíme heslo, automaticky předpokládáme, že toto slovo se váže k přihlášení do webové stránky nebo je spojeno s počítači a dalšími elektronickými zařízeními [1].

2.2 Stručná historie hesel

Na rozdíl od počítačů hesla byla využívána už v prvních civilizacích. Heslo neboli přístupový kód bylo používané slovo, které povolilo osobě se identifikovat a umožnit vstup přes stráž do zabezpečeného místa. Když se lidé chtěli dostat přes stráž, museli vědět heslo, které jim povolilo vstup. Pokud heslo neznali, nebyli puštěni dovnitř. Tento způsob neověřuje identitu osoby, ale udává, že daná osoba má přístup k místu pro určené lidi. Problém této metody spočívá v lidech, kteří heslo znají a na jejich schopnosti udržet ho v tajnosti [1].

2.3 Útoky založené na uhodnutí hesla

2.3.1 Random guess (Náhodný odhad)

Náhodný odhad hesla může hacker použít pokud zná ID uživatele nebo email osoby, kterou chce hacknout. Tato uživatelská jména se obvykle nemění. Ve většině případů se uživatel přihlašuje přes svoji email adresu, se kterou veřejně komunikuje, proto je lehké tuto adresu zjistit. V této chvíli už hacker zná polovinu údajů, aby se dostal do cizího účtu. Útok funguje na princpu náhodného zadávání hesla do formuláře pro heslo, tedy manuální zadávaní hesel, které mají danou spojitost s cílovou obětí. Náhodný pokus o odhadnutí hesla má malou úspěšnost, pokud neznáme žádné osobní informace o uživateli, kterého chceme hacknout, pokud není heslo natolik primitivní a běžné, aby ho hacker mohl lehce odhadnout. Heslo majitele účtu může být například oblíbený film. Tyto informace můžeme zjistit na sociálních sítí nebo přímou komunikací. Nejbežnější hesla uživatelů, na které hádání hesel funguje, většinou obsahují slovo jako „heslo“ nebo podobné výrazy s obměnou písmena za číslo „hesl0“. Lidé často zaměňují písmena za čísla jako A za 4, E za 3 nebo a za @. Těchto ekvivalentů existuje mnoho. Dále může být heslo uživatele uvedeno jako uživatelské jméno, které má přidaná čísla nebo speciální znaky. V dalším případě se často uvádí datum narození uživatele nebo jeho příbuzných. Nejčastěji lidé používají datum narození svých dětí. Uvádí se také památná místa a události, jména domácích zvířat, oblíbené barvy, jídlo a další důležité věci pro uživatele [1].

2.3.2 Dictionary attack (Slovníkový útok)

Slovníkový útok je násilná technika uhodnutí hesla. Útočník při útoku využívá takzvané wordlisty neboli seznamy slov, kde jsou uloženy různé kombinace hesel. Při pokusu o nabourání do cizího účtu se začne vkládat každé heslo ze seznamu, dokud se nenajde shoda. Útok je automatizován speciálními nástroji, jako jsou John the Ripper, L0phtCrack, Aircrack-ng. Je velmi důležité vybrat

správný seznam pro konkrétní cíl. Když se budeme snažit nabourat do českého účtu, je velmi pravděpodobné, že cílový uživatel účtu bude mít nastavené české heslo, proto nemůžeme použít seznam, ve kterém se nacházejí převážně anglická hesla [2].

Při slovníkovém útoku velmi záleží, jestli se přihlašujeme do účtu online nebo offline. Při online útoku do účtů musí útočník počítat s omezením počtu pokusů pro zadání hesla. Po několika neúspěšných pokusech zadání hesla může systém prevence průniku vyhodnotit pokusy o přihlášení se do účtu jako útok, nebo může dojít k vyčerpání pokusů pro přihlášení. Pokud tato situace nastane, účet se zamkne a útočník nemůže dál pokračovat [3].

Když cílová webová stránka nemá toto zabezpečení nasazené, neznamená to, že útočník může zkoušet náhodná hesla, jak se mu zlíbí. Hádání hesla ze stejné IP adresy vypadá velmi podezřele, proto se doporučuje při každých například 10 pokusech změnit IP adresu přes proxy server nebo počkat 30 minut, než se znova pokusí o uhodnutí hesla. Tyto funkce jsou už vytvořené v programech pro hackování hesel [3].

Aplikování slovníkového útoku offline je možné, pokud máme hash hesel. Útok je prováděn na vlastím počítači, proto nás nelimitují žádné ochranné zámky jako při online útoku. Offline slovníkový útok hashuje každé slovo z připraveného slovního seznamu. Zahashované heslo se porovná s uniklým hashem. Když se hashe shodují, úspěšně jsme nalezli zahashované heslo. Při offline útoku je zapotřebí mít kvalitní procesor a grafickou kartu. Čím výkonnější hardware počítač má, tím více pokusů dokáže aplikovat za sekundu [3].

Jak moc jsou efektivní slovníkové útoky? To, že lidé by měli používát silná hesla neznamená, že tak dělají. Mnoho lidí se spokojí s používáním stejného hesla na svých účtech. Data Breach Investigations Report (DBIR) ukazují, že 80 % ukradených a opakovaně používaných hesel se použije při slovníkovém útoku [2]. Balbix State of Password Use Report 2020 uvádí: *„Více než 99 % uživatelů používá stejná hesla, ať už v pracovních nebo osobních účtech a v průměru je každé heslo použito na 2,7 účtech [4].“*

2.3.3 Brute force attack (Útok hrubou silou)

Útok hrubou silou je metoda, která funguje jako pokus omyl. Útok se obvykle provádí automatizovanými programy, skripty nebo boty. Při útoku hrubou silou dochází k vytváření různých kombinací uživatelských jmen a hesel, které jsou vkládány do cílového webového přihlašovacího formuláře nebo do přihlašovacího formuláře aplikace, dokud nenajde správnou kombinaci. Pro představu můžeme útok hrubou silou ilustrovat u Rubikovy kostky. Máme zamíchanou kostku, abychom ji úspěšně složili, nebudeme postupovat podle pravidel, dle kterých se kostka skládá, ale využijeme princip útok hrubou silou. Začneme točit každou hranou kostky do různých směrů, dokud kostka nebude mít každou stranu vyplněnou jednou barvou [3].

Útok hrubou silou můžou zprostředkovávat programy jako jsou Brutus, Aircrack-ng, John the Ripper, Rainbow Crack, L0phtCrack, Ophcrack, Hascat, Dave Grohl, Ncrack, THC Hydra [5].

Brutus je program pro prolamování hesel, který umožní provést slovníkový útok a útok hrubou silou. Dokáže prolomit protokoly HTTP, POP3, FTP, SMB, Telnet. V programu se nastavují parametry, podle kterých se bude přistupovat k prolomení hesla [6].

U útoku hrubou silou lze nastavit:

- délka hesla,
- použití pouze číslic,
- použití pouze malých písmen,
- použití pouze velkých písmen,
- použití pouze malých a velkých písmen,
- použití pouze malých a velkých písmen s číslicemi,
- použití všech znaků včetně speciálních znaků,
- vlastní nastavení (pokud tušíme znaky, které by heslo mohlo obsahovat, je umožněno přidat seznam znaků, které bude program využívat).

Každá z uvedených možností má jiný počet možných kombinací. U hesla dlouhého 6 znaků bude možných kombinací:

- v případě použití pouze číslic: 1 000 000 možných kombinací,
- v případě použití pouze malých písmen bez diakritiky: 321 272 406 možných kombinací,
- v případě použití pouze velkých písmen bez diakritiky: 321 272 406 možných kombinací,
- v případě použití pouze malých a velkých písmen bez diakritiky: 20 158 268 676 možných kombinací,
- v případě použití pouze malých a velkých písmen s číslicemi bez diakritiky: 57 731 386 986 možných kombinací,
- v případě použití všech znaků včetně speciálních znaků bez diakritiky: 697 287 735 690 možných kombinací [3].

U 6 znakového hesla, ve kterém se použíjí všechny znaky, máme 697 287 735 690 možných kombinací. Pokud bychom u tohoto hesla využili 7 znaků, počet kombinací výrazně vzroste na 65 545 047 154 955 [3].

Výhodou útoku hrubou silou je jeho jednoduchost provedení. Všechna hesla lze prolomit pomocí tohoto útoku. Musíme si ale uvědomit, kolik znaků heslo obsahuje. Heslo o délce 6 znaků dokážeme prolomit okamžitě. Nevýhoda útoku spočívá, když heslo začíná obsahovat více znaků. Delší hesla lze prolomit za pár minut, hodin, dní, let. Pokud heslo obsahuje všechny symboly jako čísla, malá a velká písmena, specialní znaky a je dlouhé alespoň 11 znaků, heslo se stává prakticky neprolomitelné pomocí útoku hrubou silou [7].

V tabulce je zobrazeno, jakou rychlostí lze prolomit různě dlouhá hesla. Každý rok vychází nové technologie procesoru a grafických karet, které ovlivňují rychlost generování kombinací hesel, proto se tato tabulka hesel každým rokem mění.

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	1 secs
7	Instantly	Instantly	6 secs	21 secs	50 secs
8	Instantly	1 secs	5 mins	22 mins	59 mins
9	Instantly	33 secs	5 hours	23 hours	3 days
10	Instantly	14 mins	1 weeks	2 months	7 months
11	1 secs	6 hours	1 years	10 years	38 years
12	6 secs	7 days	76 years	623 years	2k years
13	1 mins	6 months	3k years	38k years	187k years
14	10 mins	12 years	204k years	2m years	13m years
15	2 hours	324 years	10m years	148m years	917m years
16	17 hours	8k years	552m years	9bn years	64bn years
17	1 weeks	219k years	28bn years	571bn years	4tn years
18	2 months	5m years	1tn years	35tn years	314tn years

Obrázek 1: Tabulka ukazuje maximální dobu prolomení hesla pomocí brute force techniky v roce 2023 [8].

2.3.4 Credential Stuffing (Tabulky odcizených údajů)

Tabulky odcizených údajů využívají ke svému prospěchu odcizená uživatelská jména a hesla, která se používala nebo stále používají k přihlašování do webových stránek. Uniklé uživatelské jméno a heslo uživatele může uživatel použít ke svému přihlášení například do bankovnictví [9].

Útočník vkládá uniklý seznam uživatelských jmen a hesel do přihlašovacího formuláře. Při útoku útočník doufá, že některé údaje ze seznamu budou využívány k přihlašení do webového serveru. Pokud uživatel používá stejné údaje k přihlašování, je velmi pravděpodobné, že se hacker do účtu dostane[9].

Tabulky odcizených údajů jsou populární technikou kvůli velkému množství uniklých uživatelských jmen a hesel. Tyto údaje lze volně stáhnout na internetu. Některé údaje jsou vyměnovány nebo prodávány na černém trhu [9].

Podle statistik mají tyto útoky velmi malou úspěšnost. Udává se úspešnost k prolomení účtu kolem 0.1 %. Kolekce údajů obsahují miliony až miliardy přihlašovacíh údajů. Když útočník využije milion přihlašovacích údajů, podle statistik by mohl prolomit tisíc účtů. Podaří se prolomit menší procento účtů, než je udáváno. Útok má pořád svůj smysl. Většinou se jedná o čísla kreditních karet nebo citlivé údaje, které mohou být použity dál pro další útoky, jako je phising. Hacker může tyto kolekce využít k prolomení dalších uživatelských účtů na jiném webovém serveru [9].

Efektivitu útoku zvyšuje využití automatizovaných botů. Zabezpečené stránky používají k obraně serveru banování IP adresy po několika neúspěšných pokusech přihlášení do stránky. Moderní boti se vydávají vždy za jiné zařízení a přihlašují se z jiné IP adresy, čímž se tento útok jeví jako normální přihlášení uživatele k serveru a je velmi efektivní [9].

2.3.5 Password spraying

Passwords spraying je podobný útoku hrubou silou. Kyberzločinci takzvaně sprejují obvyklá hesla k získání přístupu do cizího účtu [10].

K použití potřebuje útočník vlastnit seznam uživatelských jmen a seznam uživatelských hesel. K útoku se také vyžívají automatizovaní boti [10].

Hacker vkládá do přihlašovacího formuláře vždy jiné přihlašovací jméno a pro každý účet zkouší stejné heslo, ze seznamů, které využívá. Tento postup opakuje pořád dokola, dokud se útočníkovi nepodaří získat přístup k účtu. Díky tomuto způsobu dokáže útočník jednoduše obejít zabezpečení servu před blokováním účtu nebo blokováním IP adresy [10].

2.3.6 Hybrid attack (Hybridní útok hrubou silou)

Hybridní útok je technika, která využívá pro větší efektivitu spojení slovníkového útoku s útokem hrubou silou. K provedení je potřeba stejně jako u slovníkového útoku mít seznam s hesly. Hybridní útok hrubou silou upravuje každé heslo ze seznamu [3].

Tento útok si můžeme představit jako lidskou osobu, která je pořád stejná, ale podle naší volby si můžeme upravovat vnější vzhled, vlasy, oblečení [3].

Například ve školách nebo firmách můžeme jako uživatelé získat heslo, které je generované podle našeho jména a datumu narození. Když máme Jana Nováka, který se narodil 1. ledna 2000, jeho heslo by mohlo vypadat jnovak01012000. Samozřejmě jsme donuceni si toto heslo změnit, ale jsou výjimky lidí, kteří toto pravidlo nedodrží. Pro prolomení tohoto hesla se dokonale hodí hybridní útok. Stačí vlastnit seznam se studenty školy. Modifikace jména studenta pro útok hrubou silou by mohla vypadat nějak takto (iniciál křestního jména)(příjmení)[0-9][09][0-9][09][0-9][09][0-9][09]. Tímto způsembom by hacker prolomil všechna hesla studentů, kteří si nezměnili heslo za několik minut [3].

