Západočeská univerzita v Plzni

FAKULTA PEDAGOGICKÁ

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

POČÍTAČOVÁ HESLA

BAKALÁŘSKÁ PRÁCE

Tomáš Bozděch

Přírodovědná studia, obor Informatika se zaměřením na vzdělání

(2009 - 2012)

Vedoucí práce: *Ing. Petr Michalík, Ph.D.*

Plzeň, 1. leden 2012

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

Plzeň, 1. leden 2012

……
vlastnoruční podpis

Rád bych poděkoval vedoucímu bakalářské práce Ing. Petru Michalíkovi, Ph.D. za jeho rady, opravy a připomínky, které mě inspirovaly a pomohly vytvořit tuto práci.

OBSAH

1 ÚVOD

Samotná počítačová hesla vznikala na počátku počítačové éry. Počítače jako takové se šířily obrovskou rychlostí. V počátcích se jednalo o kusy sloužící armádě. Tyto počítače zabíraly velké plochy kvůli svým rozměrům, byly primitivní a měly pouze pár možností na využití. V dnešní době je běžné, že jedna domácnost má více počítačů, které jsou již mnohonásobně výkonnější a zvládají složitější a rozmanité programy.

Většina počítačových systémů a některé programy jsou chráněny hesly. Dnes každý uživatel čelí mnoha útokům, proto je dobré svá důvěrná data chránit pomocí silných hesel, která se snaží prolomení zamezit, nebo alespoň zkomplikovat.

Tato práce je rozdělena do teoretické a praktické části.

V teoretické části jsou objasněny se základními pojmy, které souvisí s problematikou počítačových hesel, jako jsou bezpečnost, autentizace, autorizace, PIN kód, elektronický podpis, útoky na heslo a správa hesel. Dále jsou uvedeny doporučení, jak vytvořit silné heslo, kde všude se dá heslo nastavit pro zvýšení bezpečnosti, jaké máme možnosti prokázání své totožnosti, kde zjistit, jestli naše heslo je opravdu bezpečné, a kde je možno generovat hesla.

Praktická část je zaměřena na vlastní aplikaci Testovač. Tato aplikace hodnotí zadaná hesla podle zvolených kritérií a také dokáže vygenerovat silné heslo.

2 ZÁKLADNÍ POJMY SOUVISEJÍCÍ S POČÍTAČOVÝMI HESLY

2.1 DEFINICE

Heslo lze definovat jako posloupnost po sobě následujících znaků sloužících k zabezpečení soukromých dat. Za soukromá data můžeme považovat: osobní údaje, přístup do počítače, přístup k různým webovým portálům či sociálním sítím, přístup k serverům atd.

2.2 VÝSKYT HESEL V POČÍTAČI

Již při samotném startu osobního počítače je možno narazit na heslo pro BIOS. Tato hesla nejsou moc využívána u počítačů či notebooků v domácnosti. U většiny služebních notebooků jsou pak nedílnou součástí bezpečnosti. BIOS heslo totiž nepustí uživatele bez znalosti hesla ani do nastavení BIOSU, pokud nepoužijí reset BIOSU či jinou metodu odstraňující heslo BIOSU. Tudíž při odcizení nemůže zloděj tak snadno nastavit bootování z optické mechaniky či USB a přeformátovat operační systém, aby mohl počítač využívat dle svých potřeb bez znalosti hesla pro operační systém. [1]

Nyní jsme se dostali k dalšímu heslu, a tím je heslo k samotnému operačnímu systému. Toto heslo je využíváno především pro počítače, na kterých pracuje více uživatelů. Určitě každý uživatel chce zachovat své soukromí, tudíž udělá základní krok a nastaví si své heslo pro svůj účet. Uživatelské účty neslouží jen k uchování soukromí, ale dozajista každému uživateli vyhovuje i jiné pracovní prostředí.

V pracovním prostředí hned narazíme na spousty dalších hesel. V dnešní době každý zná společnost Microsoft a její sadu Microsoft Office. Ta nabízí možnost zamčení dokumentu, kdy je pomocí šifrování zaručena větší bezpečnost obsahu dokumentu. V programu Excel, z již jmenované sady od Microsoftu, se nabízí hned 3 možnosti zabezpečení. A to buď zamčení celého sešitu nebo jakéhokoliv listu v sešitě, či zamčení jednotlivých buněk.

S dalšími hesly se můžeme setkat v programu od této firmy, který se jmenuje Microsoft Internet Explorer. V tomto programu si může uživatel hesla ukládat. To uživateli usnadňuje a urychluje práci a dobu trávenou zadáváním hesel na různých webových stránkách, kterými mohou být různé sociální sítě, e-mailové servery, portály, atd. Tato

hesla se však nevztahují jen na prohlížeč od Microsoftu, ale také na všechny ostatní webové prohlížeče.

V e-mailovém klientu Microsoft Outlook se musí také vyplnit heslo k zadané e-mailové adrese. Samozřejmě i tento e-mailový klient není jediný klient na světě, který lze používat. Řada jiných aplikací také používá hesla k zabezpečení různých činností. Například simulační program Multisim, se kterým jsme se setkávali dosti často během studia, používá možnost zamčení návrhu zapojení heslem, stejně tak zadání chyby součástek, nastavení pracovní plochy apod.

2.3 BIOMETRICKÁ AUTENTIZACE

2.3.1 AUTENTIZACE

Systém potřebuje přesně vědět, s kým komunikuje, proto využívá tzv. autentizaci. Autentizace slouží k prokázání jednoznačné identity uživatele. U většiny bezpečných systémů dochází k autentizaci uživatele pomocí vnitřních systémů, nejčastěji databázového serveru. Autentizaci vyžaduje také např. školní Orion, kde se pomocí svého vytvořeného jména a hesla můžeme hlásit k systému. K autentizaci také lze využít speciální aplikace, kterými jsou například bezpečnostní nebo adresářové servery, čipové karty či služby operačního systému. [2]

2.3.2 HLAS

Jednou z možných biometrických možností zabezpečení systému je uzamknutí pomocí hlasu. S touto autentizací se v dnešní době stále setkáváme spíše ojediněle. Velkou bezpečnostní výhodou, kterou lze využít s hlasovým uzamčením, je tzv. vícefaktorová autentizace, kde lze kombinovat více typů autorizací. Na rozpoznání hlasu jsou podle [3] využívány dvě metody, a to speaker recognition a speech recognition. První zmiňovaný systém slouží k rozpoznání, komu hlas patří. Druhý systém rozpoznává, co uživatel říká.

Samotné uložení hlasu může proběhnout bez vědomí uživatele. Příkladem může být monitorovaný hovor od operátora. Ten může hlas využít se souhlasem uživatele k otisku hlasu (voiceprint). Hlas se musí nejprve analyzovat, až poté dochází k vytvoření otisku. Již vytvořený otisk hlasu uložíme do systému, kde jej chceme využívat. Poté při přihlášení do systému dochází k ověření otisku pomocí speciálního algoritmu, který

porovnává hlasový otisk aktuálním namluveným záznamem uživatele, ten může být buď úspěšně vpuštěn do systému, nebo naopak mu může být odepřen přístup.

Nebezpečným místem tohoto zabezpečení se stává nastavení neměnného hlasového hesla, které může způsobit to, že uživatelovo hlasové zadávání hesla může být nahráváno útočníkem. Tento typ útoku se nazývá opakovaný útok (replay attack). Snadno se tomuto lze vyhnout tím, že nastavíme zadávání hesla jako opakování fráze, kterou nám systém nadiktuje. Nejčastěji to bývá náhodné opakování číslic nebo frází. Člověk nedokáže přesně zopakovat slovo, aniž by to systém nerozpoznal. Proto i neměnnému heslu můžeme zvýšit bezpečnost tak, že nastavíme zadávání hesla dvakrát po sobě. Systém totiž pak rozpozná, zda se jedná o zopakovanou frázi uživatele nebo o nahrávku. Člověk totiž nedokáže i při sebevětší snaze vyslovit dvakrát to samé heslo stejným způsobem. Vždy se bude o nějakou hlasovou složku lišit, kdežto přehrávané heslo bude pořád stejné i po několikátém přehrání. [3]

2.3.3 OTISK PRSTU

Za heslo můžeme považovat i otisky prstů, je to určitý typ autentizace. Jak již každý z nás ví, co prst, to jiný otisk. Tohoto se využívá především v kriminalistice. Na světě neexistují dva lidé se stejným otiskem, nepočítáme-li podvodníky, co dokážou vyrobit přesnou kopii jakéhokoliv prstu. Dnes již dávno není novinkou, že se na noteboocích přidává čtečka otisku prstů, která slouží pro přihlášení do systému, aby uživatel nemusel zadávat heslo na klávesnici. Nevýhodou je, že tyto především levné čtečky nedokáží přesně zmapovat prst nebo při přihlašování prst jednoduše nerozpoznají. I přesto, že se jedná o majitelův originální prst, který přejede přes čidlo konstantní rychlostí, dochází občas k tomu, že heslo ve formě otisku prstu není přijato systémem. Větší problém nastává, když se prst špatně zmapuje a uživatel si nenastaví jinou možnost přihlášení do systému pomocí jiného prstu či použití hesla z klávesnice.

2.3.4 BAREVNÝ OTISK PRSTU

Modernějším využitím otisku prstu je barevný otisk prstu. Tento systém využívá barevnou škálu odpovídající tlaku, který působí na detektor. Tato metoda je mnohem přesnější, tudíž následná identifikace uživatele je jednoznačnější. Tato metoda oproti normálnímu otisku prstu má tu výhodu, že zachycuje vlastnosti kůže, hloubku linie a tvar

prstu. Vědci tvrdí, že prolomení zabezpečení pomocí této metody je téměř nemožné. S těmito detektory se můžeme setkat např. na letištích. [4]

2.3.5 OBLIČEJ

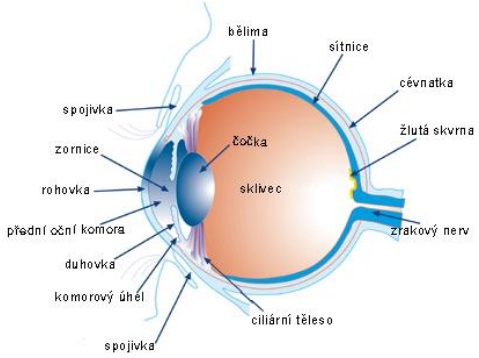
Autentizaci na základě obličeje můžeme rozdělit na dvě techniky. První technikou je apearence-based, která udělá příznakový vektor na základě obrazové informace z jasu pixelů. Poté tento vektor podstoupí nějakou z metod pro snížení dimenze. Tato metoda je pak méně kvalitní u obrazů s nižším rozlišením, kde nemusí dojít k jednoznačnému a rychlému rozpoznání, např. nedokáže reagovat na určitou pózu. Druhá technika, model-based, vychází z významných rysů, které se nacházejí na lidské tváři (uši, oči, atd.). Druhá technika je bezpečnější, protože vyžaduje kvalitní obraz. Za nevýhodu tohoto typu autentizace můžeme považovat lidské emoce, které mění významné rysy obličeje. [4]

2.3.6 GEOMETRIE RUKY

Autentizace pomocí geometrie ruky u nás není moc rozšířená. U nás je používán spíše otisk prstu. Tento typ autentizace se zabývá celkovou geometrií ruky, kam spadá zakřivení prstů, žíly na hřbetu ruky aj. Samotný proces snímání zjišťuje hlavní rysy ruky, čímž jsou významné body na obvodu rukou, šířka a délka prstů či dlaně, tvar dlaně, atd. Jeden z hlavních důvodů, proč není autentizace pomocí geometrie rukou rozšířená, je, že v průběhu dospívání se ruka často mění, změnou velikosti se posouvají významné body, a proto je znemožněno rozpoznání uživatele. Dalším důvodem, proč není uživatel rozpoznán, může být například změna váhy (mění se šířka prstů a dlaně). [4]

2.3.7 DUHOVKA

Pomocí duhovky dochází k jednomu z nejbezpečnějších identifikací uživatele, neboť duhovka je chráněna čočkou a rohovkou (Obrázek 1).



Obrázek 1: Lidské oko převzato z http://lidske-smysly.wbs.cz/zrak/dfvgbnm.jpg

Duhovka je dobře chráněna, tudíž nemůže dojít k poškození jako u otisku prstu (popálení, říznutí). Změna duhovky nebývá během lidského života nikterak výrazná.

