VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza a implementace špehovacího softwaru
Analysis and Implementation of Spyware

2020 Daniel Krupa

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Daniel Krupa**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Analýza a implementace špehovacího softwaru
Analysis and Implementation of Spyware

Jazyk vypracování: čeština

Zásady pro vypracování:

Práce se zabývá analýzou a tvorbou škodlivého kódu, který spadá do kategorie špehovacího softwaru (spyware). Student provede rešerši aktuálního stavu na poli bezpečnosti a ochrany proti infekci špehovacím softwarem. Dále student prezentuje trendy pozorované u aktuálních vzorků škodlivých softwarů. Na základě zjištěných faktů navrhne student architekturu vlastní implementace škodlivého softwaru a své řešení implementuje. Funkčnost řešení student otestuje v předem připraveném a bezpečném prostředí. Na závěr provede student diskuzi výsledků a navrhne možná vylepšení.

Hlavní body zadání:
1. Rešerše aktuálního stavu na poli bezpečnosti špehovacího software.
2. Návrh řešení vlastního špehovacího softwaru.
3. Implementace a testování vlastního řešení.
4. Diskuze výsledků.

Seznam doporučené odborné literatury:

[1] Reddy N., Practical Cyber Forensics: An Incident-Based Approach to Forensic Investigations, Apress, New York, USA, 2019

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Plucar, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. května 2020 .....

Abstrakt

Tato bakalářská práce se zabývá efektivitou špehovacího softwaru proti programům z kategorie tzv. správců hesel, tedy programů, které pomáhají uživatelům vyrovnat se se stále rostoucí potřebou mít velké množství silných hesel do stejně tak rychle rostoucího počtu různých aplikací, stránek, formulářů apod. Konkrétně zde vyzkoušíme jednoduchý špehovací kód použít proti několika různým typům správců hesel a porovnáme výsledky jeho špehování.

Abstract

This bachelor thesis deals with effectivity of spyware against programs from the category of so called password managers, which are programs, which help the users to catch up with growing need of having large amount of strong passwords for also quickly growing amount of various applications, webpages, forms etc. Specifically, we will test using a simple spying code against several different types of password managers and compare its spies‘ results.

Klíčová slova: škodlivý software, špehovací software, správce hesel, bezpečné heslo

Keywords: malware, spyware, password manager, secure password

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

1. Úvod

Informační technologie jsou dnes již prakticky všudypřítomné. Narážíme na ně doma, v práci, ve škole, ve veřejných institucích, v dopravních prostředcích, na úřadech. Stále více zcela běžných denních činností je již částečně či úplně v jejich režii, ať už jde o nakupování, administrativu, poštovní a bankovní záležitosti nebo komunikace s ostatními. Ukládáme na internet obrovské množství svých informací, dokonce i těch nejcitlivějších. Toto logicky přitahuje lidi s nekalými úmysly i do tohoto virtuálního světa, do světa, který nemá žádné hranice, kde lze relativně snadno útočit přes celý svět. Naše informace nejsou tedy ani na Internetu dokonale chráněny. Kyberkriminalita je stále častěji a častěji zmiňována v souvislosti s vydíráním, šikanou, krádežemi, ať už informací, peněz, či dokonce identity, stalkingem čili nebezpečným pronásledováním, spamováním a podobnými aktivitami. Kyberzločinci a jejich protivníci, tedy kyberkriminalisté a vývojáři zaměření na bezpečnost, se neustále předhánějí ve vytváření nových útočných a obranných technik a taktik. Vyvíjí se neustále vedle sebe, každou chvíli vznikají nové druhy útočného software, ať už jde o malware nebo různé skenery, paketové generátory, crackery apod., a také nové strategie útoků, a bezpečnostní specialisté musí neustále tyto nové útoky detekovat a bránit jim.

Tyto útoky, především ty, které mají za cíl získávat informace nebo využívat cizí zdroje, mohou zůstat dlouhé měsíce neodhaleny. Hackeři mohou vyslat specializované programy, které dlouhou dobu nepozorovaně skenují vstupy z klávesnice, pořizují snímky obrazovek, skenují síťový provoz, či dokonce využívají část výkonu pro jiné nekalé účely. Navíc najít na internetu identitu těchto útočníků nebo zdroj malwaru je velmi obtížné, ne-li někdy nemožné. Jednou z cenných komodit, kterou se tito kybernetičtí zločinci snaží získat, jsou hesla. Kromě sociotechnických útoků je získávají pomocí programů kolektivně nazývaných spyware.