Hybridní útok se vyplatí, když máme představu, jak je heslo formátováno.

Když bude útočník vlastnit uniklou databázi hashe hesel, vyzkouší nejdříve slovníkový útok pro jejich prolomení, pokud mu po slovníkovém útoku zůstane mnoho neprolomených hesel, přikloní se k použití hybridního útoku. Přibližný formát hesla, který stránka používá, hacker nalezne jednoduše při vytváření nového hesla na stránce, odkud hashe pocházejí. Zjistí, zda je potřeba použít v hesle velké písmeno, číslice či speciální znak. Formát hesla se vytvoří podle nejlehčích způsobů, které jsou lidé schopni vymyslet, protože jsou líní si tato hesla pamatovat. Formát pro hesla, která musí obsahovat alepoň jedno číslo a jeden speciální znak, by mohl vypadat: (heslo)[0-9][speciální znak] [3].

2.3.7 Reverse brute force attack (Otočený útok hrubou silou)

Otočený útok hrubou silou je velmi podobný útoku hrubou silou. Tato metoda se použije za předpokladu, že útočník zná uživatelské heslo. Místo hledání různých kombinací hesla se generují různá přihlašovací jména účtu, dokud kombinace jména není správná [11].

2.3.8 Mask attack

Mask attack funguje stejně jako útok hrubou silou, ale s upřesněním formátu hesla. Populární nástroj pro aplikování Mask attacku je Hascat. Pokud hacker má představu o formátu hesla, které chce prolomit, může využít Hascat. Tento nástroj umožní útočníkovi přidat přesné specifikace, podle kterých má heslo kombinovat [12].

Příklad: webový server požaduje k vytvoření hesla

- alespoň 8 znaků,
- alespoň 1 velký znak,
- alespoň 1 číslo,
- alespoň 1 speciální znak.

Za těchto požadavků můžeme vytvořit heslo jako „Tereza20!“. Pro toto heslo se vytvoří maska v Hascatu. Podle pravidel:

- ?u = velké písmeno,
- ?l = malé písmeno,
- ?d = číslo,
- ?s = speciální znak.

Tereza20! -> ?u?l?l?l?l?l?l?d?d?s [12].

Tímto útokem dokážeme zvýšit efektivitu útoku hrubou silou. Hashcat hledá kombinace podle regulerního výrazu. Pokud ale heslo má jiný formát, než je udáván v regulerním výrazu, útok neuspěje. Tento útok je podobný hybridnímu útoku, ale pro prolamování hesel nevyužívá slovní seznam [12].

2.3.9 Rule based attack

Jedná se o nejsložitější náhodný útok, protože jeho konfigurace je podobná programovacímu jazyku. Při vytváření rule based útoku se použije slovní seznam, ze kterého se čerpají hesla a definují se pravidla. Jeden z nástrojů, který útok může provést, je Hashcat. Program nabízí velkou škálu metod, podle kterých lze znaky měnit. Můžeme také použít předpřipravená pravidla v programu. Připravená pravidla v Hashcatu jsou dělaná podle obvyklých paternů hesel, která lidé používají [13].

Pro heslo „password“ aplikujeme pravidla, pro vytvoření velkého písmena na začátku slova, zaměnění znaku za jiný znak, přidání znaku na konec slova:

- znak c - (capitalise) vytoření velkého písmena na začátku slova,
- znak s - (substitutute) zaměnění znaku hesla za znak určený útočníkem,
- znak $ - přidá určený znak na konec hesla.

Aplikování pravidel na heslo:

- c,
- sa@,
- $1 $2 $3.

specifikovaná pravidla heslo „password“ přetvoří na heslo „P@ssword123“ [14].

Hlavní rozdíl mezi mask útokem a based rule útokem je rychlost. Při rule based útoku se nemusí generovat všemožné kombinace znaků, dokud heslo nebude správné [13].

2.4 Social engineering (Sociální inženýrství)

2.4.1 Phishing

Phising útok je založený na přesvědčení uživatele, že komunikuje s legitimní webovou stránkou nebo osobou. Většinou se jedná o podvodnou zprávu v emailu, která se vydává za reálnou webovou stránku firmy, o které si lidé myslí, že je důvěryhodná. Útočník se snaží falešnou webovou stránku co nejvíce napodobit její originál, aby vypadala co nejvíce důvěryhodně. Čím více důvěryhodná webová stránka je a čím větší je počet oslovených cílových uživatelů, kteří dostanou phishingovou zprávu, tím větší šanci má útočník na úspěch [15].

Hlavním cílem útočníka je ukrást osobní informace nebo přihlašovací údaje. Phisingová zprává využívá časový nátlak na oběť, aby vyplnila potřebné informace do formuláře, pod záminkou možného ztracení peněz na bankovním účtu nebo ztráty své pracovní pozice, ale těchto lákadel na rychlé vyplnění osobních informací existuje mnoho. Uživatelé jsou obelháni jednoduchým trikem, že musí rychle vyplnit požadovaný formulář nebo se jim něco stane. Pod časovým nátlakem nepřemýšlí nad požadavky zprávy a nezastaví se nad tím, jestli dávají smysl. Později uživatelé začínají rozpoznávat zvláštní požadavky, které nedávají moc smysl [15].

Phisingové útoky jsou v neustálém vývoji, aby dokázaly obejít bezpečnostní software a lidskou pozornost, proto firmy musí neustále školit své zaměstnance před nejaktuálnějšími podobami phishingového útoku. Berme na vědomí, že stačí pouze jedna osoba, která podlehne phishingovému útoku, z čehož může vyústit velký unik osobních informacích nebo přihlašovacích údaju ze serveru [15].

2.4.2 Deceptive phishing (Klamný phising)

Klamný phising je nejběžnější technika phishingu. Útočník se vydává za důvěryhodného odesílatele emailu, který se snaží dostat z cílových uživatelů

osobní informace nebo přihlašovací údaje. Tyto emaily se pokouší obelhat uživatele k odhalení jejich osobních informací na základě požádání o ověření účtu, změny hesla nebo provedení platby [16].

K ochraně před klamným phishingem je důležité číst pozorně emailovou adresu, jestli adresa vypadá důvěryhodně. Nalezení obecného oslovení uživatele nebo špatné gramatiky v emailu může být další vodítko pro odhalení klamného emailu [17].

2.4.3 Smishing

Smishing je útok provádějící se přes mobilní telefon, známý jako SMS phishing. Při používání této techniky oběť nedostane podvodný email, ale obrdží SMS zprávu. SMS zpráva může obsahovat URL adresu, která do mobilu nainstaluje malware. Tento malware se může schovávat za normální aplikaci, která po uživatelích bude požadovat přihlášení pomocí uživatelských údajů k přístupu do aplikace. URL adresa může odkazovat i na podvodnou webovou stránku, která požaduje vyplnění formuláře osobních údajů. Typický příklad pro SMS phising je výhra nejnovější iPhone. Jediné co musí udělat uživatel, který dostal SMS zprávu, je zaregistrovat se a zaplatit dopravu [18].

2.4.4 Vishing

Vishingový útok na rozdíl od smishingu nepoužívá k odcizení osobních údajů SMS zprávy, ale telefonní hovory. Někdy jsou hovory uskutečněny automatizovanými boty, které převádějí text na slovní podobu a donutí oběť, aby zavolala na uvedené číslo od útočníka. Hovory můžou být i rovnou iniciovány od strany útočníka. Útočník se obvykle vydává za finančního poradce, banku či policii. Pomocí psychického nátlatku se snaží oběť přinutit, aby sdělila své přihlašovací údaje do účtu nebo čísla karet. V dnešní době má tento útok velký potenciál uspět v důsledku nové technologie Machine Learning, která dokáže napodobit jakýkoliv hlas [19].

2.4.5 Spear phishing

Spear phishing cílí na konrektní osoby, o kterých útočník zná nějaké informace. V podvodném emailu může být obsaženo jméno oběti, firma, ve které pracuje, pracovní pozice, telefonní číslo a další různé informace, které pomohou obelstít uživatele k uvěření věrohodnosti emailu. Tyto informace o lidech se dají snadno dohledat na známém webovém portálu LinkedIn. Úkol emailu zůstává stejný jako u klamného emailu, nalákat oběť na falešnou webovou adresu [16].

Pokud osoba obrdží email, který obsahuje její osobní informace, email se stává více důvěryhodným a oběť zranitelnější. Je více pravděpodobné, že ciloví uživatelé nebudou mít podezření o phishingovém útoku. I když se email zdá být od důvěryhodného zdroje, je důležité dávat pozor na příchod nečekaných emailů, které nutí uživatele vyplnit požadované informace[17].

2.4.6 Link manipulation

Phishingové útoky jsou založeny hlavně na URL adresách. Lidem je doporučováno, aby na podezřelé URL adresy vůbec neklikali. Existuje několik řešení, jak podvodné adresy, které vypadají na první pohled podezřele, schovat za URL adresy s důvěryhodnější doménou[20].

Schování URL adresy v textu lze jednoduše zamaskovat do slova například: „vstoupit“ jako hypertextový odkaz nebo vytvoření URL adresy v textu „www.airbank.cz“.Vytvořené hypertextové odkazy se ukazují jako legitmí adresy, ale po kliknutí na odkaz se oběť přesměruje na phishingovou webovou stránku, která vypadá téměř identicky jako originální. Tímto způsobem po přesměrování na podvodnou webovou stránku uživatel snadno zapomene zkontrolovat legitimitu adresy webové stránky v prohlížeči [20].

Adresa s překlepem, jedná se o adresu, která reprezentuje například bankovní adresu „www.airbank.cz“, ale adresa obsahuje znak, který se snadno přehlédne „www.airbanks.cz“ . Podvodná webová adresa je téměř indentická jako originální [20].

2.4.7 Whaling (CEO fraud)

Whaling je mířený phishingový útok na takzvanou velrybu, což znamená vysoce postavavená osoba ve firmě. Hlavní cíl útoku se soustředí na převod velké částky peněz útočníkovi. Obvykle se útočník vydává za nadřízenou osobu. Email je přesně upravený pro danou osobu, využívá osobní a firemní informace, které jsou dostupné na sociálních sítí, na webovém serveru LinkedIn nebo dalších webových stránkách, kde oběť komunikuje. Útočník pro zlepšení důvěry může zmínit i různé situace z osobního setkání, aby podnítil, že komunikuje opravdu se svým nadřízeným, kterého zná. Takhle vytvořený email, s velkým obsahem osobních informací, je velmi težké rozpoznat od pravého emailu [21].

2.4.8 Content spoofing

Content spoofing je chyba webového zabezpečení. Útočník využije slabých míst webového serveru pro svoji úpravu obsahu na webové stránce. Toto umožňuje útočníkovi přesměrovat různé akce, které provádí uživatel, jako je například klikání na tlačítka, a přes odkaz dostat oběť na svoji phishingovou stránku. Uživatel se volně pohybuje na legitimní stránce a najednou se ocitne na phishingové webové stránce bez povšimnutí [22].

2.4.9 "Evil Twin" Wi-Fi

Evil Twin útok znamená druhý Wi-Fi access point v okolí se stejným SSID, které je používáno na pravém Wi-Fi access pointu. Tyto access pointy jsou většinou v místech, kde se pohybuje hodně lidí, jako jsou kavárny, letiště, knihovny a nádraží. Útočník doufá, že uživatelé Wi-Fi zvolí právě jeho acces point pro připojení. Může to ovlivnit přemístěním blíž k oběti, aby se lidem zobrazoval lepší signál Wi-Fi, který je většinou donutí se připojit k silnější síti. Jakmile se uživatel připojí k Wi-Fi, obvykle jsou ve veřejných sítích povinni přihlásit se k autorizační stránce. U falešného acces pointu se zobrazí idetická autorizační stránka pro přihlášení k Wi-Fi. Už v této chvílí útočník

sbírá přihlašovací údaje uživatelů. Dále útočník je schopen provádět útok man-in-the-middle, ve kterém vidí aktivitu uživatelů v síti a může sbírat například platební informace z karty při právě provedené platbě. Pokud uživatel nepoužívá VPN nebo stránka není zabezpečená HTTPS protokolem, hrozí odcizení těchto soukromých informací. Při tomto útoku uživatelé nemají šanci zjistit, že jsou pod phishingovým útokem. Access point poskytuje Wi-Fi službu bez žádných omezení a chová se jako normální [23].

2.4.10 Pharming

Pharming útok využívá DNS servery pro přesměrování domény na phishingové webové stránky. Když uživatel zadá webovou stránku do prohlížeče, potřebuje DNS server, aby tuto doménu překonvertoval na IP adresu, útočník přesměruje doménu na svoji IP adresu phishingové stránky. Po přesměrování se uživateli zobrazí indentická falešná webová stránka, kterou zadal. To znamená při zadání „www.airbank.cz“ je uživatel přesměrován na jinou webovou stránku, než zamýšlel. Většinou se jedná o kopii zamýšlené webové stránky, aby uživatel nic nepoznal. Jméno domény v prohlížeči zůstává stejné. Existují dva způsoby, jak ovlivnit DNS server ve prospěch útočníka [24].

První přístup spočívá ve využití malwaru, který přesměruje IP adresu v cílovém počítači. Obvykle pošle nebezpečný email se souborem, po stáhnutí souboru do počítače se malware nainstaluje. Malware má za úkol upravit sobour hosts, ve kterém přepíše IP adresu domény na svoji falešnou webovou stránku [24].

Druhá technika se soutředí na účelovou úpravu DNS serveru. Útočník je schopen upravovat DNS tabulku, ve které jsou uloženy IP adresy domén. V této tabulce přepíše IP adresy domén na svoje vlastní IP adresy phishingových stránek. Upravení DNS serveru ovlivní všechny uživatele, kteří využívají tento server pro směrování domén [24].