Tato metoda je využívaná na některých letištích například v Holandsku, USA a Kanadě. Důvodem proč je tato spolehlivá metoda autentizace tak málo rozšířena, jsou vysoké pořizovací náklady zařízení na snímání duhovky. Za druhou nevýhodu můžeme považovat nutnost blízkého kontaktu oka se zařízením. Další nevýhodou může být možnost útoku výrobou speciálních čoček, které nahradí lidskou duhovku. [4]

2.3.8 SÍTNICE

Identifikace pomocí sítnice je taktéž velmi přesnou metodou prokázání totožnosti uživatele. Tuto metodu využívá FBI, NASA, CIA pro identifikaci uživatele, který chce získat oprávnění pro vstup do nějaké části budovy. Tato metoda autentizace kontroluje sítnici ve 320 obrazových bodech, díky těmto bodům je každá sítnice jedinečná. [4]

2.3.9 DNA

Metoda autentizace pomocí DNA[1] je prozatím ve fázi vývoje, neboť nejsou vynalezeny snímače ani nic podobného, co by umělo odebrat vzorek DNA bez kontaktu s uživatelem. Odběr vzorku DNA se používá například v kriminalistice či při určování biologického otce po porodu. [4]

2.3.10 EEG[2] KŘIVKA

Autor této metody testoval 70 jedinců, kterým vysílal podněty do mozku a ten u každého jedince vykazoval jiné výsledky. Metoda je založena na akci a reakci. Předpokládá se další vývoj a testování této metody. [4]

2.3.11 ANALÝZA PSANÍ NA KLÁVESNICI

Analýza psaní na klávesnici nezkoumá jen, jaké klávesy jsou stisknuty, ale především zkoumá to, jakou intenzitou jsou drženy a jak dlouho jsou drženy. Velkou výhodou této metody jsou úplně nulové náklady na pořízení hardwaru. [4]

[1] DNA - biologická láska, která nese informace o živém organismu
[2] EEG – elektroencefalograf, zobrazuje průběh reakcí mozku na elektrické vstupní podněty

2.3.12 CHŮZE

Tento typ autentizace je prozatím v intenzivní fázi vývoje. Od tohoto systému se očekává velké využití v kriminalistice, díky němuž by měla být snadnější identifikace pachatele.

Každý člověk má svůj typ chůze, který je pro něj charakteristický. Typ chůze určuje dynamika, délka nohou, délka kroku, zakřivení nohou, vady chůze aj. [4]

2.4 AUTORIZACE

Na autentizaci navazuje ve většině případů autorizace. Autorizace je potvrzení přístupu uživatele do systému. Úkolem autorizace je také zjistit, zda má daný uživatel dostatečná oprávnění na vykonání příslušné akce. [2]

2.5 PIN KÓD

V překladu tato zkratka znamená osobní identifikační číslo. Jedná se o typ hesla využívaného především u mobilních zařízení, dále je možno se s ním setkat u platebních karet, vstupních kódů atd. PIN slouží jako autorizační požadavek zařízení. Tento typ hesla má omezený počet pokusů (zpravidla tři). Po třech špatných pokusech následuje zadání hesla PUK, což je zkratkou anglického výrazu Personal Unlocking Key. To můžeme přeložit do češtiny jako osobní odblokovací klíč. [10]

2.6 ELEKTRONICKÝ PODPIS

Elektronický podpis se v České republice objevil v roce 2000 se zákonem nesoucím název Zákon o elektronickém podpisu. Elektronický podpis slouží jako vlastnoruční podpis s tím rozdílem, že není třeba psací potřeby, ale pouze speciálního pera a zařízení, které snímá pohyb pera. [5]

2.7 ÚTOKY NA HESLA

Každý uživatel počítače ví, že na světě se nachází spousta lidí, kteří se snaží prolomením hesla zviditelnit, získat informace, data, peníze nebo například někomu změnit obsah webových stránek, atd. Jedná se o podvodné jednání, které se v praxi bohužel vyskytuje. V následujících kapitolách budou podrobněji popsány některé útoky na hesla. [6]

2.7.1 ÚTOK HRUBOU SILOU

Jednou z metod prolomení hesla je útok hrubou silou. Tento útok využívá všechny možné kombinace znaků, kterými se snaží prolomit heslo. Tento způsob útoku má teoreticky a za určitých podmínek 100% úspěšnost. Stoprocentně úspěšný by útok byl, kdybychom na něj měli neomezenou dobu a heslo se za tu dobu nezměnilo. Tudíž u velmi silných hesel by odhalení trvalo až statisíce let. Doba potřebná k prolomení hesla je zobrazena v tabulce v kapitole 3.1.1. [6]

2.7.2 SLOVNÍKOVÝ ÚTOK

Tato metoda útoku využívá slovník, ze kterého používá slova, a zkouší je dosadit jako heslo. Jedná se o velmi úspěšnou metodu praktikovanou na uživatelích, kteří používají jednoduchá hesla, jejichž pravděpodobnost výskytu ve slovníku je vysoká. [6]

2.7.3 ÚTOK PŘENESENÍM

Téměř každý uživatel, zejména neprofesionál, využívá stejné heslo pro různé programy, přihlášení, aplikace aj. Metoda útoku přenášení hesla je nebezpečná především pro správce sítě nebo jinak vysoce postavené lidi či uživatele starající se o bezpečnost a funkčnost systémů. Po odhalení prvního hesla totiž útočník okamžitě získá přístup ke všemu s týmž heslem. [6]

2.8 SPRÁVCE HESEL

Správce hesel funguje nejen jako evidence hesel v počítači, ale také pinů, kódů, atd. Jedná se o elegantnější způsob pro zapamatování hesel, které v počítači uživatel používá, než využívat notýsek s vypsanými hesly, uložený někde v zamčeném šuplíku či trezoru.

Správce hesel je vhodný převážně pro uživatele, kteří používají různorodá hesla a mohlo by tak dojít k zapomenutí hesla uživatelem. Velkou výhodou disponují programy pro správu hesel ve snadném dohledání hesla na uzamčené místo, které se právě chystají používat. Uživatelům, kteří používají jedno stejné heslo na vše, je tento program zcela k ničemu. Velkou nevýhodou správce hesel je možnost „prolomení“ programu a získání dat z něj. Člověk během zlomku vteřinky může přijít o citlivé informace, které můžou být pro něj důležité.

Většina správců hesel se dá zdarma stáhnout z internetu, je však důležité pečlivě zvážit, zda se jedná o bezpečný program pro správu hesel nebo pouze o snadno prolomitelný program. Na internetu můžeme najít hodně správců hesel, zde jsou některé z nich: KeePass, RoboForm (pouze 10 hesel zdarma, poté se musí přejít na placenou verzi), Password Manager XP, Password Pro, Handy Password, IntelliLogin, Správce hesel aj. Jako ukázku jednoho ze správců hesel jsem zvolil poslední zmiňovaný Správce hesel.

2.8.1 SPRÁVCE HESEL 1.73

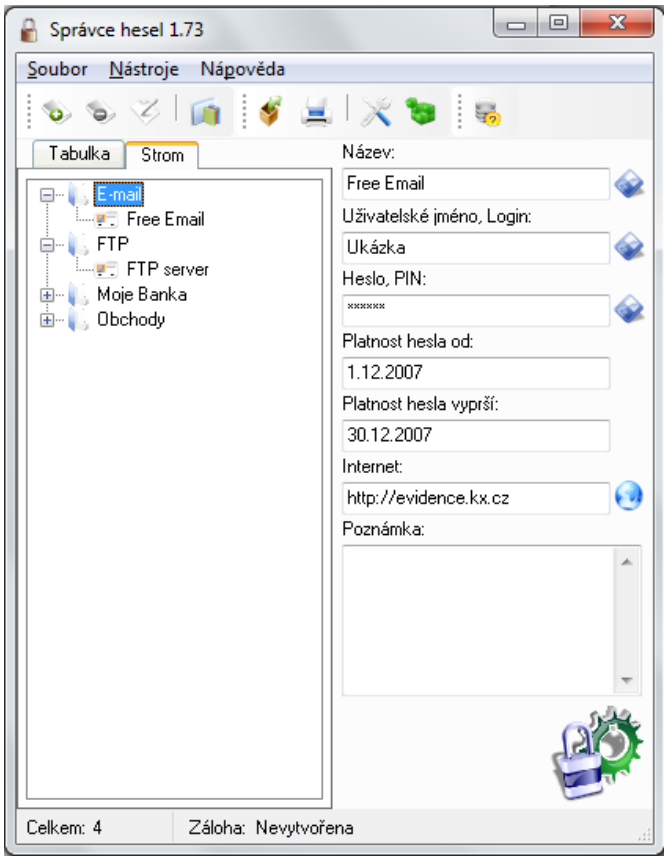
Program Správce hesel nefunguje pouze jako správce hesel, ale nabízí také možnost generování hesel. Samotná správa hesel funguje tak, že uživatel si může vybrat buď ze 4 předvolených kategorií e-mail, FTP, moje banka, obchody nebo si může vytvořit novou kategorii. Předvolené kategorie lze editovat či odstranit. Tudíž uživatel může mít vytvořené pouze své kategorie. To je lepší pro přehlednost.

Program zobrazuje kategorie ve stromu. Zobrazení lze ze stromu změnit na tabulku, kde se vypíší všechny záznamy jednotlivých kapitol. I záznamy v jednotlivých kapitolách může uživatel dle potřeby editovat, mazat a vytvářet. Při vytváření záznamu uživatel vybírá, do které kapitoly bude daný záznam přiřazen. Pokud kapitolu nezvolí, a daný záznam vytvoří, tak se nově vytvořený záznam nezobrazí ve stromu. Pro zobrazení záznamu bez kategorie musí uživatel v zobrazení zvolit tabulku.

Program dále nabízí možnost exportu. Zde uživatel zadává cestu k souboru, ze kterého bude export proveden. Další možnost programu je volba přímého tisku, kde uživatel bude vyzván k výběru výstupu tisku. Zde se nachází dvě možnosti tisku, a to tisk seznamu kategorií nebo tisknout seznam hesel.

Program nabízí i své nastavení, ovšem toto nastavení je značně omezeno. V nastavení lze měnit pouze dva prvky. Jedním je zobrazení, tudíž uživatel si může nastavit, zda se mu bude po spuštění či po přidání nějakého záznamu zobrazovat strom nebo tabulka. Druhou možností nastavení je možnost vyčistit schránku po ukončení programu. Nyní se pojďme podrobněji podívat na vytvoření nového záznamu. Po kliknutí na ikonu přidat nový záznam na panelu snadného spuštění nebo při výběru soubor – přidat nový záznam se nám zobrazí formulář pro vyplnění nového záznamu. Zde si můžeme vybrat z rozbalovacího menu, do jaké kategorie tento záznam přidáme. Dále

uživatel má možnost vyplnit pět polí, která se jmenují název, uživatelské jméno, login, heslo, PIN, internet a poznámka. Dále si uživatel může ve formuláři zvolit platnost hesla. Ukázku programu si můžete prohlédnout na obrázku (Obrázek 2).



Obrázek 2: Screenshot programu Správce hesel 1.73

3 ZÁSADY PRO TVORBU KVALITNÍHO HESLA

3.1 BEZPEČNOST

Bezpečnost hesla určuje jednak kombinace různorodých znaků, dostatečná délka hesla a také perioda, za kterou je potřeba heslo změnit. Jak tedy vypadá dostatečně bezpečné heslo? Prvotní je dostatečná délka hesla. Délka hesla by měla být více než 8 znaků. Heslo by mimo velká a malá písmena mělo obsahovat také číslice a speciální znaky. Shodnost hesla s osobními údaji či loginem je velice nevhodná.