2.4.11 Angler phishing

Jedná se o jednu z novější technik phishingu. Angler phishing se využívá na sociálních sítích. Útočník vytvoří falšený účet firmy na sociální síti. Většinou se zaměří na uživatale, kteří si stěžují na sociálních sítích na neschopnost firmy. Když útočník nalezne oběť, vytvoří si falešný účet zmíněné firmy a kontaktuje nespokojeného zákazníka pod záminkou zlepšení služeb. Při komunikaci útočník zjištťuje osobní informace zákazníka nebo ho přesměruje odkazem na phishingovou stránku, kde musí vyplnit požadované informace [25].

2.4.12 Watering hole

Útočník pro získaní informací cílové osoby se nepokouší spáchat útok napřímo, ale využije webovou stránku se špatným zabezpečením, kterou vyhlédnutá oběť často používá. Útočník webovou stránku infikuje s nebezpečným kódem, obvykle JavaScript nebo HTML. Jakmile oběť navštíví webovu stránku, nebezpečný kód se spustí a infikuje počítač uživatele. Tento proces může být automatizován nebo je nutné, aby si uživatel nebezpečný soubor s kódem stáhnul sám. Po infikování počítače útočník může shromažďovat osobní infomace oběti nebo používat počítač k dalším útokům [26].

2.4.13 Shoulder surfing

Shoulder surfing si můžeme představit jako pozorování přes rameno. Úkol útoku je zaznamenat pozorováním osobní informace uživatele při přihlašování do systému. Útočník se při útoku snaží nenápadně dostat k oběti, přičemž dbá na dobrý výhled do klávesnice nebo obrazovky oběti. Útok se nemusí provádět pouze fyzickým kontaktem, je možné zaznamenávat informace pomocí nastražených kamer nebo fotoaparatů [27].

2.4.14 Clickjacking-baiting

Clickjacking využívá přídání další vrstvy přes webovou stránku. Vrstva nese v sobě vytvořené html elementy, většinou tlačítka. Tyto elementy se nachází na místech, kam s velkou pravěpodonností návštěvníci kliknou. Když uživatel klikne například na tlačítko pro zavření reklamy, na pozadí se může spustit funkce, kterou uživatel nezamýšlel spustit a o jejím běhu nemá tušení [28].

Cíle clickjackingu jsou různé, útočník si může převést peníze uživatele na svůj účet, lajkování profilů na sociálních sítí, zapnutí kamery nebo mikrofonu v počítači, vytvořit objednávku položky, kterou musí uživatel zaplatit. Pro úspěšné provedení musí být uživatel přihlášen na webové stránce, kterou chce hacker využít. Když je útok mířen na převod peněz, uživatel musí být přihlášen v internetovém bankovnictví [28].

Situaci si můžeme představit pod nalákáním uživatele vyskakovací reklamou na velmi zajímavý výlet zdarma. Při kliknutí na reklamu útočník zkontroluje, jestli je uživatel přihlášen do internetového bankovnitví. Pokud ano, útočníkovo číslo účtu je vyplněno do platebního formuláře v internetovém bankovnicví a uživatel je přesměrován na webovou stránku výletu. Pro úspěšné zarezervování výletu uživatel musí kliknout na tlačítko rezervovat, na kterém se skrývá neviditelná vrstva. Tímto kliknutím ale nevědomky potvrdí platbu převodu peněz na bankovní účet útočníka. Uživatel je poté přesměrován na stránku s informacemi o výletu, který ovšem neexistuje [28].

2.4.15 Password for purchase (Prodej hesel)

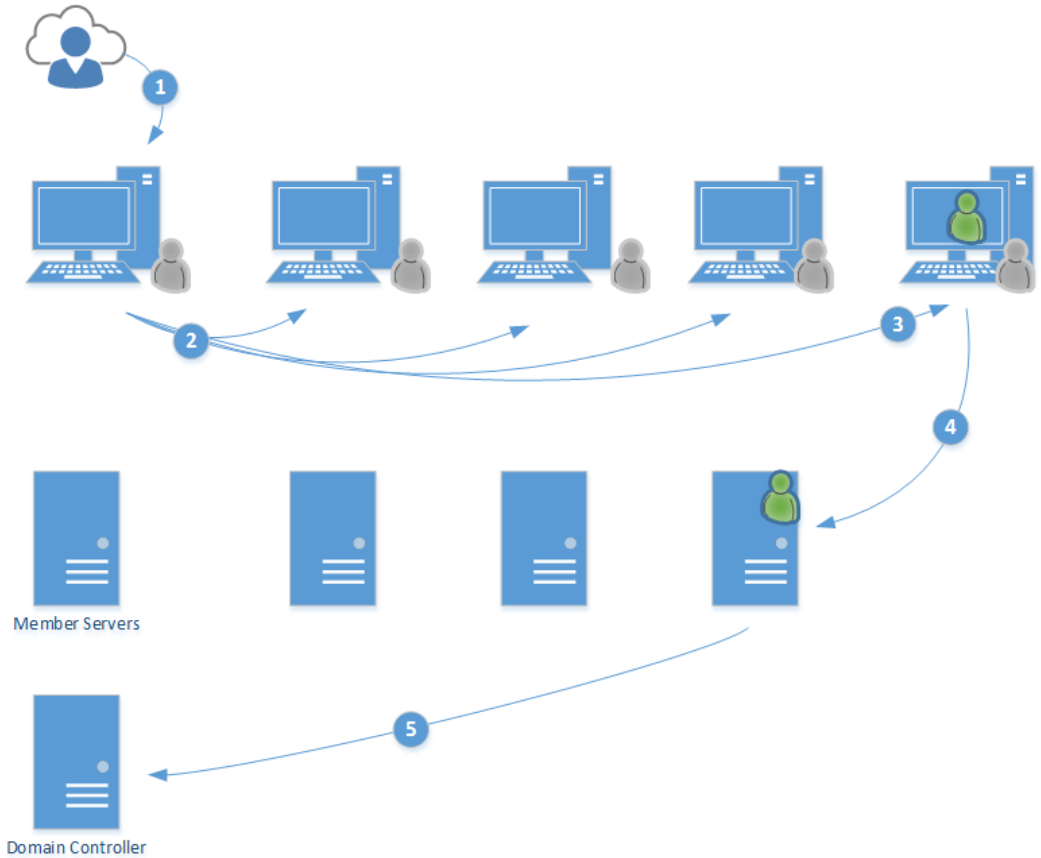
Na osobních údajích lze vydělávat prodejem odcizených seznamů, které pak může někdo jiný použít k nabourání se do systému. Seznamy hesel, zahashovaných hesel a duhových tabulek jsou dostupné na dark webu k volnému stáhnutí z internetu. Některé seznamy jsou k dostání pouze po zaplacení peněžní částky. Na dark web útočník nahraje uniklé seznamy, které buď odcizil prolomením systému nebo pracuje jako zaměstanec ve firmě a má

přístup k seznamům přihlašovacích údajů[1].

2.5 Hash Based Attacks (Útoky založené na hashi)

2.5.1 Pass the hash attack

Při použití pass the hash útoku se útočník dokáže připojit k účtu v síti bez znalosti čistého hesla. Pro připojení útočníkovi stačí uživatelské jméno a hash hesla. Útok funguje laterálně a vertikálně. Připojí se do prvního účtu, do kterého znal uživatelské jméno a hash hesla. Útočník v systému odchytává další uživatelská jména a hesla hashe, která mu dovolují postupovat dál v síti. V každém systému hledá další uživatelské údaje. Útočníkův cíl je získat administrátorské údaje, které mu dají přístup k domain controleru. Pokud útočník dostane přístup do domain controleru, bude mít přístup k celé síti [29].



Obrázek 2: Postup útočníka v síti [30].

Hashe hesel jsou používána v NTLM protokolu, který se stará o autentifikaci uživatele do systému. Funguje na principu

1. zadání uživatelského jména a hesla v textu,
2. počítač zahashuje zadané heslo podle hashovacího algoritmu,
3. počítač odešle k nejbližšímu domain controleru žádost o přihlášení s přihlašovacím jménem,
4. domain controler odešle náhodné číslo známé jako logon challenge,
5. počítač zašifruje náhodné číslo pomocí hashe hesla a odešle na domain controler,
6. domain controler obsahuje databázi zahashovaných hesel pro všechy uživatele, kteří se připojují k sítí. Pokud se při porovnání hashe shodují v databázi a na počítači, uživatel je autentifikován a dostane přístup.

Pass the hash obchází první a druhý krok. Pokud útočník vlastní hash hesla, může přejít rovnou ke třetímu kroku k odeslání logon requestu. Pro útok se vyžívají různé softwary, mezi nejznámější patří mimikatz nebo metasploit [29].

Když se útočník dostane do systému, může se laterálně pohybovat po sítí. Každý počítač má LSASS pamět, která obsahuje všechny hashe hesel přihlášených uživatelů. Některé z těchto účtů jsou více privilegované například jako administátor sítě. Když se administrátor sítě připojí vzdáleně na server, jeho uživatelské jméno a hash hesla se uloží do LSASS paměti. Odcizeních těchto údajů umožní útočníkovi přístup po celé síti a může získát všechny hashe hesel z domain controleru[29].

2.5.2 Lookup Tables (Vyhledávací tabulky)

Existuje několik způsobů, jak lámat hashe hesel. Jeden z těchto způsobů spočívá v hashováváním každého hesla v textové podobě do hashe, dokud výsledný hash hesla se neshoduje s cílovým hashem. Tímto způsobem se vytvářejí i takzvané vyhledávací tabulky [31].

Vyhledávací tabulka obsahuje řádky s heslem v textové podobě a hashem hesla. Pro efektivnost musí tabulka obsahovat velké množství záznamů. Nevýhoda takového řešení se odrazí na velkém zahlcení místa na disku a snížení rychlosti vyhledávání. Generování hashe pro každé heslo zabere nějaký čas. Každé heslo v textové podobě s hashem hesla se musí uložit na disk [32].

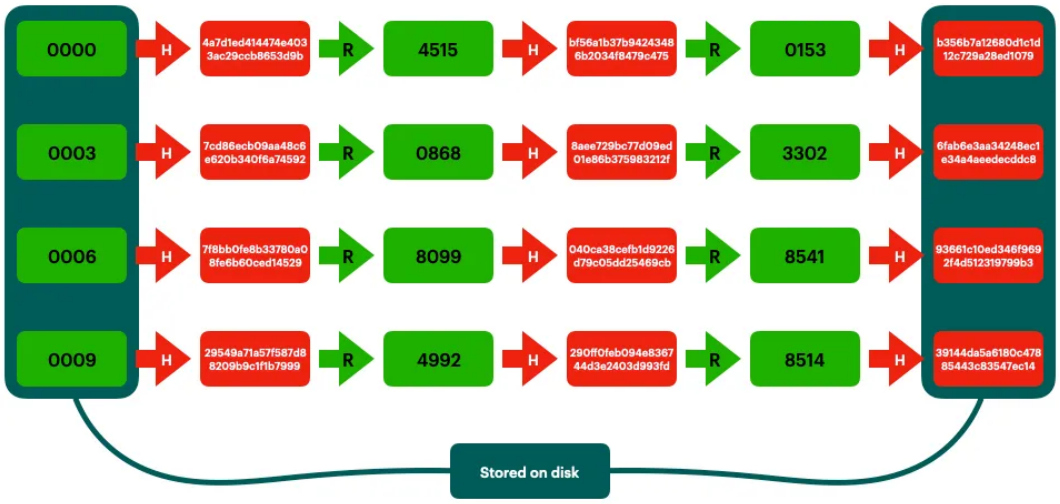
2.5.3 Rainbow tables (Duhové tabulky)

Duhové tabulky umožní útočníkovi rychle prolomit hash hesla, pokud je toto heslo uloženo v duhové tabulce.

Způsob, který využívají duhové tabulky, je komplikovanější. V duhové tabulce není uložený hash pro každé heslo, ale řetěz, pomocí kterého útočník dokáže najít heslo v textové podobě nebo prolomit hash hesla. Pro vytvoření řetězu je nutné použít hashovací funkci a takzvanou redukční funkci. Hashovací fuknce zahashuje heslo v textové podobě do hashe a redukční funkce z hashe hesla vytvoří nové heslo v textové podobě [31].

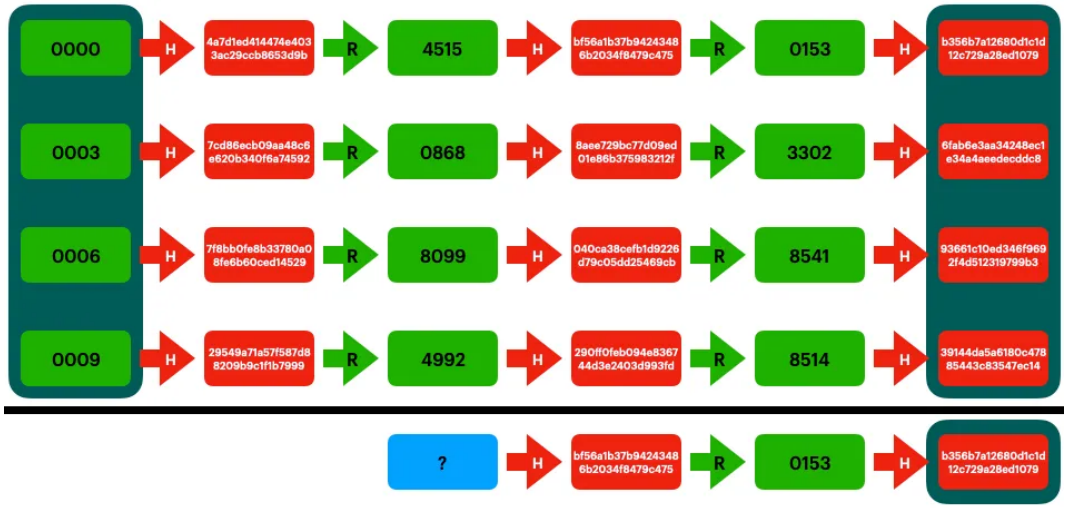
Do jednoho řetězu se může uložit několik hesel, záleží na velikosti řetězu, kolik bude obsahovat záznamů. Při vytváření řetězu se nejprve zahashuje heslo. Z vygenerovaného hashe se pomocí redukční fuknce překonvertuje hash na jiné textové heslo, které se opět zahashuje. Tato posloupnost pokračuje podle toho, kolik bude řetěz obsahovat počet hesel. V duhové tabulce je řetěz uložen jako první záznam vstupu (heslo v textu) a poslední záznam výstupu (hash hesla). V duhové tabulce je nutno ukládat metadata jako je délka řetězu a redukční funkce, aby bylo možné vyhledávání v tabulce. Tento způsob oproti vyhledávací tabulce výrazně sníží velikost zabraného místa na disku a zvýší

rychlost vyhledávání [32].



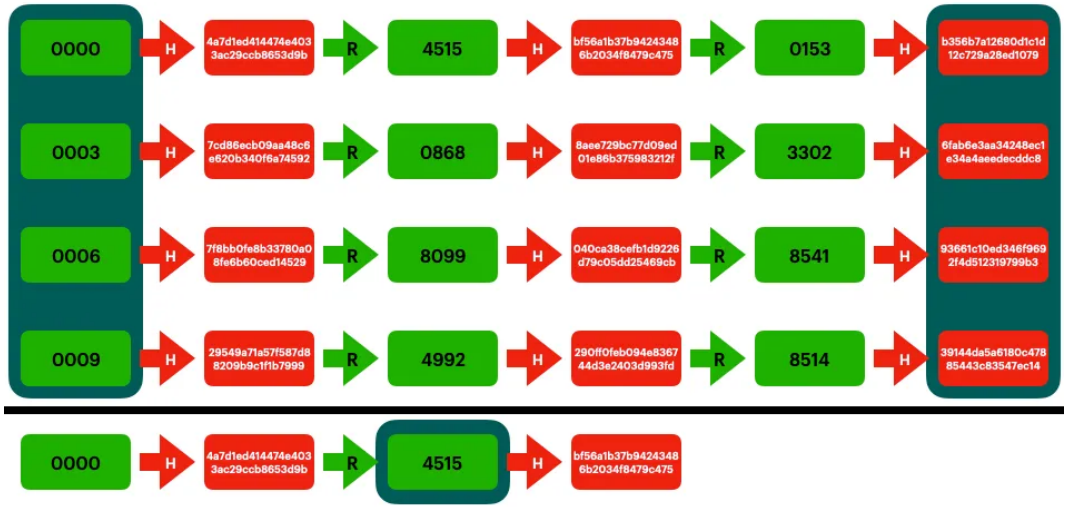
Obrázek 3: Znázornění dat, která se ukládají na disk [32].

V řetězu je možné postupovat pouze jedním směrem, není možné procházet řetěz pozpátku. Při vyhledávání v duhové tabulce útočník vyplní vstup (textové heslo nebo hash hesla). Pro zadaný vstup proběhně stejná posloupnost kroků jako při vytváření řetězu [32].



Obrázek 4: Znázornění, jak probíhá hledání shody v duhových tabulkách [32].

Na obrázku 4 je vidět, že se vyplnil hash hesla. Pro jeho prolomení je nutné řetěz znovu vytvořit, aby bylo možné získat předchozí hodnotu, protože řetěz nelze procházet pozpátku. Nejdřív se musí dojít na konec řetězu. Duhová tabulka jako výstup vyhodí svůj konečný záznam, který je vždy reprezentován jako hash hesla. Tento konečný hash hesla má k sobě přiřazený první vstup do řetězu. Pro dokončení hledání hesla se v duhové tabulce se použije první vstup řetězu, který se zjistil v předchozím kroku. Pro tento vstup proběhně posloupnost funkcí. Z toho postupu se vytvoří celý řetěz, ze kterého je už známá požadovaná hodnota [32].



Obrázek 5: Znázornění, jak probíhá konečné hledání hesla v textové podobě [32].

3 Praktická část

Cílem praktické části je zabezpečit webovou stránku Bobříka informatiky pomocí tabulek odcizených údajů. Odcizená hesla hackery nemusí sloužit pouze pro páchání škody, ale i pro zabezpečení webové stránky nebo dalších jiných aplikací, kde je potřeba se přihlásit. Tyto odcizené údaje využijeme k porovnávání uživatelských údajů při přihlášení uživatele do webové stránky Bobříka informatiky. Tímto způsobem je možné určit, zda uživatel používá bezpečné přihlašovací údaje. Při nalezení shody s tabulkou odcizených údajů dostane uživatel upozornění o nebezpečnosti jeho hesla a bude vyzván ke změně. Z těchto záznamů se pokusíme analyzovat, jak silná hesla uživatelé používají, jestli přistoupili na změnu hesla, za jak dlouho si heslo změnili a jak silné heslo zvolili oproti starému. Druhým cílem praktické části práce je získat data pro analýzu síly hesel uživatelů, jejichž přihlašovací údaje nebyly odcizeny (hacknuty).

3.1 Analýza požadavků

Primárním cílem modulu je zvýšit bezpečnost uživatelských údajů na webové stránce Bobříka informatiky, pomocí tabulek odcizených údajů a pro analyzování síly hesla uživatele.

Při přihlášení do systému modul porovnává uživatelský email a heslo s emailem a heslem v databázové tabulce odcizených údajů. Pokud se při porovnání těchto údajů nalezne shoda, uživatel obdrží na hlavní stránce Bobříka informatiky upozornění o jeho nebezpečnosti uživatelských údajů, které používá, a bude vyzván ke změně údajů. Při nalezení shody se odešlou do databáze záznamy, jako síla právě používaného hesla a čas, kdy údaje byly nalezené jako nebezpečné.

Další funkcí modulu při přihlášení do webové stránky Bobříka informatiky bude hodnocení síly uživatelského hesla podle jeho struktury. Heslo bude bodováno každému uživateli bez ohledu na to, zda byly jeho přihlašovací údaje

odcizeny, či nikoliv. Další zpracování této síly hesla se bude lišit podle toho, zda byly přihlašovací údaje daného uživatele odcizeny či nikoliv.

Vytvoříme algoritmus metriky hesel, který na základě po sobě jdoucích znaků oboduje sílu hesla. Čím více heslo obdrží bodů, tím je heslo silnější. Číselná hodnota se poté bude ukládát do databáze Bobříka informatiky.

Modul bude při změně hesla bodovat pouze uživatele, kteří mají odcizené uživatelské údaje a jsou vyzváni ke změně aktuálního hesla. Po změně se do stejné databázové tabulky, jako při nalezení shody uživateslkých údajů, uloží síla nového hesla, čas, kdy bylo heslo změněno a číselná hodnota 0 nebo 1, která udává, jestli nově nastavené heslo se nachází v tabulce odcizených údajů.

3.2 Získání tabulek odcizených údajů

Pro vylepšení ochrany webové stránky Bobříka informatiky plánujeme použít tabulky odcizených údajů Collections 1#-5#. Tyto kolekce obsahují přibližně 2,7 miliardy záznamů, z toho 1,2 miliardy unikátních emailových adres s hesly, 773 milionů unikátních emailových adres a 21 milionů unikátních hesel v textové podobě. Celková velikost dat Collections #1-#5 se blíží k 1 terabytu [33].

Pro optimalizaci databáze je nutné v kolekcích najít české a slovenské uživatelské údaje. Záznamy budeme vyhledávat podle doménového jména emailu, které končilo zkratkou cz nebo sk. Pro třídění záznamů použijeme 010 editor, který se využívá pro editování velkých textových souborů.

Po prohledání všech textových souborů v kolekcích jsme úspěšně nalezli přibližně 100 milonů českých a slovenských údajů. Vytříděný záznam obsahuje i uživatelské údaje z jiných zemí, neboť některé emailové adresy obsahovaly klíčová slova cz a sk, podle kterých jsme vyhledávali. Emailové adresy například gmail.com, které neobsahovaly české nebo slovenské domény, jsme nebyli schopni dohledat v Collections #1-#5 z důvodu náročnosti rozeznání státní příslušnosti. Dále jsme do databáze nepřidávali přihlašovací uživatelská jména,

protože by bylo velmi obtížné vyhledat české a slovenské přihlašovací jméno v Collections #1-#5. Možností bylo importovat všechna uživatelská jména a hesla pro celý svět, ale velikost tabulky by byla extrémně velká.

3.3 Návrh modulu

Modul bude vytvořen ve frameworku Joomla. Tento framework je vytvořen v jazyce PhP a používá MySQL pro ukládání dat do databáze. Jedná se o open source systém pro správu obsahu webové stránky (content management system - CMS). CMS je webová aplikace, která umožňuje uživatelům spravovat obsah stránky, data, informace webové stránky nebo intranet aplikace. Ve správě obsahu, uživatel Joomly může vytvářet, editovat, archivovat a publikovat svůj obsah na webové stránce[34].

Pro přidání modulu, který kontroluje bezpečnost uživatelských údajů, na webové stránce Bobříka informatiky, bude zapotřebí v adresáři webového serveru (var/www/html) v komponentě users upravit kontrolér, konkrétně metodu login() v souboru users.php.

V souboru users.php bude také vytvořena nová metoda metrikaHesel(), která bude počítat sílu hesla v číselných hodnotách.

Modul pro zaznamenávání specifických záznamů při změně uživatelského hesla bude implementován do adresáře webového serveru (var/www/html) v komponentě users, ve které bude zapotřebí upravit model, metodu save() v souboru profile.php

3.3.1 Databáze

Pro vytvoření funkčního modulu je zapotřebí vytvořit 3 nové tabulky v databázi Bobříka informatiky. Databázová tabulka *ibobr_hacknute_hesla2* slouží k uložení tabulek odcizených údajů z databází Collections 1#-5#. Tabulka obsahuje 3 atributy

- *id* (integer) - primární klíč,

- *email* (varchar),
- *heslo* (varchar).

Sloupec email a heslo slouží k uložení odcizených emailů a hesel.

Pro ukládání dat hacknutých uživatelů je vytvořená tabulka *ibobr_hacknnuti_uzivatele*. Tabulka obsahuje

- *id* (integer) - primární klíč,
- *id_user* (integer) - cizí klíč, který odkazuje na přihlášeného uživatele z tabulky jos_users,
- *id_shoda* (integer) - cizí klíč, který odkazuje na hacknutý záznam uživatele z tabulky, hacknutá hesla,
- *čas nálezu shody* (date),
- *čas změny hesla* (date),
- *metrika starého hesla* (integer),
- *metrika nového hesla* (integer),
- *nové hacknuté heslo* (tinyint).

Do tabulky se ukládají informace při přihlášení uživatele do systému a při změně hesla uživatele. Při přihlášení do systému a nalezení shody zadávaných uživatelských údajů s údaji v tabulce *ibobr_hacknute_hesla2*, se do tabulky *ibobr_hacknnuti_uzivatele* vloží id, id user, čas nálezu shody, metrika starého hesla. Při změně hesla uživatele, jehož údaje byly identifikovány jako odcizené, se do tabulky *ibobr_hacknnuti_uzivatele* vloží čas změny hesla, metrika nového hesla a nové hacknuté heslo.

Časové údaje jsou v tabulce vytvořeny, aby bylo možné analyzovat, za jak dlouho si uživatel změnil své heslo po obdržení oznámení o jeho nebezpečnosti uživatelských údajů. Stará a nová metrika hesel slouží k analyzování síly hesla

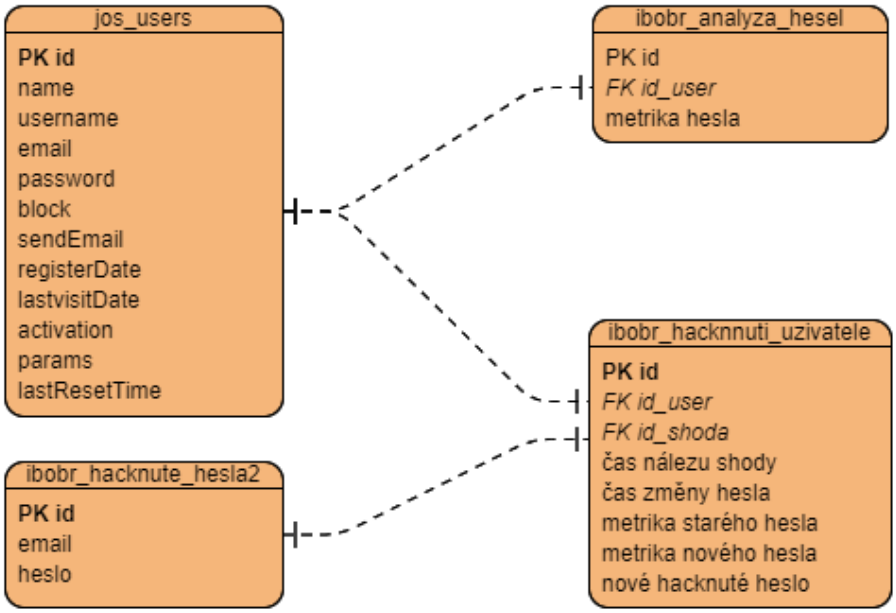
uživatele. Nové hacknuté heslo značí, zda uživatel zvolil nové heslo, které se vyskytuje v tabulce *ibobr_hacknute_hesla2*.