Jak by tedy mohlo vypadat takové bezpečné heslo v praxi? Vezměme si například nějaké nám známé přísloví, které obsahuje více než 8 slovních spojení. Příkladem může být přísloví: „**Ž**ádná **p**íseň **n**ení **t**ak **d**louhá**, a**by **jí n**ebylo **k**once**.**“ Nabízí se nám hned několik možností na vytvoření bezpečného hesla. Z každého slova může být využito první písmeno a může být buď ponecháno, nebo vyměněno za číslo či speciální znak. Pro lepší názornost jsem si dovolil toto zvýraznit v textu a ve výsledném heslu jsem zaměnil písmeno j za číslici 1. Vzniklé heslo má tuto finální podobu: „**Žpntd,a1nk.**.“ Těchto záměn by mohlo pro větší bezpečnost vzniknout více. Další problematika související s bezpečností hesla je shodnost všech vlastních hesel. Nyní se pojďme podrobněji podívat do jednotlivých problematik, které jsou podrobněji vysvětleny v následujících kapitolách. [7]

3.1.1 ODLIŠNOST ZNAKŮ

Každý z nás si pod pojmem odlišnost znaků může představit něco jiného. U hesel se nejedná o nic jiného, než užití různých abeced v hesle. V následující tabulce je vidět doba potřebná k prolomení hesla v závislosti na užitých znacích. Na tuto tabulku by se měli podívat především amatéři, kteří tuto problematiku neznají a používají jednoduchá hesla. Tabulka (Tabulka 1) byla celá převzata z webových stránek. [8]

3.1.2 DÉLKA HESLA

Jak již bylo ukázáno v tabulce, délka hesla je velmi důležitou součástí bezpečného hesla. V tabulce lze vidět s přibývajícími znaky exponenciální nárůst doby potřebné k prolomení hesla. Toto však neplatí u slovníkových útoků, když si uživatel zvolí heslo, které bude přímo ve slovníku, bude prolomeno během několika vteřin. Taktéž heslo tvořeno posloupností opakujících se znaků o jakékoliv délce je snadno prolomitelné.

Tabulka 1: Čas potřebný k prolomení hesla při rychlosti 100 hesel za sekundu

Znaky Počet / Užité	4	5	6	7	8
0-9	2 minuty	16 minut	3 hodiny	1 den	11 dní
a-z, 0-9	5 hodin	7 dní	8 měsíců	25 let	900 let
a-z, A-Z, 0-9	2 dny	3 měsíce	18 let	1 000 let	70 000 let
a-z, A-Z, 0-9, speiální znaky	6 dní	1 rok	120 let	10 000 let	800 000 let

3.1.3 DOBA ZMĚNY HESLA

Další bezpečnostní požadavek pro heslo je perioda určená pro změnu hesla. Určitě každý vysokoškolský student je s touto změnou hesla seznamován každý semestr. Jedná se o bezpečnostní prvek, který po uživateli žádá zadání nového hesla. Když uživatel nestihne heslo zavčasu změnit, je nucen jít za pověřenou osobou, která má tuto záležitost na starost a pomůže mu pomocí jednorázového hesla umožnit přístup k zablokovanému účtu. Toto jednorázové heslo si pak uživatel musí neprodleně změnit.

3.1.4 SHODNOST HESLA S LOGINEM ČI OSOBNÍMI ÚDAJI

Heslo shodné s loginem či osobními údaji je krajně nevhodné pro zabezpečení soukromých dat. Toto heslo je snadno odhalitelné, ať už od kolegů či kolegyň v práci či ve škole nebo i od útočníků nám neznámých. Dále je nevhodné používat hesla typu administrátor, admin apod. Všem známá připojená čísla za naše heslo 123, 123456, 654321, 321 apod. nejsou také nic bezpečného pro uchování osobních údajů.

3.1.5 SHODNOST VŠECH VLASTNÍCH HESEL

Velký problém nastane pro uživatele, když útočník odhalí heslo a uživatel využívá heslo pro všechny své účty a loginy. Během zlomku vteřiny tak může uživatel být „okraden“ o své citlivé údaje, peníze a nejen to. Útočník může stáhnout data, smazat či je snadno změnit, a to určitě pro každého uživatele není nic příjemného.

3.2 GENERÁTOR HESEL

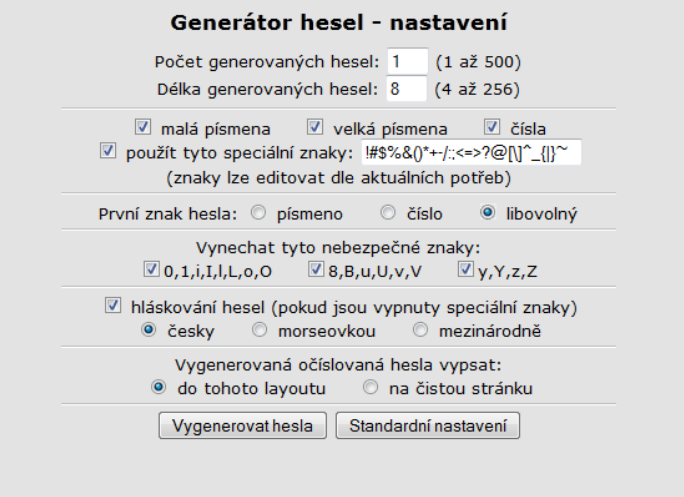
Generátor hesel plní funkci náhodného zobrazení po sobě následujících znaků. Tyto znaky jsou náhodně losovány ze zvolené abecedy. Převážná většina generátorů vám

nabídne široké spektrum možností. Možnostmi je myšleno například zvolení abecedy, ze které se výsledné heslo bude generovat, kolik znaků se má z dané abecedy vygenerovat a také kolik generovaných hesel se má zobrazit. Nevýhodou generátorů je malá pravděpodobnost zapamatování hesla uživatelem. To nese další rizika pro uživatele generátoru, kterými může být ukládání vygenerovaného hesla do souboru nebo vypsání generovaných hesel na papír. Tato rizika se vztahují k oběma následujícím podkapitolám. Někteří uživatelé dokonce dělají takovou chybu, že si lepí hesla na monitor.

3.2.1 ONLINE GENERÁTORY HESEL

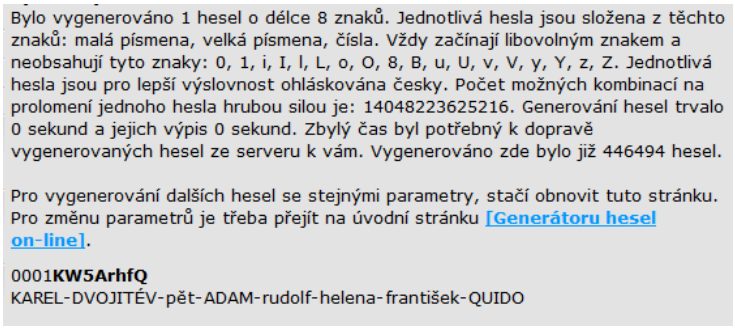
V dnešní době se nachází na internetu široká řada generátorů. Ovšem zdaleka ne všechny generátory nabízí možnost použití speciálních znaků, což může být pro bezpečnost hesla velký problém. Odhalení hesla tak může trvat podstatně kratší dobu, než při generování a následného užití hesla se speciálními znaky. Pro názornou ukázku jsem zvolil online generátor hesel nacházející se na adrese http://www.hsgi.cz/generator-hesel/.

Tento generátor byl vytvořen uživatelem, který preferuje velkou bezpečnost hesla, proto nabízí široké spektrum možností pro uživatele. V prvním poli si může uživatel nastavit, kolik hesel má být vygenerováno. Toto pole nijak neovlivňuje kvalitu hesla, pouze určuje množství vygenerovaných hesel, tudíž jestliže potřebujeme jediné heslo, stačí zadat do pole číslo 1 a vygeneruje se jedno jediné heslo. Pokud si však uživatel chce vybrat nějaké, které by se mu co nejsnáze pamatovalo, tak může využít všech 500 možných generovaných hesel a z bohaté nabídky si vybrat pro něj to nejvhodnější. Druhé pole s názvem délka generovaných hesel může být v rozmezí od 4 do 256. Jak již z názvu pole je vidět, jedná se o celkový počet znaků, které budou generovaná hesla obsahovat. Další možností generátoru jsou čtyři zaškrtávací políčka (Check Boxes), kde si uživatel volí, jaké abecedy má heslo využívat. V nabídce tohoto generátoru jsou velká písmena, malá písmena, číslice a speciální znaky, které si uživatel může zvolit dle libosti. Tyto čtyři volby může uživatel volit dle svého uvážení či potřeby v libovolné kombinaci. Další částí formuláře jsou tři rádiová tlačítka, z nichž je možno zvolit typ počátečního znaku hesla. V dalších krocích uživatel může zvolit vynechání nebezpečných znaků, hláskování hesel a jak chce generovaná hesla zobrazit. Ukázku okna tohoto generátoru si můžete prohlédnout na obrázku (Obrázek 3).



Obrázek 3: Screenshot z online generátoru hesel http://www.hsgi.cz/generator-hesel/

Nyní se pojďme podívat, jak generátor funguje při generování malých písmen, velkých písmen a číslic. Při generování znaků z těchto abeced je využíváno náhodného losování slov a číslic. Slova jsou v tomto případě jména z kalendáře, výjimkou je pouze w, které ve slovníku reprezentuje slovo „dvojitév“. U jmen generovaných z kalendáře se dále rozhoduje o počátečním písmenu, zda bude velké nebo malé. Jednotlivými znaky vygenerovaného hesla jsou počáteční písmena generovaných jmen ze slovníku. Je-li generována číslice, generuje se pod psaným názvem čísla. Například generátor vygeneruje slovo šest a v hesle se objeví pod klasickou číslicí 6. Jak takovéto generování vypadá, si můžete prohlédnout na obrázku 3. Dole za číslem 0001 bylo vygenerováno celé heslo a o řádek níž je sled slov, které tvoří svými počátečními písmeny generované heslo (Obrázek 4). [9]

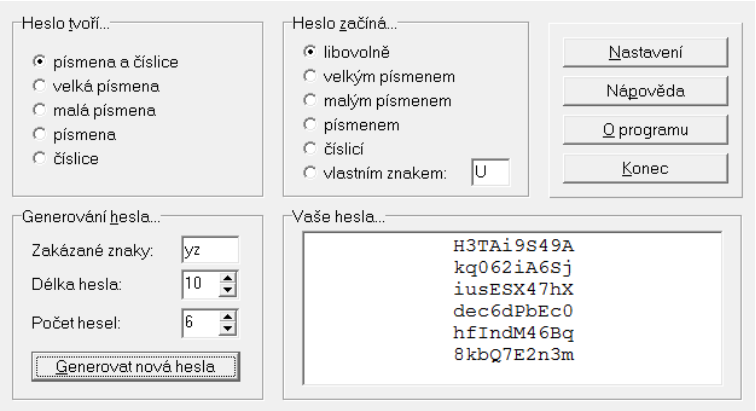


Obrázek 4: Screenshot vygenerované hesla v online generátoru hesel http://www.hsgi.cz/generator-hesel/

3.2.2 PROGRAMY PRO GENEROVÁNÍ HESLA

Rozdíl mezi online generátory hesel a softwarem není téměř žádný, nepočítáme-li to, že musíme daný software stáhnout, rozbalit či nainstalovat a spustit. Toto je jedna z velkých nevýhod oproti online generátorům. Další nevýhodou pak může být zjištění, že software nemusí nabízet některé funkce jako například použití speciálních znaků do generovaného hesla. Pro ukázku jsem zvolil program nesoucí název Tvorba hesel verze 1.0, kterou jsem stáhl z www.slunecnice.cz.

Tento software nabízí volby ve třech polích. V prvním poli programu uživatel volí, jakou abecedu bude využívat generátor. K dispozici má uživatel možnost těchto voleb: písmena a číslice, velká písmena, malá písmena, písmena, číslice. Pod samostatným názvem písmena jsou myšlena velká i malá písmena. V druhém poli se nachází možnost volit, čím bude generované heslo začínat. V posledním poli lze nastavovat hned tři možnosti. Jsou to zakázané znaky, délka hesla a počet generovaných hesel. Ve čtvrtém poli se zobrazí výsledná generovaná hesla vygenerovaná programem. Jak samotný program vypadá, si lze prohlédnout na obrázku (Obrázek 5).



Obrázek 5: Screenshot z programu Tvorba hesel 1.0

4 DOSTUPNÉ PROGRAMY PRO HODNOCENÍ KVALITY HESEL

4.1 ONLINE TESTOVAČE HESEL

Jestliže si uživatel není jist kvalitou bezpečnosti svého hesla, může si ji ověřit na řadě webových testovačů. Většina registračních formulářů na internetu má také svůj testovací algoritmus. Pro přiblížení těchto „testovačů“ jsem zvolil dva online testovače z adres http://www.passwordmeter.com/ a http://www.testyourpassword.com/. Druhý zmiňovaný testovač současně funguje jako generátor. Kombinace generátoru a testovače na sobě není nijak závislá.

4.1.1 PASSWORD METER

Ukažme si, jak funguje testovač z adresy http://www.passwordmeter.com/. Uživatel zadá své heslo pro otestování do pole password. Password meter nabízí možnost zaškrtávacího políčka hide, tudíž když je toto políčko zaškrtnuté, tak se místo jednotlivých znaků objevují zástupné symboly *. Naopak když políčko zaškrtnuto není, zobrazují se znaky tak, jak jsou zadány na klávesnici. Dále pak pod již zmíněným políčkem najdeme score (hodnocení) a complexity (složitost).

Napravo od zadávaného hesla se nachází minimální požadavek na heslo. Tento požadavek se skládá ze dvou bodů. Prvním je minimální délka hesla 8 znaků a druhým požadavkem je to, že dané heslo musí obsahovat tři ze čtyř známých abeced velká písmena, malá písmena, číslice a speciální symboly. Pod polem s názvem Složitost se nachází další dva podformuláře, ze kterých se počítá výsledná hodnota hodnocení udávaná v procentech.

První podformulář poukazuje na kladnou stránku hesla (Additions) v po sobě jdoucích bodech Number of Characters (počet znaků), Uppercase Letters (velká písmena), Lowercase Letters (malá písmena), Numbers (čísla), Symbols (speciální symboly), Middle Numbers or Symbols (prostřední znaky hesla tvoří čísla nebo speciální symboly), Requirements (požadavky).

Druhý podformulář poukazuje na zápornou stránku hesla (Deductions) v následujících údajích: Letters Only(pouze písmena), Numbers Only (pouze čísla), Repeat Characters - Case Insensitive (opakování znaků – malá a velká písmena), Consecutive

Uppercase Letters (velká písmena po sobě jdoucí), Consecutive Lowercase Letters (malá písmena po sobě jdoucí), Consecutive Numbers (čísla po sobě jdoucí), Sequential Letters (3+), Sequential Numbers (3+), Sequential Symbols (3+). Každý z těchto zmíněných bodů má u sebe napsanou škálu (Rate), podle kterého se počítá výsledné hodnocení.

Dále se na této stránce nachází legenda. Legenda se sestává ze čtyř úrovní, které se objevují vlevo v každém bodu z polí ať pro kladné hodnocení nebo záporné. Tyto úrovně jsou seřazeny od nejlepší po nejhorší. Výjimečná úroveň překračuje požadavky na minimální kvalitu hesla. Dostatečná úroveň splňuje pouze minimální požadavky. Varovací úroveň upozorňuje na negativní vliv hodnocení hesla. Selhávající úroveň upozorňuje uživatele v kladné části formuláře na to, že nebyly použity abecedy či minimální požadavky na heslo. Uživatel má logicky na začátku bez zadaného hesla v záporném hodnocení dostatečnou úroveň a v kladné části selhávající úroveň, protože nejsou splněny žádné z minimálních požadavků na kvalitní heslo. Jak samotný online program vypadá, si můžeme prohlédnout na obrázku (Obrázek 6).

Test Your Password		Minimum Requirements
Password:		• Minimum 8 characters in length • Contains 3/4 of the following items: - Uppercase Letters - Lowercase Letters - Numbers - Symbols
Hide:	☑	
Score:	0%	
Complexity:	Too Short	

Additions		Type	Rate	Count	Bonus
✗	Number of Characters	Flat	+(n*4)	0	0
✗	Uppercase Letters	Cond/Incr	+((len-n)*2)	0	0
✗	Lowercase Letters	Cond/Incr	+((len-n)*2)	0	0
✗	Numbers	Cond	+(n*4)	0	0
✗	Symbols	Flat	+(n*6)	0	0
✗	Middle Numbers or Symbols	Flat	+(n*2)	0	0
✗	Requirements	Flat	+(n*2)	0	0
Deductions					
✓	Letters Only	Flat	-n	0	0
✓	Numbers Only	Flat	-n	0	0
✓	Repeat Characters (Case Insensitive)	Comp	-	0	0
✓	Consecutive Uppercase Letters	Flat	-(n*2)	0	0
✓	Consecutive Lowercase Letters	Flat	-(n*2)	0	0
✓	Consecutive Numbers	Flat	-(n*2)	0	0
✓	Sequential Letters (3+)	Flat	-(n*3)	0	0
✓	Sequential Numbers (3+)	Flat	-(n*3)	0	0
✓	Sequential Symbols (3+)	Flat	-(n*3)	0	0

Legend

Exceptional: Exceeds minimum standards. Additional bonuses are applied.
Sufficient: Meets minimum standards. Additional bonuses are applied.
Warning: Advisory against employing bad practices. Overall score is reduced.
Failure: Does not meet the minimum standards. Overall score is reduced.

Obrázek 6: Screenshot testovače kvality hesel http://www.passwordmeter.com/

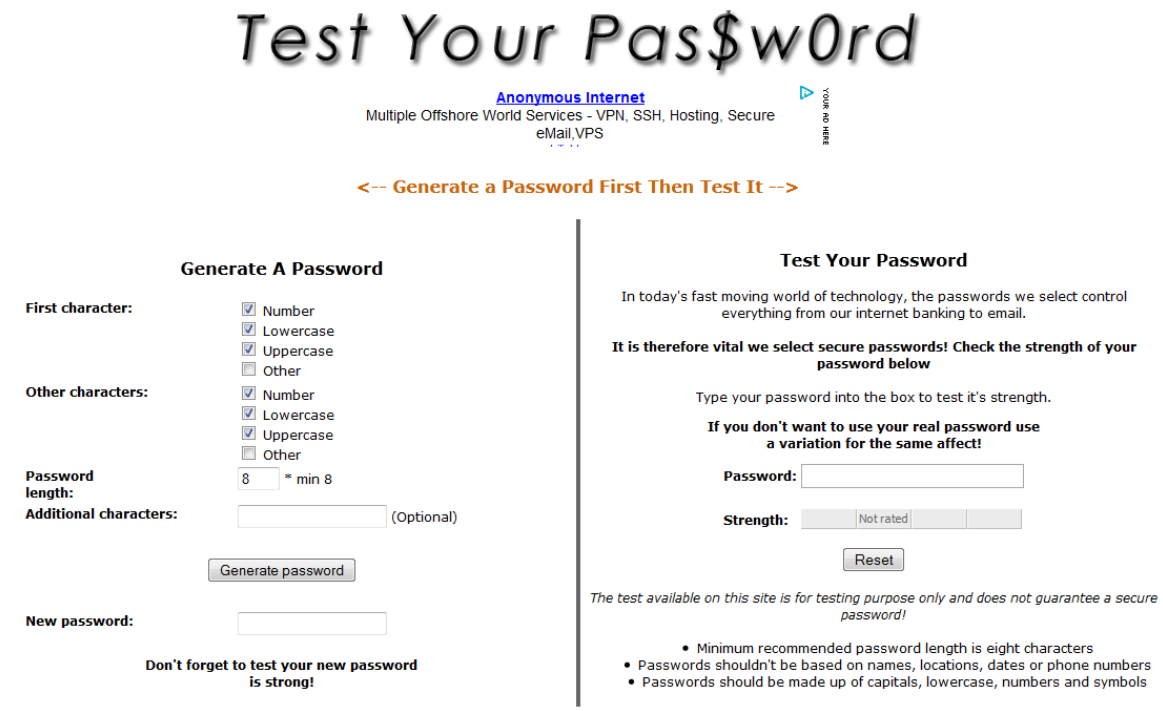
4.1.2 TEST YOUR PASSWORD

Jak je již z názvu patrné, jedná se o další online testovač kvality hesla, který se nachází na adrese http://www.testyourpassword.com/. Avšak tato webová stránka nenabízí pouze online testování zadaného hesla, ale také možnost generování hesla, tudíž uživatel se může hned ujistit, že dané vygenerované heslo je opravdu bezpečné. Majitel tohoto webu udává minimální požadavky na heslo ve třech bodech.

Prvním bodem je minimální délka hesla, což se neliší od jiných autorů, tudíž je udáno 8 znaků. Druhý bod zahrnuje varování před používáním jmen, míst, dat nebo telefonních čísel. Posledním bodem je využití velký a malých písmen, číslic a speciálních symbolů.

Generátor zde nabízí volby prvního znaku, a dalších znaků, které může heslo obsahovat, minimální délku hesla, pole, kam lze napsat symboly, které chceme použít, tlačítko pro generování hesla a pole, kde se zobrazuje výsledné generované heslo.

Nyní přejděme k samotnému testovači. Testovač nabízí čtyři úrovně kvality. Když není zadán ani jeden znak z hesla, testovač vypisuje not rated, což se dá přeložit jako nehodnoceno. Po zadání jakéhokoliv prvního znaku do pole s heslem vypíše testovač hlášení weak (slabé). Druhou úrovní je pak medium (střední), většinou se jedná o heslo obsahující pouze velká a malá písmena, malá písmena s číslicemi, velká písmena s číslicemi. Nutno podotknout, že u této úrovně zabezpečení jsou zadávaná hesla kratší než 8 znaků. Třetí a zároveň předposlední úrovní je úroveň nesoucí název strong (silná). Do této úrovně spadají hesla obsahující více jak 8 znaků s využitím velký a malých písmen, číslic i speciálních znaků. V porovnání s předchozím online testovačem mé heslo obsahující 11 znaků, z čehož tři jsou speciální znaky, jedno velké písmeno, pět malých písmen, a dvě číslice, které získalo 100% bodů, zde zapadá do třetí úrovně zabezpečení ze čtyř možných. Poslední, čtvrtou úrovní je úroveň nesoucí název best (nejlepší). Tato úroveň musí mít heslo delší než 13 znaků a musí nutně obsahovat speciální znaky a číslice. Písmena, ať už malá nebo velká, zde nehrají takovou velkou roli. Na obrázku si můžete prohlédnout, jak vypadá tato webová aplikace pro generování a testování hesla (Obrázek 7).



Obrázek 7: Screenshot z online testovače a generátoru hesel http://www.testyourpassword.com/

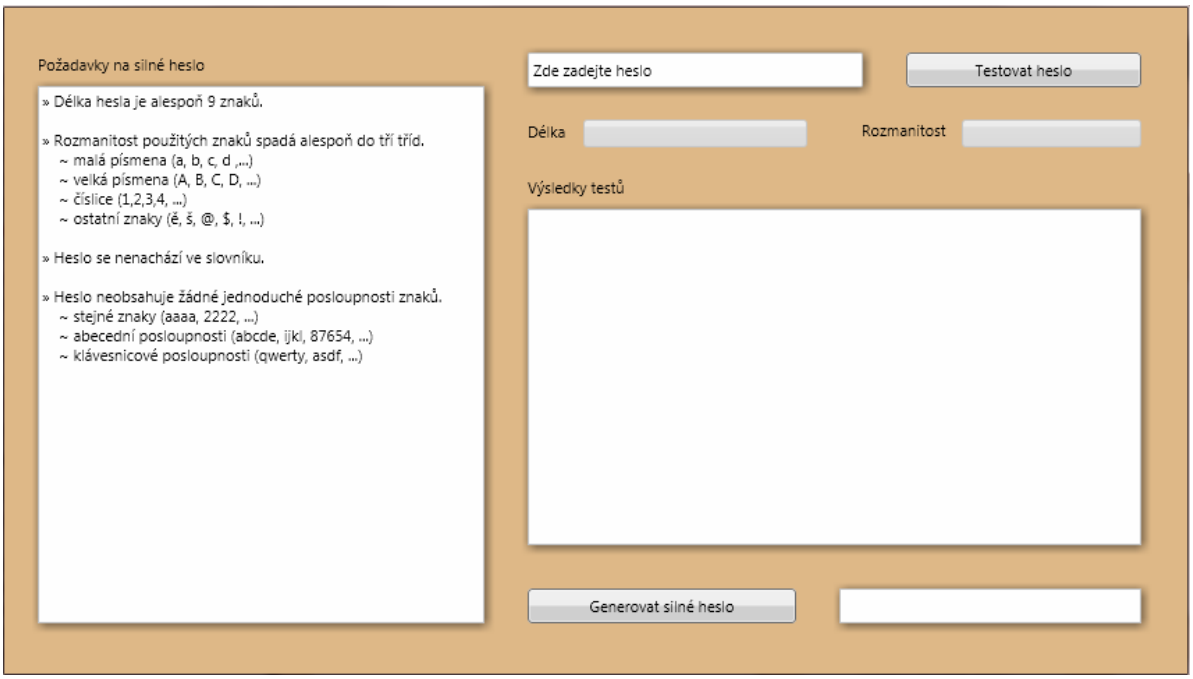
Tyto dva testovače se liší především v přesnosti hodnocení zadaného hesla. Testovač „passwordmeter“ udává hodnocení v procentech, tudíž víme přesněji, jakého výsledku dosáhlo testované heslo. Testovač „testyoupassword“ má větší nároky na nejbezpečnější hesla, která musí splňovat všechny požadavky na kvalitní heslo a musí být delší než 11 znaků.