Pro zapisování síly metriky hesel uživatelů, kteří mají bezpečné přihlašovací údaje, slouží tabulka *ibobr_analyza_hesel*, která obsahuje tyto atributy

- *id* (integer) - primární klíč,
- *id_users* (integer) - cizí klíč,
- *metrika hesel* (integer).

Tabulky jsou navržené takovým způsobem, aby nebylo možné číst hesla uživatelů z databáze. V databázi je vidět pouze číselná hodnota hesla uživatele. Tabulka odcizených údajů obsahuje uživatelské údaje v otevřené podobě. Tyto záznamy jsou ale dostupné na internetu, takže se nejedná o zvýšení bezpečnostních rizik webové stránky Bobříka infromatiky.

Je možné najít konkrétní uživatelské údaje uživatele, kterému byla nalezena shoda. Při nalezení shody se id záznamu v tabulce odcizených údajů uloží do tabulky *ibobr_hacknnuti_uzivatele* do sloupce id_shoda. Podle tohoto cizího klíče lze dohledat v tabulce *ibobr_hacknute_hesla2* jeho hacknuté uživatelské údaje.



Obrázek 6: Vytvořené databázové schéma.

Tabulka *jos_users* již byla vytvořena v systému. Tato tabulka obsahuje záznamy registrovaných uživatelů a je propojena s tabulkou *ibobr_hacknnuti_uzivatele* a s *ibobr_analyza_hesel* pomocí vazeb 1:1, tím je umožněno dohledat nasbírané záznamy konkrétních uživatelů. Základní atributy tabulky jsou

- *id* (integer) - primární klíč,
- *name* (varchar),
- *username* (varchar),
- *email* (varchar).

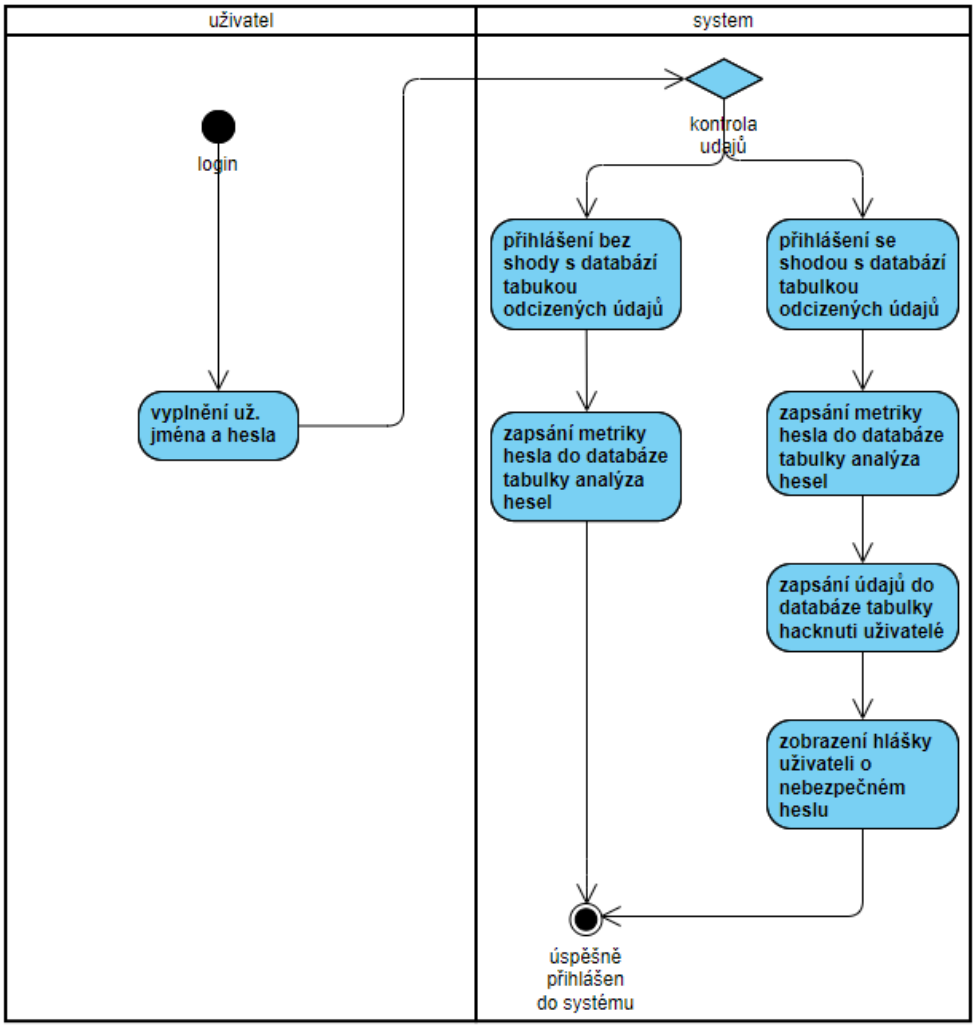
3.3.2 Princip fungování modulu

Pro vstup do webové stránky Bobříka informatiky musí uživatel vyplnit svoje uživatelské jméno a heslo, po vyplnění přihlašovacího formuláře se uživatelské údaje porovnají s databázovou tabulkou *ibobr_hacknute_hesla2* (vyplněné uživatelské údaje se porovnanjí s řádky tabulky emailem a heslem). Když se

v databázi vyhodnotí shoda, do databázové tabulky *ibobr_hacknnuti_uzivatele* se vloží nový záznam (id, id user, čas nálezu shody, metrika starého hesla), uživatel se úspěšně přihlásí do webové stránky a po přihlášení se uživateli zobrazí chybová hláška, která vyzývá uživatele ke změně hesla.

Po úspěšném přihlášení bez shody se v databázi vkládájí data (id, id_user, metrika hesla) do tabulky *ibobr_analyza_hesel*.

Při změně hesla upravený modul zaznamená pouze uživatele, kteří mají odcizené uživatelské údaje. Po vyplnění formuláře pro změnu hesla se do databázové tabulky *ibobr_hacknnuti_uzivatele* aktualizuje záznam reprezentující daného uživatele, konkrétně se vloží (čas změny hesla, metrika nového hesla a nové hacknuté heslo). Sloupec nové hacknuté heslo slouží k vložení záznamu hodnoty True nebo False. Pokud se nové zvolené heslo nachází, ve sloupci hesla v databázové tabulce *ibobr_hacknute_hesla2*, do databázové tabulky, do sloupce nové hacknuté heslo se vloží 1. Pokud nově zvolené heslo nebude mít shodu, do sloupce nové hacknuté heslo se vloží 0. 1, udává hodnotu True, 0 udává hodnotu False.



Obrázek 7: Activity diagram - modul pro přihlášení.

3.3.3 Návrh metriky hesel

Algoritmus pro vyhodnocování metriky hesel je navhrnut jako bodový systém. Čím více heslo obdrží bodů, tím silnější heslo je.

V hesle je možné používat malé znaky, velké znaky, čísla a speciální znaky. Každý z těchto znaků algoritmus počítá jinak. Zaleží, jak jsou znaky po sobě poskládány. Algoritmus metriky hesel prochází po sobě jdoucí znaky. Každý znak v hesle je bodován podle předem připravených podmínek. Výsledná hodnota algoritmu se skládá ze součtu bodů všech znaků.

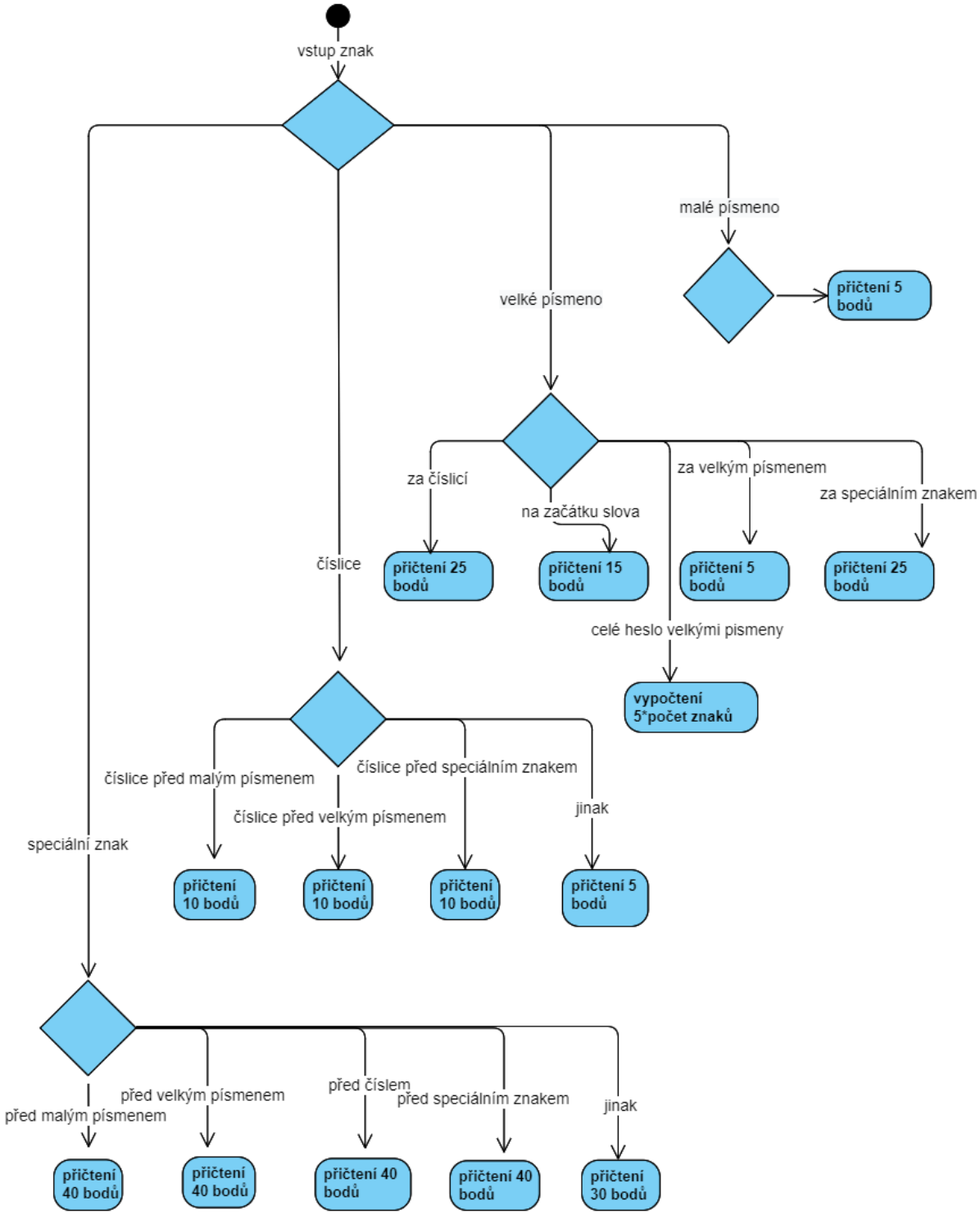
Při vyhodnocování malého písmena v hesle se každé malé písmeno boduje

za 5 bodů.

Při bodování velkého písmena záleží, na kterém místě se znak nachází. Když se velké písmeno vyskytuje na začátku hesla, je mu přiděleno 15 bodů. Pokud se velké písmeno vyskytuje za číslicí, velké písmeno je bodováno za 25 bodů. Pokud před velkým písmenem je speciální znak, velké písmeno je bodováno za 25 bodů. Pokud se velké písmeno nachází za jiným velkým písmenem, obdrží pouze 5 bodů. Heslo, které je napsáno pouze velkými písmeny, bude obodováno 5 body za každé velké písmeno v hesle.

Při bodování číslice, číslice dostává 10 bodů nebo 5 bodů. Když se číslice nachází před malým nebo velkým písmenem, číslice se oboduje za 10 bodů. Když se před číslicí nachází speciální znak, číslice je obodována za 10 bodů. Když číslice nesplňuje výše uvedené podmínky, je automaticky bodována za 5 bodů.

Při bodování speciálního znaku je možné přidělit 40 nebo 30 bodů. Pokud se speciální znak nachází před malým písmenem, speciální znak je obodován za 40 bodů. Pokud se speciální znak nachází před velkým písmenem, speciální znak je obodován za 40 bodů. Pokud se speciální znak nachází před číslem, speciální znak je obodován za 40 bodů. Pokud se speciální znak nachází před speciálním znakem, speciální znak je obodován za 40 bodů. Když zmíněné podmínky nejsou splněny, speciální znak je automaticky bodován za 30 bodů.



Obrázek 8: Activity diagram - algortimus počítání metriky hesla.

3.4 Porovnání algoritmu metriky hesla s dostupnými online algoritmy

Výsledky našeho algoritmu pro metriku hesla jsme porovnávali na dostupných internetových stránkách, které slouží k hodnocení síly hesla. Během testování hesel jsme získali různé a někdy dokonce i protichůdné výsledky. Pro porovnání jsme zvolili algoritmus od University of Illinois at Chicago, jelikož jeho hodnotící algoritmus je veřejně dostupný.

Vytvořený algoritmus pro výpočet metriky hesla jsme otestovali na 17 různě složených heslech. Tato hesla jsme následně použili na webové stránce https://www.uic.edu/apps/strong-password/, která hodnotí hesla pomocí slovního ohodnocení: velmi slabé, slabé, dobré, silné, velmi silné. V tabulce níže můžete vidět bodové ohodnocení hesla našeho algoritmu a slovní ohodnocení hesla pomocí algoritmu od University of Illinois at Chicago.

heslo	metrika hesla	testované heslo UIC algoritmem
petrpetrpetr	60	slabé
petr1234	40	dobré
Petr1234	50	silné
petrpetr1234	60	dobré
Petrpetr1234	70	velmi silné
Petr1234!	85	velmi silné
P€tr1234	85	velmi silné
P!t3r123	90	velmi silné
P!t33rON	110	silné
PETPETRPETR	60	velmi slabé
PETR1234	50	dobré
PETR1234!	85	silné
PETRPETR1234	70	dobré
P€t3Rs@m3C	180	velmi silné
PetrSamec123456	105	velmi silné
PetrSamec123456!	140	velmi silné

Obrázek 9: Srovnání bodování hesel našeho algoritmu s UIC algoritmem

Výstup obou algoritmů je celkem podobný, ačkoli se u některých hesel trochu liší. Rozdíly se vyskytují v heslech jako „Petr1234!“, „petrpetrpetr“ nebo „Petr1234“. Přestože se oba algoritmy liší ve výsledcích pro některá hesla, je obtížné jednoznačně určit, která metrika je lepší při hodnocení hesel.