Pro srovnání „testovačů“ passwordmeter a testyourpassword jsem zvolil následující hesla: H4fíKo5é, Matrix“1900“, !D!*!0!*!m!*!O!*!v!, 20. Ledna 1900, jenommalapismena, JENOMVELKAPISMENA, 8912122012. Heslo H4fíKo5é získalo v prvním zmiňovaném testovači 92%, druhý testovač dané heslo označil jako „silné“. Ani u jednoho z těchto „testovačů“ nezískal 100% hodnocení, protože heslo neobsahuje speciální znak. Například heslo „H4fíKo5é*“ by dosáhlo u prvního testovače 100% hodnocení. U druhého by bylo stále „silné“, protože na hodnocení „nejlepší“ obsahuje málo znaků. Dalším testovaným heslem je heslo „Matrix“1900““, které u prvního testovače dosáhlo 100% hodnocení a pro druhý testovač splňuje kritéria na silné heslo, pro hodnocení nejlepší obsahuje málo znaků. Heslo „!D!*!0!*!m!*!O!*!v!“ uspělo v obou „testovačích“ na výbornou a splnilo všechna kritéria pro kvalitní heslo. Následující heslo „20. Ledna 1900“ utvořené z data splnilo také všechna kritéria a dosáhlo u obou

„testovačů“ 100% výsledku. Heslo tvořené pouze malými písmeny „jenommalapismena“ v testech neobstálo, protože nesplňuje hned několik kritérií. V prvním „testovači“ dosáhlo 17% a ve druhém hodnocení „slabé“. Heslo tvořené pouze velkými znaky „JENOMVELKAPISMENA“ si v testech nevedlo o moc lépe, v prvním testovači získalo hodnocení 18% a ve druhém testovači na tom bylo stejně jako heslo tvořené pouze z malých písmen, tudíž výsledné hodnocení bylo „slabé“. Posledním testovaným heslem bylo heslo tvořeno pouze číslicemi „8912122012“. Délka hesla je shodná s délkou rodného čísla bez lomítka. Na online testovači „passwordmeter“ získalo toto heslo hodnocení 21% a na online testovači „testyoupassword“ bylo vyhodnoceno jako slabé heslo.

5 VLASTNÍ APLIKACE PRO VYHODNOCOVÁNÍ KVALITY HESLA

Pro vypracování vlastního programu jsem zvolil programovací jazyk delphi, kde jsem uplatnil znalosti získané během studia. Název programu je Testovač. Program obsahuje dvě na sobě nezávislé funkce. Jednou z nich je testování zadaného hesla a druhou funkcí je generování silného hesla. Desing programu si lze prohlédnout na následujícím obrázku (Obrázek 8).



Obrázek 8: Screenshot z vytvořeného programu Testovač

5.1 VZHLED

Jako barvu pozadí jsem zvolil sytě krémovou barvu, která je příjemná pro oči a neměla by tolik zatěžovat zrak.

V levé části obrazovky jsou vypsány požadavky na silné heslo, kterými jsou: délka hesla alespoň 9 znaků, rozmanitost použitých znaků, která musí zahrnovat alespoň 3 ze 4 abeced (malá písmena, velká písmena, číslice, ostatní znaky), heslo se nenachází ve slovníku (slovník je použitý z adresy http://scrabble.hrejsi.cz/pravidla/blex.htm). Dalším požadavkem na silné heslo je, že dané heslo nesmí obsahovat jednoduché posloupnosti opakujících se znaků nebo po sobě následujících znaků, ať už v abecedách či klávesových posloupností.

Do pole vyskytujícího se uprostřed nahoře se zadává heslo, které chce uživatel testovat. V tomto poli je zobrazen text: Zde zadejte heslo. Pokud do něj uživatel klikne,

text automaticky zmizí a uživatel může zadat heslo k testování. Tlačítko nacházející se v pravém horním rohu slouží k celkovému vyhodnocení zadaného hesla.

Dále se zde nachází dva ukazatelé délka a rozmanitost. Délka zobrazuje první požadavek na silné heslo, tudíž délka hesla musí být větší než 9. Sto procent z ukazatele lze dosáhnout pouze s heslem obsahující deset a více znaků. Další ukazatel, rozmanitost, zobrazuje počet použitých abeced pro zadané heslo. Stejně tak jako délka hesla, i rozmanitost se mění v průběhu zadávání hesla. V tomto ukazateli lze dosáhnout stoprocentního hodnocení již po 4 zadaných písmenech, protože jsou vyhodnocovány 4 abecedy.

Pod těmito ukazateli se zobrazují výsledky jednotlivých testů. Uživatel může testovat více hesel a jednotlivé výsledky si prohlížet a porovnávat ve výsledkovém listu. Podrobnosti o tomto výpisu se dočtete v následujících kapitolách. Program obsahuje celkově jedenáct procedur a jednu funkci.

5.2 PROCEDURY

Zde jsou uvedeny většinou fragmenty procedur, úplné programy jsou v přílohách.

5.2.1 VYHODNOCENÍ DÉLKY HESLA

```
procedure VyhodnotDelku(heslo: string);
begin
  case Length(Form1.Heslo.Text) of
    0,1,2: begin
      Form1.Vysledek.Text:='Délka: Příliš krátké heslo! ..... 0
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    3: begin
      skore:=skore+10;
      Form1.Vysledek.Text:='Délka: Heslo délky tři, to je strašně málo! ..... 10
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
```

Tato jednoduchá procedura je založena na přepínači, který podle délky zadaného hesla určí bodové ohodnocení délky, přičte jej k celkovému hodnocení a vypíše o tom patřičné hlášení.

5.2.2 VYHODNOCENÍ ROZMANITOSTI

```
procedure VyhodnotRozmanitost(heslo: string);
begin
```

```
          case SpocitejRozmanitost(heslo) of
           0: begin
             Form1.Vysledek.Text:='Rozmanitost: Prázdné heslo! ..... 0
bodů' + chr(13) + Form1.Vysledek.Text;
           end;
           1: begin
             skore:=skore+10;
             Form1.Vysledek.Text:='Rozmanitost: Použit pouze jeden typ znaků! ..... 10
bodů' + chr(13) + Form1.Vysledek.Text;;
           end;
```

Procedura VyhodnotRozmanitost funguje stejně jako předchozí procedura pro ohodnocení délky s tím rozdílem, že nejdříve musí zavolat mnou vytvořenou funkci pro výpočet rozmanitosti, jelikož na rozdíl od délky řetězce delphi takovou funkci nezná.

5.2.3 VYHODNOCENÍ SLOVNÍKŮ

V této proceduře jsem nejprve musel nastavit logickou proměnnou nalezeno na hodnotu false a také nulovou hodnotu čtyřem proměnným zmenyVeVelikosti, zamenyCislicAPismen, smazanePredpony, smazanePripony.

```
       assign(f,'slovnik.txt');
         reset(f);
         while not eof(f) do begin
           readln(f, slovo);
```

Tyto čtyři řádky se starají o to, aby můj program byl propojen s textovým souborem slovnik.txt a četlo se z něho. Dále se zde nachází cyklus while, který čte po řádcích ze souboru, dokud nenarazí na konec souboru.

```
       for i := 1 to length(heslo) do begin
             if (ord(heslo[i])>64) and (ord(heslo[i])<91) then begin
               heslo[i]:=chr(ord(heslo[i])+32);
               zmenyVeVelikosti:=zmenyVeVelikosti+1;
             end;
             if heslo[i]='0' then begin
               heslo[i]:='o';
               zamenyCislicAPismen:=zamenyCislicAPismen+1;
     end;
```

Tato část programu řeší záměny velkých a malých písmen, dále typické záměny písmene o a čísla nula, písmene l a čísla jedna, písmene E za číslo tři a písmene S za číslo pět. Jestliže je nějaká záměna provedena zvýší se tato proměnná o jedna.

```
       predpony:=-1;
           for i:= 1 to length(heslo)-length(slovo)+1 do begin
             predpony:=predpony+1;
```

```
            if heslo[i]=slovo[1] then begin
              pocet:=1;
              for j:= 2 to length(slovo) do begin
                if heslo[i+j-1]=slovo[j] then begin
                  pocet:=pocet+1;
                  if length(slovo)=pocet then begin
                    delete(heslo,1,predpony);
                    smazanePredpony:=predpony;
                  end;
                end;
              end;
            end;
end;
```

Předpony se vyhledávají tak, že se postupně berou slova ze slovníku. Poté se porovnává první znak hesla s daným slovem. Pokud se znaky nerovnají, zvýší se předpona o jedničku a porovnává se druhý znak hesla s prvním znakem slova ze slovníku. Takhle se pokračuje, dokud se nenajde shoda. Jakmile máme shodné znaky, začnou se porovnávat znaky následující, a to až do konce slova ze slovníku.

Pokud se během tohoto porovnávání znaky neshodují, pokračuje se zase od začátku, ale na další pozici v hesle, dokud se nedojede na konec hesla. Pokud se ovšem shodují všechny následující znaky, pak program našel v hesle podslovo, které se vyskytuje ve slovníku. V proměnné smazanePredpony pak má uložen počet znaků, který se nachází před tímto podslovem. Jakmile program nalezne podslovo, provede patřičné úkony a ukončí jeho další prohledávání slov ve slovníku. Proto je důležité, aby slovník byl seřazen podle velikosti slov od nejdelších, aby nenašel kratší podslovo, než heslo ve skutečnosti obsahuje, a neukončil prohledávání.

```
        pripony:=-1;
            for i:= length(heslo) downto length(slovo) do begin
              pripony:=pripony+1;
              if heslo[i]=slovo[length(slovo)] then begin
                pocet:=1;
                for j:=length(slovo)-1 downto 1 do begin
                  if heslo[i-length(slovo)+j]=slovo[j] then begin
                    pocet:=pocet+1;
                    if length(slovo)=pocet then begin
                      delete(heslo,length(heslo)-pripony+1,pripony);
                      smazanePripony:=pripony;
                    end;
                  end;
```

```
              end;
            end;
    end;
```

Hledání přípon je založeno na stejném principu jako hledání předpon s tím rozdílem, že řetězce se procházejí od konce.

```
        if (slovo=heslo) and (smazanePredpony<5) and (smazanePripony<5) then begin
              if zmenyVeVelikosti>0 then begin
                bonus:=zmenyVeVelikosti*5+5;
                skore:=skore+bonus;
                Form1.Vysledek.Text:='Slovník:   Změny   ve   velikosti   písmen!   '   +
inttostr(zmenyVeVelikosti) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13) +
Form1.Vysledek.Text;
              end;
              if zamenyCislicAPismen>0 then begin
                bonus:=zamenyCislicAPismen*5+5;
                skore:=skore+bonus;
                Form1.Vysledek.Text:='Slovník:   Záměny   číslic   a   písmen!   '   +
inttostr(zamenyCislicAPismen) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13)
+ Form1.Vysledek.Text;
              end;
              skore:=skore-60;
              Form1.Vysledek.Text:='Slovník: Heslo nalezeno ve slovníku! ..... -60
bodů' + chr(13) + Form1.Vysledek.Text;
              nalezeno:=true;
              exit;
            end;
  end;
```

Po provedení předchozích algoritmů se provede porovnání, zda se heslo rovná řetězci ve slovníku a pokud ano, provede se nejprve výpis, zda byly v heslu zaznamenány nějaké jednoduché záměny nebo přidány znaky před či za heslo, a poté výpis, že se heslo nachází ve slovníku. Odečte se 60 bodů, nicméně za každou záměnu, předponu nebo příponu lze získat bonusové body, které se přičítají k celkovému hodnocení.

```
        close(f);
          if not nalezeno then VyhodnotSlovnikyPozpatku(heslo2);
        end;
```

Pokud program prohledal všechny záznamy ze slovníku a nebyl ani jednou úspěšný, zavře soubor se slovníkem a pokusí se ještě zavolat proceduru VyhodnotSlovnikyPozpatku, která, až na detaily ve výpisech a celkovém ohodnocení hesla, funguje totožně jako tato procedura, jen si na začátku převrátí heslo pozpátku.

5.2.4 VYHODNOCENÍ POSLOUPNOSTI

```
        assign(f,'klavesnicove_posloupnosti.txt');
```

```
          reset(f);
          while not eof(f) do begin
            readln(f, slovo);
            if slovo=heslo then begin
              skore:=skore-60;
              Form1.Vysledek.Text:='Posloupnost: Heslo je klávesnicová posloupnost! ... -60
bodů' + chr(13) + Form1.Vysledek.Text;
            end;
          end;
          close(f);
```

V této proceduře používám stejný způsob načtení souboru, jako jsem použil v předchozích procedurách. Využil jsem také stejný cyklus a čtu ze souboru též po řádcích. Pokud se zadané heslo vyskytuje ve slovníku posloupností, je odečteno 60 bodů z celkového hodnocení a soubor, ze kterého je čteno, se zavře.

```
        nalezeno:=true;
          nalezeno2:=true;
          for i := 2 to length(heslo) do begin
            if ord(heslo[i-1])<>ord(heslo[i])-1 then begin
              nalezeno:=false;
            end;
            if ord(heslo[i-1])<>ord(heslo[i])+1 then begin
              nalezeno2:=false;
            end;
          end;
          if nalezeno or nalezeno2 then begin
            skore:=skore-60;
            Form1.Vysledek.Text:='Posloupnost: Heslo je abecední posloupnost! ....... -60
bodů' + chr(13) + Form1.Vysledek.Text;
  end;
```

Pro zjištění, zda heslo je abecední posloupnost, máme tento for cyklus, který postupně porovnává dva sousedící znaky zadaného hesla a zkoumá, zda se nachází vedle sebe i v ascii tabulce, a to oběma možnými směry.

```
        nalezeno:=true;
          for i := 2 to length(heslo) do begin
            if heslo[i]<>heslo[1] then nalezeno:=false;
          end;
          if nalezeno then begin
            skore:=skore-60;
            Form1.Vysledek.Text:='Posloupnost: Heslo je tvořeno stejnými znaky! ..... -60
bodů' + chr(13) + Form1.Vysledek.Text;
          end;
end;
```

Na podobném principu je založeno také vyhodnocení, zda je heslo tvořeno stejnými znaky. Na začátku předpokládáme, že heslo takto tvořeno je (nalezeno:=true). Poté postupně procházíme znaky zadaného hesla a porovnáváme ho s prvním znakem.

5.2.5 VYHODNOCENÍ CELKOVÉ SÍLY HESLA

```
        skore:=0;
          pocetPokusu:=pocetPokusu+1;
          Form1.Vysledek.Text:=
'_____________________________________________________' + chr(13) + chr(13) +
Form1.Vysledek.Text;
          VyhodnotPosloupnosti(heslo);
          VyhodnotSlovniky(heslo);
          VyhodnotRozmanitost(heslo);
          VyhodnotDelku(heslo);
          if skore<0 then skore:=0;
          if skore>100 then skore:=100;
          Form1.Vysledek.Text:='Celková síla hesla je ' + inttostr(skore) + ' bodů.' +
chr(13) + chr(13) + Form1.Vysledek.Text;
          Form1.Vysledek.Text:='Testované heslo: ' + Form1.Heslo.Text + chr(13) +
Form1.Vysledek.Text;
  Form1.Vysledek.Text:='Výsledek ' + inttostr(pocetPokusu) + '. testu:' + chr(13) +
Form1.Vysledek.Text;
```

Celkové hodnocení síly hesla přičítá jednotlivé hodnocení z předchozích procedur. Poté se provede patřičný výpis. Výpis je omezen nulou a stovkou. Při dosáhnutí záporného hodnocení hesla uživatel získá nula bodů. Naopak při kvalitním velmi silném heslu nedostane uživatel více jak sto bodů.

5.2.6 GENERÁTOR

```
        var genHeslo: string;
            i: integer;
        begin
          genHeslo := '';
          Randomize;
          for i := 1 to 10 do
          begin
            genHeslo:= genHeslo + Chr(Random(94)+33);
          end;
          VygenerovaneHeslo.Text:=genHeslo;
end;
```

Generátor obsahuje cyklus s pevným počtem opakování, takže délka generovaných hesel se nikdy bez zásahu do programu, nezmění. Cyklus funguje tak, že do prázdné proměnné genHeslo ukládá v každém kroku náhodný znak z ASCII tabulky. Tento cyklus se opakuje právě desetkrát.

5.3 FUNKCE

Funkce na rozdíl od procedury vrací nějakou hodnotu.

5.3.1 SPOČÍTÁNÍ ROZMANITOSTI ZNAKŮ

```
for i := 1 to Length(heslo) do begin
  if (ord(heslo[i])>64) and (ord(heslo[i])<91) and not velkaPismena then begin
    velkaPismena:=true;
    rozmanitost:=rozmanitost+1;
  end else
    if (ord(heslo[i])>96) and (ord(heslo[i])<123) and not malaPismena then begin
      malaPismena:=true;
      rozmanitost:=rozmanitost+1;
    end else
      if (ord(heslo[i])>47) and (ord(heslo[i])<58) and not cislice then begin
        cislice:=true;
        rozmanitost:=rozmanitost+1;
      end else
         if ((ord(heslo[i])<48)
              or ((ord(heslo[i])>57) and (ord(heslo[i])<65))
              or ((ord(heslo[i])>90) and (ord(heslo[i])<97))
              or (ord(heslo[i])>122))
            and not ostatniZnaky then begin
          ostatniZnaky:=true;
          rozmanitost:=rozmanitost+1;
        end;
end;
```

Funkce na spočítání rozmanitosti znaků využívá ASCII tabulku, kde podle příslušné hodnoty zjišťuje, z jaké abecedy daný znak je. Na následujícím obrázku si můžete prohlédnout ASCII tabulku, která byla převzata z http://asciiset.com/

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
32	20	Space	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	□

Obrázek 9: ASCII tabulka

Tato ASCII tabulka je oříznuta o prvních 31 znaků, které zobrazují neviditelné znaky typu odřádkování atd. V ASCII tabulce se po řádcích nacházejí jednotlivé znaky a k nim příslušné zobrazení v dekadické a hexadecimální hodnotě. Dekadické hodnoty znaků jsem použitím funkce ord využíval pro vyhodnocení, do které abecedy příslušný znak patří. Přičemž jsem rozpoznával tyto 4 abecedy:

- Malá písmena (97 – 122)
- Velká písmena (65 – 90)
- Číslice (48 – 57)
- Ostatní znaky (zbylé hodnoty)

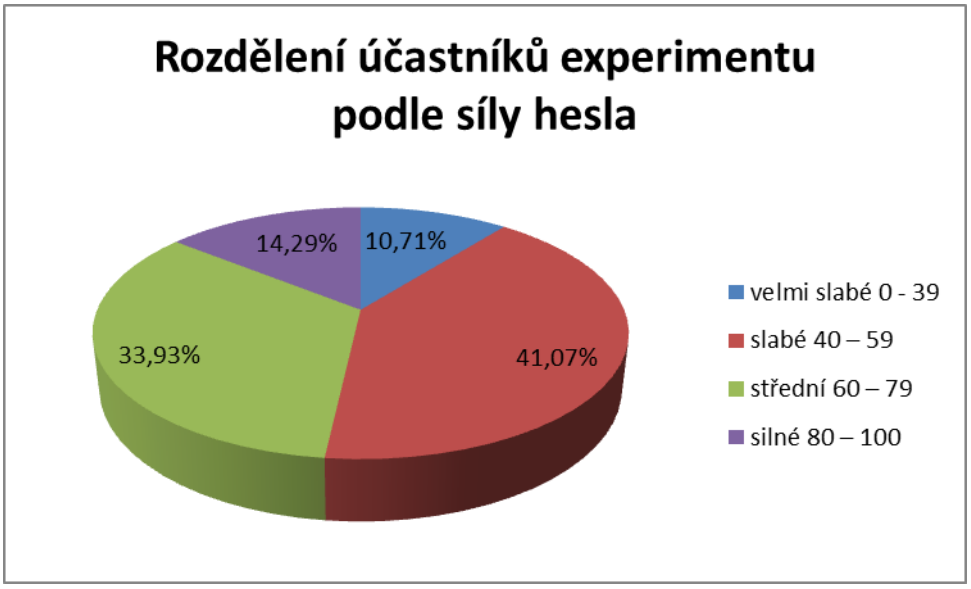
Samotný výpočet funkce probíhá tak, že na začátku jsou nastaveny logické hodnoty těchto 4 abeced na false a zároveň proměnná rozmanitost je rovna nule. Poté se prochází zadané heslo po znacích, a pokud znak spadá do abecedy, která má logickou hodnotu false, pak se tato hodnota změní na true a hodnota rozmanitosti se zvýší o jedničku.

5.4 EXPERIMENT S APLIKACÍ TESTOVAČ

Pro vyzkoušení aplikace Testovač v praxi byl proveden následující experiment. Prostřednictvím sociální sítě byly osloveny desítky lidí, kteří dostali vytvořenou aplikaci a měli za úkol pomocí ní ohodnotit své heslo, které používají například do emailové schránky. Byli ujištěni o tom, že heslo se nikam neukládá, ani nikam neposílá, pouze se zobrazí v okně aplikace. Dne 15.6.2012 byla umístěna tato aplikace na portál hhtp://uloz.to, kde si uživatelé stáhli moji aplikaci. Tato aplikace byla dne 17.6.2012 smazána. Celkem se zúčastnilo 56 lidí. Výsledky testu jsou shrnuty v této tabulce:

Tabulka 2: Experiment s Testovačem

Síla hesla	Bodové ohodnocení hesla	Počet dotázaných (v procentech)
velmi slabé	0 - 39	6 (10,71%)
Slabé	40 – 59	23 (41,07%)
Střední	60 – 79	19 (33,93%)
Silné	80 – 100	8 (14,29%)



Z experimentu, který jsem po vytvoření aplikace provedl, vyplynulo, že většina účastníků experimentu používají spíše slabá hesla. Jen necelých 15% dosáhlo alespoň osmdesáti bodů. Průměrná délka hesla byla šest znaků.

6 ZÁVĚR

Cílem mé bakalářské práce bylo seznámit uživatele s problematikou týkající se počítačových hesel. Uživatelům jsem poskytl postup a rady, jak správně vytvořit silné heslo, neboť znalost tvorby silného hesla je nedílnou součástí bezpečnosti dat.

V teoretické části bylo dále vysvětleno, jaké jsou možnosti biometrické autentizace, která by mohla v budoucnu vést k větší bezpečnosti, než poskytují silná hesla. Dále bylo nastíněno, jakým způsobem by mohla vypadat budoucnost autentizace. Také je možno se dočíst v této práci o tom, jaké jsou možnosti generování hesel na různých internetových stránkách, nebo který software si může stáhnout pro účel generování hesel.

Dále byla rozebrána problematika spojená se správou počítačových hesel a byly uvedeny možnosti, kde a jak testovat heslo. Jedna z kapitol pak nabízí porovnání několika stejných hesel na dvou na sobě nezávislých testovačích.

V praktické části jsem vytvořil program Testovač, ve kterém mi dělalo největší potíže rozhodnout o bodovém ohodnocení různorodých hesel. Obtížné bylo rozhodnout při udělování bodů heslu, které se vyskytuje ve slovníku a obsahuje více než čtyři předpony či přípony. Dle mého názoru se jedná již o silné heslo. Za opravdu silné heslo bych považoval jen to, které v mé aplikaci dostane více než 80 bodů. Pokud heslo dosáhne méně než 40 bodů, pak je velmi slabé a lehko podlehne případnému útoku.

Generátor v praktické části nabízí uživateli desetimístné heslo, které si může hned i otestovat. Výhodou generátoru je hodně malá pravděpodobnost na utvoření hesla, které by se mohlo vyskytovat ve slovníku, nebo by bylo tvořeno nějakou z posloupností. Největší procentuální šance je, že se vygeneruje v heslu speciální znak, tudíž i když heslo nebude obsahovat všechny čtyři možné abecedy, tak bude velmi bezpečné.

Součástí práce byl experiment, který testoval užívaná hesla u několika desítek uživatelů.