3.5 Implementace modulu

3.5.1 Implementace algoritmu metriky hesel

Algoritmus metriky hesel slouží k vyhodnocení síly hesla uživatele. Při přihlášení nebo při změně hesla ve webové stránce Bobříka informatiky, algoritmus vyhodnotí zadané heslo podle po sobě jdoucích znaků, přičemž záleží, jak jsou znaky v hesle za sebou poskládány. Tento algoritmus vrací číselnou hodnotu vyjadřující sílu hesla. Platí, čím větší číselná hodnota, tím silnější heslo uživatel používá.

Ukázka zdrojového kódu algoritmu metriky hesel

```
$pwArr = preg_split('//u', $data['password1'], -1, PREG_SPLIT_NO_EMPTY);
$pw = $data['password1'];
$result=0;

for ($i = 0; $i < count($pwArr); $i++) {

    $uppercase = preg_match('#[A-Z]+#', $pwArr[$i]);
    $lowercase = preg_match('#[a-z]+#', $pwArr[$i]);
    $number = preg_match('#[0-9]#', $pwArr[$i]);
    $specialChars = preg_match("[\W]", $pwArr[$i]);


    if ($uppercase) {
        if (is_numeric($pwArr[$i - 1])) {
            $result += 25;
        } elseif (ctype_upper($pwArr[$i - 1])) {
            $result += 5;
        } elseif ($pwArr[$i - 1] === NULL) {
            $result += 15;
        } elseif ($pwArr[$i - 1] == $specialChars) {
            $result += 25;
        } else {
            $result += 25;
```

```
        }
        if (ctype_upper($pw)) {
            $result = strlen($pw) * 5;
        }

    }elseif ($lowercase){
        $result+=5;
    }elseif ($number){
        if($pwArr[$i+1]!=$lowercase ||$pwArr[$i+1]!=$uppercase){
            $result+=5;
        }elseif ($pwArr[$i+1]===NULL || is_numeric($pwArr[$i+1])){
            $result+=5;
        }else{
            $result+=10;
        }
    }elseif($specialChars){
        if(is_numeric($pwArr[$i+1])){
            $result+=40;
        }elseif ($pwArr[$i+1]===NULL || $pwArr[$i+1]!= $lowercase ||
                    $pwArr[$i+1]!=$uppercase){
            $result+=30;
        }else{
            $result+=40;
        }
    }

}
```

3.5.2 Importování záznamů do databáze

Vytříděné záznamy z Collections #1-#5 jsme postupně importovali do předpřipravené tabulky *ibobr_hacknuta_hesla*, do které jsme vložili pouze email a heslo bez id záznamu. Import jsme prováděli přibližně po 50 velkých megabytových souborech. Při nahrávání záznamů do databáze se často objevovaly syntaktické chyby, které zapříčinily neúspěšný import do databáze. Bylo nutné tyto záznamy upravit do takové podoby, aby každý záznam neobsahoval přebytečné mezery, prázdné řádky a znaky jako dvojtečka nebo středník, podle kterých byl řádek záznamu rozdělen na emailovou adresu a heslo. Některé soubory jsme nebyli schopni importovat ani po těchto úpravách a nemohli jsme najít syntaktickou chybu v souboru, která tento problém způsobuje, proto jsme se rozhodli soubor se záznamy rozdělit na malé části a provést import znovu. Import do databáze po této úpravě proběhl pro některé soubory úspěšně. Ovšem pro soubory s neznámou syntaktickou chybou import nebyl úspěšný, a proto jsme se rozhodli tyto soubory neimportovat. Po dokončení importování záznamů do databáze jsme vygenerovali ID pro každý záznam v tabulce *ibobr_hacknuta_hesla*.

Po importování těchto odcizených záznamů do tabulky *ibobr_hacknuta_hesla*, tabulka obsahovala přibližně 90 miliónů českých a 2 milióny slovenských záznamů. Rychlost při hledání záznamů v tabulce *ibobr_hacknuta_hesla* nebyla uspokojivá pro rychlé vyhledání shody uživatelských údajů se záznamy v tabulce *ibobr_hacknuta_hesla*. Tento problém by zapříčinil pomalé přihlášení uživatele do webové stránky Bobříka informatiky. Pro řešení problému se nabízely různé způsoby, jako indexování tabulky nebo odstranění duplicitních záznamů. Provedli jsme odstranění duplicitních záznamů v tabulce *ibobr_hacknuta_hesla*. Technika, kterou jsme zvolili, se zdá být nejrychleší způsob, jak odstranit duplicitní záznamy v tabulce, která obsahuje velké množství záznamů. Vytvořili jsme novou tabulku s názvem *ibobr_hacknuta_hesla2*, která obsahuje stejné sloupce jako tabulka

ibobr_hacknuta_hesla. Dále jsme použili příkaz INSERT IGNORE INTO „nová tabulka“ * FROM „stará tabulka“. Tento příkaz kopíruje záznamy ze staré tabulky do nové s tím, že do nové tabulky nevkládá záznamy, které už nová tabulka obsahuje.

Po odstranění duplicit v nové tabulce zůstalo pouze 13,5 miliónů záznamů. Hledání v databázi už bylo dostatečně rychlé, aby bylo možné se rychle přihlásit do webu Boříka informatiky. Proč tabulka *ibobr_hacknuta_hesla* obsahovala tolik duplicitních záznamů? Databáze Collections #1-#5 obsahuje uživatelské údaje z různých databází, ve kterých uživatel mohl být hacknut několikrát, ovšem Collections #1-#5 obsahuje záznamy i z phishingových stránek. Jeden z důvodů může být, když uživatelé se pokoušeli přihlásit do phisingové webové stránky, své údaje do formuláře vyplnili několikrát, z důvodu, že webová stránka je nikam dál nepřesměrovala. V souborech s odcizenými údaji je vidět list po sobě jdoucích stejných emailových adres se stejným heslem nebo s jinými hesly, které uživatel nejspíš používal.

3.5.3 Implementace modulu pro přihlášení uživatele

Když se uživatel přihlašuje do webové stránky Bobříka informatiky, používá své uživatelské jméno a heslo. Abychom mohli porovnávat uživatelské údaje s naší vytvořenou databází odcizených tabulek (tabulka *ibobr_hacknuta_hesla2*), museli jsme nejdřív zjistit z databáze emailovou adresu uživatele, podle uživatelského jména, protože tabulka *ibobr_hacknuta_hesla2* neobsahuje žádná uživatelská jména.

```
$db = JFactory::getDbo();
        $query = $db
            ->getQuery(true)
            ->select('email')
            ->from($db->quoteName('jos_users'))
            ->where($db->quoteName('username') . '=' .
                $db->quote($credentials['username']),);
        $db->setQuery($query);
```

```
$email = $db->loadResult();
```

Z tohoto databázového dotazu jsme vytáhli emailovou adresu uživatele podle jeho uživatelskéhé jména.

Po vyplnění přihlašovacího formuláře můžeme pracovat s emailovou adresou a heslem uživatele, což nám dovoluje porovnávat tyto uživatelské údaje s vytvořenou tabulkou *ibobr_hacknuta_hesla2*.

```
$db = JFactory::getDbo();
  $query = $db
      ->getQuery(true)
      ->select('*')
      ->from($db->quoteName('ibobr_hacknute_hesla2'))
      ->where($db->quoteName('email') . 'LIKE' .
              $db->quote($email), 'AND')
      ->andwhere($db->quoteName('heslo') . 'LIKE BINARY' .
                 $db->quote($credentials['password']));
  $db->setQuery($query);
  $rowId = $db->loadResult();
```

Tento databázový dotaz použije uživatelský email a heslo uživatele, které porovná se záznamy v tabulce *ibobr_hacknuta_hesla2*. Výstup tohoto dotazu je ID řádku tabulky *ibobr_hacknuta_hesla2*.

Kvůli menší chybě porovnání uživatelského hesla s heslem v tabulce *ibobr_hacknuta_hesla2* se našla shoda, když měl uživatel jiné heslo. Šlo o případ, kdy uživatel zadá své heslo s velkými písmeny a v databázi je uložené heslo s malými písmeny. Pro opravu této chyby jsme místo LIKE použili LIKE BINARY operátor, který dbá při porovávání znaků na velká a malá písmena.

K upozornění uživatele o jeho nebezpečných uživatelských údajích jsme použili výpis chybové hlášky. Chybová hláška se zobrazí pouze tehdy, když předchozí dotaz našel v databázi shodu.

```
if ($rowId !== null) {

        $db = JFactory::getDbo();
        $query = $db
            ->getQuery(true)
            ->select('id')
            ->from('jos_users')
            ->where($db->quoteName('email') . '=' . $db->quote($email),);
        $db->setQuery($query);
        $resultID = $db->loadResult();
        JFactory::getApplication()->enqueueMessage
        ('Vaše údaje nejsou bezpečné, doporučujeme si je okamžitě změnit.',
            'error');

        $MatchValue=$this->metrikaHesel($data['password']);
```

Pokud se našla shoda, provede se dotaz, který vyhledá ID hacknutého uživatele a vyhodnotí se síla hesla uživatele. Tyto uživatelské údaje se vkládají do databázové tabulky *ibobr_hacknnuti_uzivatele* spolu s nalezeným ID řádku, v *ibobr_hacknuta_hesla2* a s datumem nalezení shody, který se vloží automaticky při vytvoření nového záznamu.

```
try {
        $profile = new stdClass();
        $profile->id_user = $resultID;
        $profile->id_shoda=$rowId;
        $profile->metrika_stareho_hesla = $MatchValue;
        $resultQuery = JFactory::getDbo()->
                insertObject('ibobr_hacknnuti_uzivatele', $profile);
        } catch (\Exception $e) {
            }
```

V kódu si můžeme všimnout použití zachytávání výjimky pomocí try catch. Jedná se o řešení, které zamezuje vložení duplicitních záznamů do tabulky *ibobr_hacknnuti_uzivatele*. V tabulce je sloupec ID shoda (ID řádku v tabulce

ibobr_hacknuta_hesla2) nastaven jako unikátní. Když se pokusí vložit další stejný ID řádku tabulky *ibobr_hacknuta_hesla2*, vložení do tabulky hacknutí uživatelé se neprovede. Bez použití funkce try catch by pokus o vložení duplicitního záznamu skončil chybou webové stránky.

Tato funkce je vytvořená pro přehlednost. Kdyby se daný uživatel pořád přihlašoval do webové stránky Bobříka informatiky s nebezpečnými údaji, v tabulce *ibobr_hacknnuti_uzivatele* bude zaznamenám hacknutý uživatel pouze jednou.

Každému uživateli, který se přihlásí do webové stránky Bobříka infromatiky, je bodována síla hesla.

```
$db = JFactory::getDbo();
   $query = $db
       ->getQuery(true)
       ->select('id')
       ->from('jos_users')
       ->where($db->quoteName('email') . '=' . $db->quote($email),);
   $db->setQuery($query);
   $resultID2 = $db->loadResult();

   $ReturnedValue=$this->metrikaHesel($data['password']);
```

Tento způsob funguje na stejném principu jako při hodnocení hesla hacknutých uživatelů. U každého přihlášeného uživatele proběhne dotaz na jeho ID z tabulky uživatelů (*jos_users*). Následně se tyto dva záznamy vloží do tabulky *ibobr_analyza_hesel*.

```
    try {
            $insert = new stdClass();
            $insert->id_user = $resultID2;
            $insert->metrika_hesla = $ReturnedValue;
            $resultQuery2 = JFactory::getDbo()->
                            insertObject('ibobr_analyza_hesel', $insert);

        } catch (\Exception $e){

        }
```

Pro zamezení vkládání duplicitních záznamů do tabulky *ibobr_analyza_hesel* jsme použili stejný způsob jako při vkládání záznamů do tabulky *ibobr_hacknnuti_uzivatele*.

3.5.4 Implementace modulu pro změnu hesla

Abychom mohli updatovat správný řádek tabulky hacknutého uživatele, který si právě mění své nebezpečné heslo, museli jsme z databázové tabulky *ibobr_hacknnuti_uzivatele* získat správný řádek. Toho jsme docílili pomocí níže uvedeného dotazu.

```
$db = JFactory::getDbo();
    $query = $db
        ->getQuery(true)
        ->select('id')
        ->from($db->quoteName('ibobr_hacknnuti_uzivatele'))
        ->where($db->quoteName('metrika_noveho_hesla').'='. $db->quote(0),'AND')
        ->andwhere($db->quoteName('id_user').'='. $db->quote($userId));
    $db->setQuery($query);
    $selectedID = $db->loadResult();
```

Tento dotaz nám v tabulce *ibobr_hacknnuti_uzivatele* zajištujě výstup ID požadovaného záznamu.