7 SEZNAM OBRÁZKŮ

8 SEZNAM LITERATURY

1. Moderní metody zabezpečení uživatelských počítačů (2. díl): Alternativy k šifrování pevného disku. SystemOnline: S přehledem ve světě informačních technologií [online]. 2011 [cit. 2012-06-19]. Dostupné z: http://www.systemonline.cz/clanky/moderni-metody-zabezpeceni-uzivatelskych-pocitacu-2.-dil.htm
2. Bezpečnost: Autentizace a autorizace. Web4Company [online]. [cit. 2012-06-19]. Dostupné z: http://www.web4company.cz/bezpecnost-autentizace-autorizace/
3. Autentizace: hlasová biometrie. CleverAndSmart ICT managment [online]. 2011 [cit. 2012-06-19]. Dostupné z: http://www.cleverandsmart.cz/autentizace-hlasova-biometrie/
4. Autentizační metody založené na biometrických informacích. Access server [online]. 2010 [cit. 2012-06-19]. Dostupné z: http://access.feld.cvut.cz/view.php?cisloclanku=2010110002
5. Elektronický podpis a jeho využití. BusinessInfo.cz: Oficiální portál pro podnikání a export [online]. 2002 [cit. 2012-06-19]. Dostupné z: http://www.businessinfo.cz/cz/clanek/it-telekomunikace/elektronicky-podpis-a-jeho-vyuziti/1000473/2984
6. Slovníkové útoky a útok silou. Dusatko.org [online]. 2007 [cit. 2012-06-19]. Dostupné z: http://www.dusatko.org/en/node/63
7. Lámání hesel v praxi (1.). Lupa.cz [online]. 2005 [cit. 2012-06-19]. Dostupné z: http://www.lupa.cz/clanky/lamani-hesel-v-praxi-1/
8. Wikipedia: Bezpečné heslo. [online]. [cit. 2012-05-29]. Dostupné z: http://cs.wikipedia.org/wiki/Bezpe%C4%8Dn%C3%A9_heslo
9. JURAČKA, Michal. H.S.G.I. - Informační systémy: Generátor hesel. [online]. [cit. 2012-06-19]. Dostupné z: http://www.hsgi.cz/generator-hesel/
10. PIN. Wikipedie: Otevřená encyklopedie [online]. 2012 [cit. 2012-06-25]. Dostupné z: http://cs.wikipedia.org/wiki/PIN
11. VOLČÍK, J. *Teorie konečných automatů*. Praha: Ediční středisko Českého vysokého učení technického, 1990, 256 s, ISBN 80-01-00443-0.
12. SCHWAGER, M. *Možnosti hodnocení kvality hesel*. Bakalářská práce, MU Brno, 2005

9 RESUMÉ

The aim of my bachelor thesis is to acquaint with the issues of computer passwords. I have given some course of action and advice to the user how to create a strong password because the knowledge about strong passwords is an integral part of the safety of data.

In the theoretical part of my bachelor thesis there were explained the possibilities of biometrical authentication, which could aim to better security then using strong system's passwords. It was further explained which way the future of authentication might look. The user can read about the possibilities of password generation that he can find on different internet pages or which software he can download for the purpose of generating passwords.

Further the issues related to the management of computer passwords were analyzed and opportunities, where to test passwords, were given. One of the chapters offers the comparison of disparate passwords on two independent testers.

In the practical part I created a program, tester, where I had the most trouble with deciding about the score of diverse passwords. I had a big dilemma in giving points to a password, which occurs in the dictionary in includes more than four prefixes or suffixes. In my opinion it is already a strong password. For a really strong password I would consider only a password that gets more than 80 pointy in my application. If a password gets less than 40 points it is very weak and it easily succumbs to possible attack.

The generator from the practical part offers a ten-digit password the user which he can immediately test. The advantage of the generator is the low chance of building a password that could exist in the dictionary or that would consist of any sequence. The biggest percentage chance is to generate a special sign thus even if the password does not contain all of the four possible alphabets, it will be very safe.

This word should be beneficial to all users who do not know how a strong password has to look like. This work is suitable for most users who think that their password is really safe, because unfortunately even in high schools we can meat weak passwords.

10 PŘÍLOHY

10.1 OBSAH PŘILOŽENÉHO CD

K této práci je přiloženo CD, kde se nachází vlastní vytvořená aplikace Testovač, obsah této bakalářské práce v PDF a word formátu.

10.2 PROGRAM TESTOVAČ

```
unit Unit1;

interface

uses
  System.SysUtils, System.Types, System.UITypes, System.Classes, System.Variants,
  FMX.Types, FMX.Controls, FMX.Forms, FMX.Dialogs, FMX.Edit, FMX.Layouts,
  FMX.Memo, FMX.Objects, FMX.Effects, FMX.Ani, FMX.ExtCtrls;

type
  TForm1 = class(TForm)
    Heslo: TEdit;
    Delka: TProgressBar;
    Rozmanitost: TProgressBar;
    DelkaNadpis: TLabel;
    RozmanitostNadpis: TLabel;
    ShadowEffect1: TShadowEffect;
    Generator: TButton;
    PozadavkyNaHeslo: TMemo;
    ShadowEffect2: TShadowEffect;
    VygenerovaneHeslo: TMemo;
    ShadowEffect3: TShadowEffect;
    TestovaciTlacitko: TButton;
    Vysledek: TMemo;
    ShadowEffect4: TShadowEffect;
    PozadavkyNaHesloNadpis: TLabel;
    VysledkyTestuNadpis: TLabel;
    procedure HesloClick(Sender: TObject);
    procedure GeneratorClick(Sender: TObject);
    procedure HesloKeyUp(Sender: TObject; var Key: Word;
      var KeyChar: Char; Shift: TShiftState);
```

```
    procedure TestovaciTlacitkoClick(Sender: TObject);
    procedure PozadavkyNaHesloChange(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  skore, pocetPokusu: integer;
implementation

{$R *.fmx}
```

10.2.1 FUNKCE NA SPOČÍTÁNÍ ROZMANITOSTI

```
function SpocitejRozmanitost(heslo:string):integer;
var i, rozmanitost: integer;
    velkaPismena, malaPismena, cislice, ostatniZnaky: boolean;
begin
  rozmanitost:=0;
  velkaPismena:=false;
  malaPismena:=false;
  cislice:=false;
  ostatniZnaky:=false;
  for i := 1 to Length(heslo) do begin
    if (ord(heslo[i])>64) and (ord(heslo[i])<91) and not velkaPismena then begin
      velkaPismena:=true;
      rozmanitost:=rozmanitost+1;
    end else
      if (ord(heslo[i])>96) and (ord(heslo[i])<123) and not malaPismena then begin
        malaPismena:=true;
        rozmanitost:=rozmanitost+1;
      end else
        if (ord(heslo[i])>47) and (ord(heslo[i])<58) and not cislice then begin
          cislice:=true;
          rozmanitost:=rozmanitost+1;
```

```
        end else
           if ((ord(heslo[i])<48)
                or ((ord(heslo[i])>57) and (ord(heslo[i])<65))
                or ((ord(heslo[i])>90) and (ord(heslo[i])<97))
                or (ord(heslo[i])>122))
              and not ostatniZnaky then begin
            ostatniZnaky:=true;
            rozmanitost:=rozmanitost+1;
          end;
  end;
  SpocitejRozmanitost:=rozmanitost;
end;
```

10.2.2 PROCEDURA NA VYHODNOCENÍ DÉLKY HESLA

```
procedure VyhodnotDelku(heslo: string);
begin
  case Length(Form1.Heslo.Text) of
    0,1,2: begin
      Form1.Vysledek.Text:='Délka: Příliš krátké heslo! ..... 0
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    3: begin
      skore:=skore+10;
      Form1.Vysledek.Text:='Délka: Heslo délky tři, to je strašně málo! ..... 10
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    4: begin
      skore:=skore+20;
      Form1.Vysledek.Text:='Délka: Heslo délky čtyři, to je málo! ..... 20
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    5: begin
      skore:=skore+30;
      Form1.Vysledek.Text:='Délka: Heslo délky pět, chtělo by to delší! ..... 30
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    6: begin
```

```
      skore:=skore+40;
      Form1.Vysledek.Text:='Délka: Heslo délky šest, ještě alespoň dva znaky! .. 40
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    7: begin
      skore:=skore+50;
      Form1.Vysledek.Text:='Délka: Heslo délky sedm! ........................... 50
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    8,9: begin
      skore:=skore+60;
      Form1.Vysledek.Text:='Délka: Splněna požadovaná délka na silné heslo! .... 60
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    else begin
      skore:=skore+70;
      Form1.Vysledek.Text:='Délka: Délka větší rovna desíti, bonus 10 bodů! . 60+10
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
  end;
end;
```

10.2.3 PROCEDURA NA VYHODNOCENÍ ROZMANITOSTI ZNAKŮ

```
procedure VyhodnotRozmanitost(heslo: string);
begin
   case SpocitejRozmanitost(heslo) of
    0: begin
      Form1.Vysledek.Text:='Rozmanitost: Prázdné heslo! ......................... 0
bodů' + chr(13) + Form1.Vysledek.Text;
    end;
    1: begin
      skore:=skore+10;
      Form1.Vysledek.Text:='Rozmanitost: Použit pouze jeden typ znaků! ......... 10
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    2: begin
      skore:=skore+20;
```

```
      Form1.Vysledek.Text:='Rozmanitost: Použity dva typy znaků! ..... 20
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    3: begin
      skore:=skore+30;
      Form1.Vysledek.Text:='Rozmanitost: Použity tři typy znaků! ..... 30
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
    4: begin
      skore:=skore+40;
      Form1.Vysledek.Text:='Rozmanitost: Použity všechny typy znaků! ..... 40
bodů' + chr(13) + Form1.Vysledek.Text;;
    end;
  end;
end;
```

10.2.4 PROCEDURA NA VYHODNOCENÍ SLOVNÍKŮ POZPÁTKU

```
procedure VyhodnotSlovnikyPozpatku(heslo: string);
var f: text;
    slovo: string;
    nalezeno: boolean;
    i, j, zmenyVeVelikosti, zamenyCislicAPismen, predpony, smazanePredpony,
pripony, smazanePripony, pocet, bonus: integer;
begin
  nalezeno:=false;
  zmenyVeVelikosti:=0;
  zamenyCislicAPismen:=0;
  smazanePredpony:=0;
  smazanePripony:=0;

  //otoceni hesla
  slovo:=heslo;
  for i:= 1 to length(heslo) do begin
    slovo[i]:=heslo[length(heslo)-i+1]
  end;
  heslo:=slovo;
  // nacteni slovniku
```

```
assign(f,'slovnik.txt');
reset(f);
while not eof(f) do begin
  readln(f, slovo);
  //zameny pismen
  for i := 1 to length(heslo) do begin
    //zmena velikosti pismene
    if (ord(heslo[i])>64) and (ord(heslo[i])<91) then begin
      heslo[i]:=chr(ord(heslo[i])+32);
      zmenyVeVelikosti:=zmenyVeVelikosti+1;
    end;
    //zmena pismene 'o' za cislici '0'
    if heslo[i]='0' then begin
      heslo[i]:='o';
      zamenyCislicAPismen:=zamenyCislicAPismen+1;
    end;
    //zmena pismene 'l' za cislici '1'
    if heslo[i]='1' then begin
      heslo[i]:='l';
      zamenyCislicAPismen:=zamenyCislicAPismen+1;
    end;
    //zmena pismene 'E' za cislici '3'
    if heslo[i]='3' then begin
      heslo[i]:='e';
      zamenyCislicAPismen:=zamenyCislicAPismen+1;
    end;
    //zmena pismene 'S' za cislici '5'
    if heslo[i]='5' then begin
      heslo[i]:='s';
      zamenyCislicAPismen:=zamenyCislicAPismen+1;
    end;
  end;
  //predpony .. nutno seradit slovnik podle velikosti slov od nejdelsich
  predpony:=-1;
  for i:= 1 to length(heslo)-length(slovo)+1 do begin
```

```
      predpony:=predpony+1;
      if heslo[i]=slovo[1] then begin
        pocet:=1;
        for j:= 2 to length(slovo) do begin
          if heslo[i+j-1]=slovo[j] then begin
            pocet:=pocet+1;
            if length(slovo)=pocet then begin  //pokud se vsechna pismena rovnala,
nasel podslovo a tudiz i predponu, kterou odstrani
              delete(heslo,1,predpony);
              smazanePredpony:=predpony;
            end;
          end;
        end;
      end;
    end;
    //pripony
    pripony:=-1;
    for i:= length(heslo) downto length(slovo) do begin
      pripony:=pripony+1;
      if heslo[i]=slovo[length(slovo)] then begin
        pocet:=1;
        for j:=length(slovo)-1 downto 1 do begin
          if heslo[i-length(slovo)+j]=slovo[j] then begin
            pocet:=pocet+1;
            if length(slovo)=pocet then begin  //pokud se vsechna pismena rovnala,
nasel podslovo a tudiz i priponu, kterou odstrani
              delete(heslo,length(heslo)-pripony+1,pripony);
              smazanePripony:=pripony;
            end;
          end;
        end;
      end;
    end;
    if (slovo=heslo) and (smazanePredpony<5) and (smazanePripony<5) then begin
      if zmenyVeVelikosti>0 then begin
        bonus:=zmenyVeVelikosti*5+5;
```

```
              skore:=skore+bonus;

              Form1.Vysledek.Text:='Slovník:   Změny   ve   velikosti   písmen!   '   +
inttostr(zmenyVeVelikosti) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13) +
Form1.Vysledek.Text;

            end;

            if zamenyCislicAPismen>0 then begin

              bonus:=zamenyCislicAPismen*5+5;

              skore:=skore+bonus;

              Form1.Vysledek.Text:='Slovník:   Záměny   číslic   a   písmen!   '   +
inttostr(zamenyCislicAPismen) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13)
+ Form1.Vysledek.Text;

            end;

            if (smazanePredpony>0) then begin

              bonus:=smazanePredpony*5+5;

              skore:=skore+bonus;

              Form1.Vysledek.Text:='Slovník:   Přidány   znaky   za   heslo!   '   +
      inttostr(smazanePredpony) + 'x ..... ' + inttostr(bonus) + ' bodů' +
      chr(13) + Form1.Vysledek.Text;

            end;

            if (smazanePripony>0) and (smazanePripony<5) then begin

              bonus:=smazanePripony*5+5;

              skore:=skore+bonus;

              Form1.Vysledek.Text:='Slovník:   Přidány   znaky   před   heslo!   '   +
      inttostr(smazanePripony) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13)
      + Form1.Vysledek.Text;

            end;

            skore:=skore-50;

            Form1.Vysledek.Text:='Slovník: Heslo nalezeno pozpátku ve slovníku! ..... -50
      bodů' + chr(13) + Form1.Vysledek.Text;

            nalezeno:=true;

            exit;

          end;

        end;

       if not nalezeno then Form1.Vysledek.Text:='Slovník: Heslo nenalezeno ve slovníku!
      ..... 0 bodů' + chr(13) + Form1.Vysledek.Text;

        close(f);

      end;
```

10.2.5 PROCEDURA NA VYHODNOCENÍ SLOVNÍKŮ

```
procedure VyhodnotSlovniky(heslo: string);
var f: text;
    slovo,heslo2: string;
    nalezeno: boolean;
    i, j, zmenyVeVelikosti, zamenyCislicAPismen, predpony, smazanePredpony,
pripony, smazanePripony, pocet, bonus: integer;
begin
  nalezeno:=false;
  zmenyVeVelikosti:=0;
  zamenyCislicAPismen:=0;
  smazanePredpony:=0;
  smazanePripony:=0;
  heslo2:=heslo;
  // nacteni slovniku
  assign(f,'slovnik.txt');
  reset(f);
  while not eof(f) do begin
    readln(f, slovo);
    //zameny pismen
    for i := 1 to length(heslo) do begin
      //zmena velikosti pismene
      if (ord(heslo[i])>64) and (ord(heslo[i])<91) then begin
        heslo[i]:=chr(ord(heslo[i])+32);
        zmenyVeVelikosti:=zmenyVevelikosti+1;
      end;
      //zmena pismene 'o' za cislici '0'
      if heslo[i]='0' then begin
        heslo[i]:='o';
        zamenyCislicAPismen:=zamenyCislicAPismen+1;
      end;
      //zmena pismene 'l' za cislici '1'
      if heslo[i]='1' then begin
        heslo[i]:='l';
        zamenyCislicAPismen:=zamenyCislicAPismen+1;
```

```
      end;
      //zmena pismene 'E' za cislici '3'
      if heslo[i]='3' then begin
        heslo[i]:='e';
        zamenyCislicAPismen:=zamenyCislicAPismen+1;
      end;
      //zmena pismene 'S' za cislici '5'
      if heslo[i]='5' then begin
        heslo[i]:='s';
        zamenyCislicAPismen:=zamenyCislicAPismen+1;
      end;
    end;
    //predpony .. nutno seradit slovnik podle velikosti slov od nejdelsich
    predpony:=-1;
    for i:= 1 to length(heslo)-length(slovo)+1 do begin
      predpony:=predpony+1;
      if heslo[i]=slovo[1] then begin
        pocet:=1;
        for j:= 2 to length(slovo) do begin
          if heslo[i+j-1]=slovo[j] then begin
            pocet:=pocet+1;
            if length(slovo)=pocet then begin  //pokud se vsechna pismena rovnala,
nasel podslovo a tudiz i predponu, kterou odstrani
              delete(heslo,1,predpony);
              smazanePredpony:=predpony;
            end;
          end;
        end;
      end;
    end;
    //pripony
    pripony:=-1;
    for i:= length(heslo) downto length(slovo) do begin
      pripony:=pripony+1;
      if heslo[i]=slovo[length(slovo)] then begin
```

```
        pocet:=1;
        for j:=length(slovo)-1 downto 1 do begin
          if heslo[i-length(slovo)+j]=slovo[j] then begin
            pocet:=pocet+1;
            if length(slovo)=pocet then begin  //pokud se vsechna pismena rovnala, nasel podslovo a tudiz i priponu, kterou odstrani
              delete(heslo,length(heslo)-pripony+1,pripony);
              smazanePripony:=pripony;
            end;
          end;
        end;
      end;
    end;
    if (slovo=heslo) and (smazanePredpony<5) and (smazanePripony<5) then begin
      if zmenyVeVelikosti>0 then begin
        bonus:=zmenyVeVelikosti*5+5;
        skore:=skore+bonus;
        Form1.Vysledek.Text:='Slovník:  Změny ve velikosti písmen! ' + inttostr(zmenyVeVelikosti) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13) + Form1.Vysledek.Text;
      end;
      if zamenyCislicAPismen>0 then begin
        bonus:=zamenyCislicAPismen*5+5;
        skore:=skore+bonus;
        Form1.Vysledek.Text:='Slovník:  Záměny číslic a písmen! ' + inttostr(zamenyCislicAPismen) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13) + Form1.Vysledek.Text;
      end;
      if (smazanePredpony>0) then begin
        bonus:=smazanePredpony*5+5;
        skore:=skore+bonus;
        Form1.Vysledek.Text:='Slovník:  Přidány znaky před heslo! ' + inttostr(smazanePredpony) + 'x ..... ' + inttostr(bonus) + ' bodů' + chr(13) + Form1.Vysledek.Text;
      end;
      if (smazanePripony>0) and (smazanePripony<5) then begin
        bonus:=smazanePripony*5+5;
```

```
        skore:=skore+bonus;
        Form1.Vysledek.Text:='Slovník:  Přidány  znaky  za  heslo!  '  +
inttostr(smazanePripony) + 'x ..............  ' + inttostr(bonus) + ' bodů' +
chr(13) + Form1.Vysledek.Text;
      end;
      skore:=skore-60;
      Form1.Vysledek.Text:='Slovník: Heslo nalezeno ve slovníku! ..............  -60
bodů' + chr(13) + Form1.Vysledek.Text;
      nalezeno:=true;
      exit;
    end;
  end;
  close(f);
  if not nalezeno then VyhodnotSlovnikyPozpatku(heslo2);
end;
```

10.2.6 PROCEDURA NA VYHODNOCENÍ POSLOUPNOSTÍ

```
procedure VyhodnotPosloupnosti(heslo: string);
var f: text;
    slovo: string;
    nalezeno, nalezeno2: boolean;
    i : integer;
begin
  // klavesnicove posloupnosti
  assign(f,'klavesnicove_posloupnosti.txt');
  reset(f);
  while not eof(f) do begin
    readln(f, slovo);
    if slovo=heslo then begin
      skore:=skore-60;
      Form1.Vysledek.Text:='Posloupnost: Heslo je klávesnicová posloupnost! ... -60
bodů' + chr(13) + Form1.Vysledek.Text;
    end;
  end;
  close(f);
  // abecedni posloupnosti
  nalezeno:=true;
```

```
  nalezeno2:=true;
  for i := 2 to length(heslo) do begin
    if ord(heslo[i-1])<>ord(heslo[i])-1 then begin
      nalezeno:=false;
    end;
    if ord(heslo[i-1])<>ord(heslo[i])+1 then begin
      nalezeno2:=false;
    end;
  end;
  if nalezeno or nalezeno2 then begin
    skore:=skore-60;
    Form1.Vysledek.Text:='Posloupnost: Heslo je abecední posloupnost! ..... -60
bodů' + chr(13) + Form1.Vysledek.Text;
  end;
  // posloupnost stejnych znaku
  nalezeno:=true;
  for i := 2 to length(heslo) do begin
    if heslo[i]<>heslo[1] then nalezeno:=false;
  end;
  if nalezeno then begin
    skore:=skore-60;
    Form1.Vysledek.Text:='Posloupnost: Heslo je tvořeno stejnými znaky! ..... -60
bodů' + chr(13) + Form1.Vysledek.Text;
  end;
end;
```

10.2.7 PROCEDURA NA VYHODNOCENÍ CELKOVÉ SÍLY HESLA

```
procedure VyhodnotCelkovouSilu(heslo: string);
begin
  skore:=0;
  pocetPokusu:=pocetPokusu+1;
  Form1.Vysledek.Text:=
'_____________________________________________________' + chr(13) + chr(13) +
Form1.Vysledek.Text;
  VyhodnotPosloupnosti(heslo);
  VyhodnotSlovniky(heslo);
  VyhodnotRozmanitost(heslo);
```

```
  VyhodnotDelku(heslo);
  //vypis celkoveho skore
  if skore<0 then skore:=0;
  if skore>100 then skore:=100;
 Form1.Vysledek.Text:='Celková síla hesla je ' + inttostr(skore) + ' bodů.' +
chr(13) + chr(13) + Form1.Vysledek.Text;
 Form1.Vysledek.Text:='Testované heslo: ' + Form1.Heslo.Text + chr(13) +
Form1.Vysledek.Text;
 Form1.Vysledek.Text:='Výsledek ' + inttostr(pocetPokusu) + '. testu:' + chr(13) +
Form1.Vysledek.Text;
end;
```

10.2.8 PROCEDURA NA VYGENEROVÁNÍ SILNÉHO HESLA

```
procedure TForm1.GeneratorClick(Sender: TObject);
var genHeslo: string;
    i: integer;
begin
  genHeslo := '';
  Randomize;
  for i := 1 to 10 do
  begin
    genHeslo:= genHeslo + Chr(Random(94)+33); //přidání nahodneho znaku do
'genHeslo'
  end;
  VygenerovaneHeslo.Text:=genHeslo;
end;
```

10.2.9 PROCEDURA OŠETŘUJÍCÍ KLIKNUTÍ NA TLAČÍTKO

```
procedure TForm1.TestovaciTlacitkoClick(Sender: TObject);
begin
  VyhodnotCelkovouSilu(Heslo.Text);
end;
```

10.2.10 PROCEDURA NA VYMAZÁNÍ POLE PRO ZADÁNÍ HESLA

```
procedure TForm1.HesloClick(Sender: TObject);
begin
  Heslo.Text:='';
  Delka.Value:=0;
  Rozmanitost.Value:=0;
```

```
end;

procedure TForm1.HesloKeyUp(Sender: TObject; var Key: Word;

  var KeyChar: Char; Shift: TShiftState);

begin

  Delka.Value:=Length(Heslo.Text);

  Rozmanitost.Value:=SpocitejRozmanitost(Heslo.Text);

end;
```

10.2.11 PROCEDURA VYPISUJÍCÍ POŽADAVKY NA SILNÉ HESLO

```
procedure TForm1.PozadavkyNaHesloChange(Sender: TObject);

begin

end;

//Form1.PozadavkyNaHeslo.Text:='* Délka hesla by měla být alespoň 9 znaků.' +
chr(13) + 'Rozmanitost použitých znaků spadá alespoň do tří tříd ';

end.
```