Pro rozhodnotí, zda nově změněné heslo se vyskytuje v tabulce *ibobr_hacknuta_hesla2*, jsme použili další dotaz, který prohledá tabulku *ibobr_hacknuta_hesla2* a porovná nově zvolené heslo se sloupcem hesla.

```
$db = JFactory::getDbo();
    $query = $db
        ->getQuery(true)
        ->select('*')
        ->from($db->quoteName('ibobr_hacknute_hesla2'))
        ->where($db->quoteName('heslo').'LIKE BINARY'.
                $db->quote($data['password1']));
    $db->setQuery($query);
    $matchId = $db->loadResult();

    if($matchId !== NULL){
        $matchId = true;
    }else{
        $matchId = false;
    }
```

Znovu jsme využili operátor LIKE BINARY, aby výsledek výstupu id dbal na porovnávání velkých a malých písmen. Podmínka, zda heslo se vyskytuje v databázi (True nebo False), funguje jednoduše. Pokud se našlo id v tabulce, automaticky dostává hodnotu True, pokud ne, dostává hodnotu False.

Když si uživatel změní heslo, pro toto heslo se aplikuje stejný algoritmus metriky hesla, jako když se poprvé přihlásil do webové stránky Bobříka informatiky.

Pro vyhodnocení podmínky, jestli se vybraný řádek v tabulce *ibobr_hacknnuti_uzivatele* má updatovat, jsme použili dotaz, který zjistí, zda hledanému ID chybí přidělená nová metrika hesla, tedy pokud nová metrika hesla daného záznamu se rovná 0, do tabulky se vloží čas změny hesla, metrika nového hesla a informace o výskytu hesla v tabulce *ibobr_hacknuta_hesla2*.

```
$db = JFactory::getDbo();
    $query = $db
        ->getQuery(true)
        ->select('metrika_noveho_hesla')
        ->from($db->quoteName('ibobr_hacknnuti_uzivatele'))
        ->where($db->quoteName('id').'='. $db->quote($selectedID));
    $db->setQuery($query);
    $resultMetrics = $db->loadResult();

    if($resultMetrics==0){
        $profile = new stdClass();
        $profile->id=$selectedID;
        $profile->cas_zmeny_hesla=date("Y-m-d H:i:s");
        $profile->metrika_noveho_hesla=$result;
        $profile->nove_hacknute_heslo=$matchId;
        $updateResult = JFactory::getDbo()->
            updateObject('ibobr_hacknnuti_uzivatele', $profile,'id');
    }
```

4 Analýza hesel

Nasbírané záznamy přihlášených uživatelů do webové stránky Bobříka informatiky jsme použili k analyzování hesel uživatelů a analyzování nalezených shod uživatelských údajů s tabulkou odcizených údajů. Pro vyhodnocení analýzy hesel jsme si položili následující výzkumné otázky. Zavisí síla hesla uživatele na jeho pohlaví? Závisí síla hesla učitele na jeho typu školy, ve které vyučuje?

4.1 Metodika analýzy hesel podle pohlaví a škol

Pro porovnání síly hesel dle pohlaví byl použit Welchův t-test, který je vhodný pro porovnání dvou skupin kvantitativní proměnné. Pro testování závislosti síly hesel na typu školy byla použita analýza rozptylu (ANOVA), která je vhodná pro porovnání tří nebo více skupin kvantitativní proměnné.[35] Kromě výsledné p-hodnoty byly reportovány průměry a směrodatné odchylky a v rámci krabicového grafu medián, dolní a horní kvartil, minimum a maximum. Pro metriku síly hesel byl vytvořen sloupcový graf s absolutními a relativními četnostmi. Výpočty byly provedeny pomocí programu TIBCO STATISTICA, hladina významnosti činila 5 %.

4.2 Analýza hesel podle pohlaví

H_0 : Síla hesla nezávisí na pohlaví.

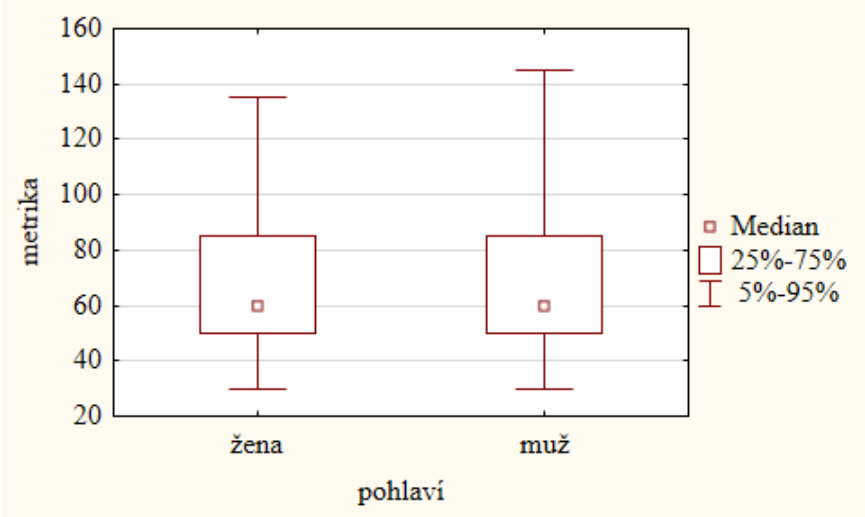
H_A : Síla hesla závisí na pohlaví.

Pro ověření závislosti síly hesel na pohlaví uživatele byla testována hypotéza H_0 (Síla hesla nezávisí na pohlaví) oproti alternativní hypotéze H_A (Síla hesla závisí na pohlaví).

Welchův t-test: p-hodnota a číselné charakteristiky

Pohlaví	počet	průměr	sm.odch.	p-hodnota
Muž	404	71,3	36.1	0,344
Žena	499	69,1	34,2	(nezamítame H_0)

Metrika síly hesla činila pro muže v průměru 71,3 při směrodatné odchylce 36,1 a pro ženy v průměru 69,1 při směrodatné odchylce 34,2. P-hodnota Welchova t-testu vyšla s ohledem na 3 desetinná místa 0,344, tj. vyšší než zvolená hladina významnosti 0,05. Nulová hypotéza nebyla zamítnuta. Na hladině významnosti 0,05 nebyla prokázána závislost síly hesla na pohlaví. Pořadové statistiky obou srovnávaných skupin byly zobrazeny pomocí kategorizovaného krabicového grafu.



Obrázek 10: Znázornění síly hesel mužů a žen, pomocí krabicového grafu

4.3 Analýza hesel podle typu školy

H_0 : Síla hesla nezávisí na typu školy.

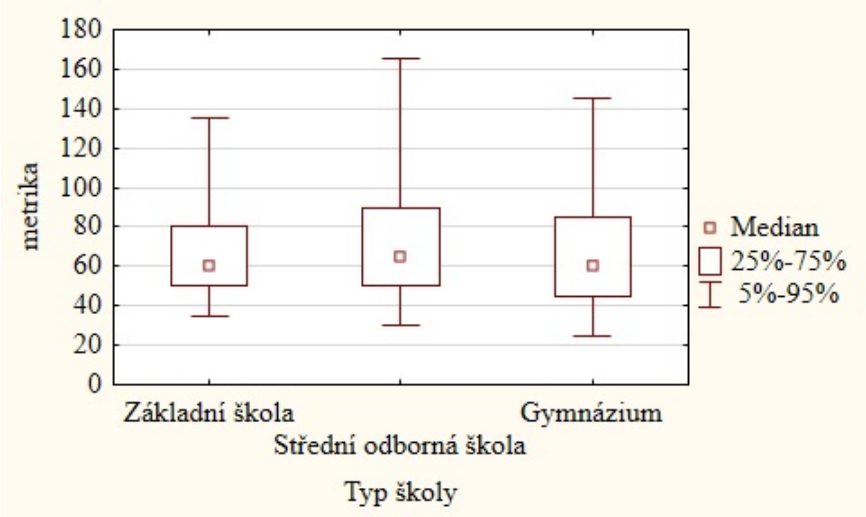
H_A : Síla hesla závisí na typu školy.

Pro ověření závislosti síly hesel učitelů podle typu školy, ve které vyučuje byla testována hypotéza H_0 (Síla hesla nezávisí na typu školy) oproti alternativní hypotéze H_A (Síla hesla závisí na typu školy).

Analýza rozptylu: p-hodnota a číselné charakteristiky

Typ školy	počet	průměr	sm.odch.	p-hodnota
Základní škola	635	70,0	33,4	0,381
Střední odborná škola	93	74,5	43,2	(nezamítame H_0)
Gymnázium	170	68,3	36,2	

Metrika síly hesla činila pro respondenty ze základní školy v průměru 70,0 při směrodatné odchylce 33,4, pro respondenty ze střední odborné školy v průměru 74,5 při směrodatné odchylce 43,2 a pro respondenty z gymnázia v průměru 68,3 při směrodatné odchylce 36,2. P- hodnota analýzy rozptylu vyšla s ohledem na 3 desetinná místa 0,381, tj. vyšší než zvolená hladina významnosti 0,05. Nulová hypotéza nebyla zamítnuta. Na hladině významnosti 0,05 nebyla prokázána závislost síly hesla na typu školy. Pořadové statistiky srovnávaných skupin byly zobrazeny pomocí kategorizovaného krabicového grafu.



Obrázek 11: Znázornění síly hesel, která používají učitelé vyučující na základní škole, středí odobrné škole a gymnáziu, pomocí krabicového grafu

4.4 Četnost výskytu podobně silného hesla

Zajímalo nás rozložení síly hesel uživatelů Bobříka informatiky, z tohoto důvodu jsme vytvořili skupiny 20 až 40 bodů, 41 až 60 bodů, 61 až 80 bodů, 101 až 120 bodů a vetší než 120 bodů. V rozmezí mezi 20 až 40 body se nachází velmi slabá hesla. Pod touto škálou si můžeme představit hesla jako „kočka“, „pes123“ nebo „heslo“.

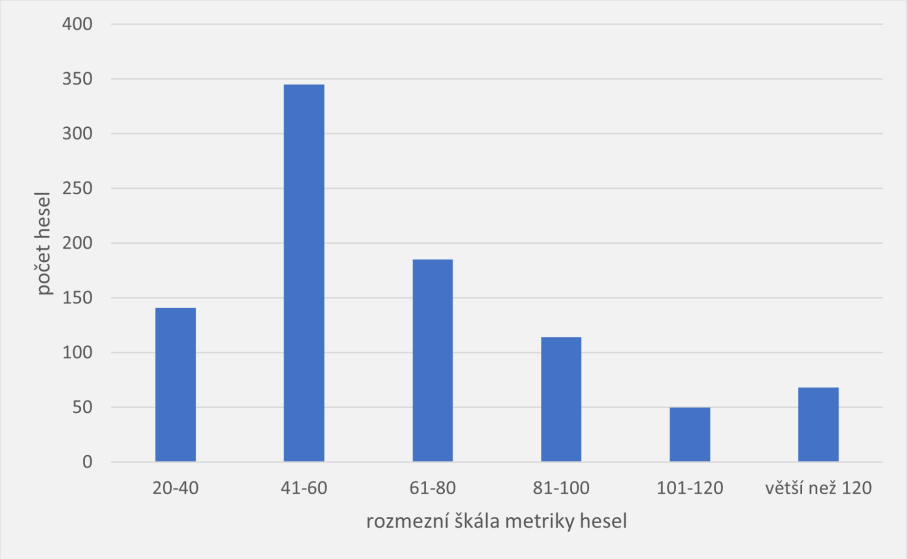
V rozmezí mezi 41 až 60 body se nachází hesla jako „Petr1234“ nebo „petrpetr1234“.

V rozmezí mezi 61 až 80 body jsou hesla jako „Mojeheslo123“ nebo „hesloheslo12345“. Tato hesla bývají pořád slabá, obvykle bez speciálního znaku, jsou jen o pár znaků delší než hesla v rozmezí 41 až 60 body.

V rozmezí mezi 81 až 100 bodů hesla už většinou obsahují velké písmeno nebo speciální znak. Pod touto škálou si můžeme představit hesla jako „P€tr1234“, „P!t3r123“, „PetrSamec1234“.

V rozmezí mezi 101 až 120 bodů můžou být hesla zajímavě poskládána. Heslo se 110 body může vypadat jako „P!t33rON“ nebo „PetrSamec1234567“.

V rozmezí větší než 120 bodů se nacházejí hesla jako „PetrSamec123456!“ (140 bodů) nebo „P€t3Rs@m3C“ (180 bodů).



Obrázek 12: Znázornění četnosti hesel uživatelů podle jejich síly.

Z grafu je na první pohled vidět, že největší výskyt podobně silného hesla je v rozmezí 41 až 60 bodů s četností téměř 350 hesel. Druhá největší četnost se nachází mezi škálou 61 až 80 bodů s téměř 200 hesly. Jako třetí nejpoužívanější hesla uživatelů na webové stránce Bobříka informatiky jsou hesla v rozmezí 20 až 40 bodů s četností téměř 150 hesel. Čtvrtá největší skupina v rozmezí 81 až 100 bodů obsahuje lehce přes 100 hesel. V rozmezí škály větší než 120 bylo zaznamenáno téměř 70 hesel. Ve skupině v rozmezí 101 až 120 bodů bylo zaznamenáno pouze 50 hesel.

4.5 Analýza zjištěných odcizených údajů

V databázi zjištěných odcizených údajů bylo zaznamenáno pouze 9 uživatelů, u kterých se shodovala emailová adresa a heslo s tabulkou *ibobr_hacknuta_hesla2*. Pouze 4 uživatelé přistoupili na změnu hesla.

První uživatel, u kterého byla nalezena shoda s databází, byl zaznamenán dne 4. 11. 2022 v 7:56 s metrikou hesla 40 bodů. Uživatel změnil své nebezpečné heslo až 17 dní po zjištění shody, konkrétně dne 21. 11. 2022 v 16:37. Nové heslo mělo metriku 65 bodů, což bylo o 25 bodů silnější než původní. Nově

zvolené heslo se neshodovalo s žádným záznamem v *ibobr_hacknuta_hesla2* .

Druhý uživatel byl zaznamenán dne 4. 11. 2022 v 8:28 s metrikou hesla 50 bodů. Navzdory doporučení změnit heslo tak neučinil.

Třetí uživatel byl zaznamenán dne 6. 11. 2022 v 13:14 s metrikou hesla 30 bodů a také na doporučení změny hesla nereagoval.

Čtvrtý uživatel byl zaznamenán dne 7. 11. 2022 v 7:49 s metrikou hesla 50 bodů. Na doporučení změny hesla reagoval okamžitě a své heslo změnil během 4 minut od oznámení nebezpečnosti údajů. Nově zvolené heslo dosáhlo hodnoty 120 bodů a heslo se neshodovalo s žádným záznamem v tabulce *ibobr_hacknuta_hesla2*. Rozdíl mezi starým a novým heslem byl 60 bodů, což ukazuje, že tento uživatel přistoupil k změně hesla zodpovědně.

Pátý uživatel byl zaznamenán dne 7. 11. 2022 v 13:00 s metrikou hesla 40 bodů. Změna hesla mu trvala pouze 1 minutu, ale metrika nového hesla se zvýšila pouze o 5 bodů. Toto heslo se neshodovalo s žádným záznamem v tabulce *ibobr_hacknuta_hesla2*.

Šestý uživatel byl zaznamenán dne 7. 11. 2022 v 18:16 s metrikou hesla 50 bodů. Stejně jako předchozí uživatelé ani tento na doporučení změny hesla nereagoval.

Sedmý uživatel byl zaznamenán 11. 11. 2022 v 10:34 s metrikou hesla 35 bodů. Změnu hesla provedl za pouhou 1 minutu. Nové heslo dosáhlo 55 bodů, což značí rozdíl 20 bodů oproti původnímu heslu. Nové heslo se neshodovalo s žádným záznamem v tabulce *ibobr_hacknuta_hesla2*.

Osmý uživatel byl zaznamenán dne 8. 11. 2022 v 11:12 s metrikou hesla 45 bodů a také na doporučení změny hesla nereagoval.

Devátý uživatel byl zaznamenán 14. 11. 2022 v 19:54 s metrikou hesla 30 bodů. Navzdory doporučení k změně hesla nepřistoupil.

uživatel	změnil	čas do změny hesla	síla starého heslo	síla nového hesla	rozdíl v síle hesel	nové heslo v tabulce odcizených údajů
1	ano	17 dnů	40	65	25	ne
2	ne	x	x	x	x	x
3	ne	x	x	x	x	x
4	ano	4 minuty	50	120	70	ne
5	ano	1 minuta	40	45	5	ne
6	ne	x	x	x	x	x
7	ano	1 minuta	35	55	20	ne
8	ne	x	x	x	x	x
9	ne	x	x	x	x	x

Obrázek 13: Přehledová tabulka zjištěných odcizených údajů.

5 Závěr

Jedním z cílů této práce bylo ukázat běžnému uživateli internetu, jak snadno lze hesla odcizit nebo prolomit, pokud nejsou dostatečně silná.

V praktické části práce byla přiblížena implementace modulu pro sbírání dat o bezpečnosti a síle hesla uživatelů systému Bobřík informatiky. Během průběhu školního kola soutěže Bobříka informatiky, kdy sbírala data indikující dřívější odcizení přihlašovacích údajů, se nepodařilo nashromáždit dostatek záznamů pro konstruktivní analýzu. Proto byl vytvořen dílčí cíl práce - analýza hesla učitelů registrovaných v systému Bobříka informatiky. Zde vznikly dvě výzkumné otázky týkající se závislosti síly hesla na pohlaví a na typu škole učitele, ve které vyučuje.

Z výsledků bylo zjištěno, že síla hesla není ovlivněna pohlavím uživatele ani typem školy učitele. Polovina testovaných uživatelů Bobříka informatiky používá slabá hesla, jedna čtvrtina používá obstojná hesla a druhá čtvrtina používá relativně silná hesla. Během sbírání záznamů bylo nalezeno devět uživatelů, jejichž hesla spadala do kategorie slabá a byla shodná se záznamy v tabulce odcizených údajů. Těmto uživatelům bylo zobrazeno upozornění na nastalou situaci, avšak pouze čtyři z těchto uživatelů přistoupili na změnu hesla.

Hacknuté databáze uživatelských údajů se dají využít pro zabezpečení uživatelských účtů. Obtížnost tohoto vylepšení spočívá v nalezení správných záznamů v hacknutých databázích, aby byly vhodné pro danou skupinu uživatelů a nedošlo zbytečně k zahlcení databáze.

Vytvořený modul lze aktualizovat importováním dalších hacknutých záznamů, což může zlepšit jeho efektivitu. Databáze neobsahuje emailové adresy na jiné doméně než .cz a .sk. Bylo by dobré přidat další záznamy, ve kterých bude emailová adresa s doménou .com. To by však způsobilo enormní nárůst velikosti tabulky, a tak by bylo nutné optimalizovat tabulku odcizených údajů pomocí indexování tabulky.

Seznam použité literatury a zdrojů

[1] MILER, Matt. Password Cracking 101: Attacks Defenses Explained. *BeyondTrust* [online]. 2022 [cit. 2023-02-24]. Dostupné z:https://www.beyondtrust.com/blog/entry/password-cracking-101-attacks-defenses-explained

[2] TECHTARGET. Dictionary attack. *TechTarget* [online]. 2021 [cit. 2023-02-24]. Dostupné z: https://www.techtarget.com/searchsecurity/definition/dictionary-attack#:~:text=A%20dictionary%20attack%20is%20a,an%20encrypted%20message%20or%20document..

[3] ALEXANDER. 135 - Password Cracking Techniques *PUNISHER* [online]. 2012 [cit. 2023-03-05]. Dostupné z: https://github.com/purplebyteone/Mega-Hacking-E-Books-Collection/blob/master/135%20-%20Password%20Cracking%20Techniques%20%5B-PUNISHER-%5D.pdf

[4] BALBIX. State of Password Use Report 2020. *Balbix* [online]. [cit. 2023-03-05]. Dostupné z: https://www.balbix.com/resources/state-of-password-use-report-2020/

[5] SHANKDHAR, Pavitra. Popular tools for brute-force attacks [updated for 2020]. *Infosec* [online]. 2020 [cit. 2023-02-24]. Dostupné z:https://resources.infosecinstitute.com/topic/popular-tools-for-brute-force-attacks/

[6] TECHTARGET. Ethical hacking tools and techniques: Password cracking. *TechTarget* [online]. 2007 [cit. 2023-02-24]. Dostupné z: https://www.techtarget.com/searchitchannel/feature/Ethical-hacking-tools-and-techniques-Password-cracking#:~:text=Brutus%20is%20a%20password%20cracking,%2C%20FTP%2C%20SMB%20and%20Telnet.

[7] CLOUDFLARE. What is a brute force attack?. *Cloudflare* [online]. 2023 [cit. 2023-02-24]. Dostupné z: https://www.cloudflare.com/learning/bots/brute-force-attack/

[8] NESKEY, Corey. Are Your Passwords in the Green? *Hive Systems* [online]. 2023 [cit. 2023-03-13]. Dostupné z: https://www.hivesystems.io/blog/are-your-passwords-in-the-green

[9] CLOUDFLARE. What is credential stuffing? | Credential stuffing vs. brute force attacks. *Cloudflare* [online]. 2023 [cit. 2023-02-24]. Dostupné z: https://www.cloudflare.com/learning/bots/what-is-credential-stuffing/#:~:text=Insights-,What%20is%20Credential%20Stuffing%3F,in%20to%20another%20unrelated%20service.

[10] POZA, Diego. What Is Password Spraying? How to Stop Password Spraying Attacks. *Auth0* [online]. 2021 [cit. 2023-02-24]. Dostupné z:https://auth0.com/blog/what-is-password-spraying-how-to-stop-password-spraying-attacks/

[11] HARSHMASTER07705. Reverse Brute Force Attack in System Hacking. *GeeksforGeeks* [online]. 2022 [cit. 2023-02-24]. Dostupné z:https://www.geeksforgeeks.org/reverse-brute-force-attack-in-system-hacking/

[12] SEVENLAYERS. HASHCAT MASK ATTACK. *SEVENLAYERS* [online]. 2023 [cit. 2023-02-24]. Dostupné z: https://www.sevenlayers.com/index.php/287-hashcat-mask-attack

[13] CALIFORNIA STATE UNIVERSITY BAKERSFIELD. RevsUp Lab: Hashcat 06. *California state university Bakersfield* [online]. [cit. 2023-02-24]. Dostupné z: https://www.cs.csub.edu/~melissa/revs-up/sum2018/polo/hashcat06.html

[14] ARMOURINFOSEC. Performing Rule Based Attack Using Hashcat. *Armour Infosec* [online]. 2022 [cit. 2023-02-24]. Dostupné z: `https://www.armourinfosec.com/performing-rule-based-attack-using-hashcat/`

[15] PROOFPOINT. What Is Phishing?. *Proofpoint* [online]. 2023 [cit. 2023-02-24]. Dostupné z: `https://www.proofpoint.com/us/threat-reference/phishing`

[16] THOMPSON, Katarina. What Is Phishing?. *Tripwire* [online]. 2023 [cit. 2023-02-24]. Dostupné z: `https://www.tripwire.com/state-of-security/6-common-phishing-attacks-and-how-to-protect-against-them`

[17] HELIXSTORM. 12 Types of phishing attacks to watch out for. *Helixstorm* [online]. 2023 [cit. 2023-02-24]. Dostupné z: `https://www.helixstorm.com/blog/x-types-of-phishing-attacks-to-watch-out-for/`

[18] KASPERSKY. What is Smishing and How to Defend Against it. *Kaspersky* [online]. 2023 [cit. 2023-02-24]. Dostupné z: `https://www.kaspersky.com/resource-center/threats/what-is-smishing-and-how-to-defend-against-it`

[19] ESET. Vishing. *Eset* [online]. 2023 [cit. 2023-02-24]. Dostupné z: `https://www.eset.com/cz/vishing/`

[20] MORAMARCO, Stephen. Link manipulation. *Infosec* [online]. 2016 [cit. 2023-02-24]. Dostupné z: `https://resources.infosecinstitute.com/topic/link-manipulation/`

[21] LUTKEVICH, Ben, Casey CLARK a Sharon SHEA. Whaling attack (whaling phishing). *TechTarget* [online]. 2021 [cit. 2023-02-24]. Dostupné z: `https://www.techtarget.com/searchsecurity/definition/whaling`

[22] SUB154. Content Spoofing. *GeeksforGeeks* [online]. 2018 [cit. 2023-02-24]. Dostupné z: https://www.geeksforgeeks.org/content-spoofing/

[23] KASPERSKY. Evil twin attacks and how to prevent them. *Kaspersky* [online]. 2023 [cit. 2023-02-24]. Dostupné z: https://www.kaspersky.com/resource-center/preemptive-safety/evil-twin-attacks

[24] KASPERSKY. Pharming meaning and definition. *Kaspersky* [online]. 2023 [cit. 2023-02-25]. Dostupné z: https://www.kaspersky.com/resource-center/definitions/pharming

[25] EASYDMARC. What is Angler Phishing and How Can You Avoid It?. *EasyDMARC* [online]. 2022 [cit. 2023-02-25]. Dostupné z: https://easydmarc.com/blog/what-is-angler-phishing-and-how-can-you-avoid-it/

[26] WRIGHT, Gavin a Madelyn BACON. Watering hole attack. *TechTarget* [online]. 2021 [cit. 2023-02-25]. Dostupné z: https://www.techtarget.com/searchsecurity/definition/watering-hole-attack

[27] HANNA, Katie Terrell. Shoulder surfing. *TechTarget* [online]. 2021 [cit. 2023-02-26]. Dostupné z: https://www.techtarget.com/searchsecurity/definition/shoulder-surfing

[28] GUARDIO. Clickbait / Clickjacking. *Guardio* [online]. 2023 [cit. 2023-02-26]. Dostupné z: https://guard.io/dictionary/clickbait-clickjacking

[29] MORANO, Jason. Pass the hash (PtH) attacks: How they work – and how to defend against them. *Quest* [online]. 2022 [cit. 2023-03-04]. Dostupné z: https://blog.quest.com/pass-the-hash-pth-attacks-how-they-work-and-how-to-defend-against-them/

[30] RZOMERMAN. PASS THE HASH. *Azureinfra* [online]. 2015 [cit. 2023-03-13]. Dostupné z: `https://blog.azureinfra.com/2015/02/24/pass-the-hash/`

[31] KULIUKAS, Kestas. How Rainbow Tables work. *Kestas Kuliukas* [online]. 2006 [cit. 2023-04-01]. Dostupné z: `https://kestas.kuliukas.com/RainbowTables/`

[32] SHEASBY, Ryan. Rainbow Tables (probably) aren't what you think — Part 1: Precomputed Hash Chains.*Medium* [online]. 2021 [cit. 2023-03-05]. Dostupné z: `https://rsheasby.medium.com/rainbow-tables-probably-arent-what-you-think-30f8a61ba6a5`

[33] LONG, Joshua. Collection 1 (and 2–5) are the latest massive password dumps. *Intego*[online]. 2019 [cit. 2023-03-13]. Dostupné z: `https://www.intego.com/mac-security-blog/collection-1-and-2-5-are-the-latest-massive-password-dumps/`

[34] JOOMLA. About Joomla! *Joomla* [online]. 2023 [cit. 2023-04-11]. Dostupné z: `https://www.joomla.org/about-joomla.html`

[35] BEVANS, Rebbeca. Choosing the Right Statistical Test | Types Examples. *Scribbr* [online]. 2022 [cit. 2023-05-13]. Dostupné z: `https://www.scribbr.com/statistics/statistical-tests/`