Spyware neboli špionážní software jsou škodlivé programy, které mají pravděpodobně nejvyšší nároky na své ukrytí a na perzistenci, protože často musejí zůstat v infikovaném systému po delší čas. Mnoho lidí bezpečnost hesel podceňuje. I přes neustálá varování bezpečnostních expertů stále mnoho lidí používá stále stejná hesla k mnoha účtům, jednoduchá hesla jako „password“, „heslo“, „12345“ apod. Nebo používají krátká a velmi snadno uhodnutelná hesla, jako je například něčí jméno, datum narození, státní poznávací značka auta. A ještě k tomu si ho často napíšou na nějaký snadno dostupný papír, v tom „nejlepším“ případě přilepený viditelně přímo na monitor. A to nejen amatéři, ale dokonce i pracovníci v informatice. Pro tento fakt však mluví skutečnost, že hesla musí být dostatečně složitá, aby se nedala jednoduše uhodnout, z druhé strany však dostatečně dobře zapamatovatelná. Což může být obtížné skloubit, protože jde o celkem protichůdné požadavky.

Výpočetní technika je čím dál výkonnější, což samozřejmě ovlivňuje i rychlost prolamování hesel, takže heslo bezpečné před 30 roky nemusí být bezpečné dnes. Jedním ze způsobů, jak tento problém překonat, jsou programy z kategorie Password Manager, neboli správci hesel. Jsou to programy objevující se ve formě pluginů, ale i samostatných programů, které slouží jako šifrovaná databáze hesel, schopná automaticky doplňovat přihlašovací údaje tam, kam patří. Jsou však tyto programy opravdu bezpečným řešením? Většina běžných uživatelů užívá operační systém MS Windows (1), nebo respektive Android v případě mobilních zařízení. (2) V této práci tedy otestujeme proti jednoduchému spywaru, tedy programu, jehož účelem je shrabovat informace o zařízení a jeho uživatelích, některé z Password Managerů určených pro tyto operační systémy, konkrétně pro operační systém MS Windows.

2. State of the art

Malware je téměř tak starý jako informační technologie samotné a za svou existenci prošel výrazným vývojem. Úplně první program považován za „malware“ je pravděpodobně program The Creeper z roku 1971, který však byl spíše otravný než nebezpečný. (3) (4) Dnes má škodlivý software bezpočet podob, funkcí a může přicházet naprosto odkudkoliv. Z výměnných médií, emailem, jako součást webové stránky a spoustou dalších způsobů se do počítačů mohou dostat všemožné druhy škodlivých programů s nejrůznějšími cíli. Od jednoduchých „organismů“ soupeřících o nadvládu nad strojem se hrozba posunula k programům, které vyřazují z provozu celé korporace.

Dnes není žádné zařízení připojené nebo připojitelné k síti v bezpečí. Evidují se útoky na počítače, mobily i spoustu různých dalších chytrých zařízení, včetně zařízení IoT, tedy internetu věcí. (5) Na prvních příčkách se dnes drží platformy Windows u desktop zařízení a Android u zařízení mobilních, avšak byl evidován i nárůst útoků na zařízení firmy Apple. (6) Malware se již nevyhýbá ani zařízením typu NAS, tedy síťovým úložištím, určeným ke skladování dat. (7)

Existuje bezpočet různých druhů více či méně škodlivých programů. Některé z nich jsou v současné době na ústupu, jako například programy typu miner, tedy programy, které těží kryptoměny, a to často bez vědomí či svolení uživatele. Společnost MalwareBytes zjistila v posledních letech nárůst těchto cryptominerů ve formě drive-by, tedy jako součást webové stránky. S klesající cenou BitCoinu však jejich počet v současnosti mírně klesá. (6)

Vzestup naopak zaznamenávají například vyděračské útoky, jako byl například útok ransomwaru WannaCry v roce 2017, který je největším zdokumentovaným útokem tohoto nebezpečného vyděračského programu v historii. (8) Dalším příkladem mohou být útoky trojských koní, jako je například nedávno vylepšený trojský kůň pro Android zvaný Triada, jehož popis zpracovala společnost Kaspersky. Tento trojský kůň, představitel jedné z nejzastoupenějších tříd malwaru v současnosti (6), je sám o sobě sice starší, ale nedávno se objevil ve vylepšené, modulární verzi schopné útoku na internetové bankovnictví dokonce i přes zabezpečovací SMS, které se staly standardem vícefázového ověření. (9)

Triada vykazuje některé rysy, které lze pravděpodobně očekávat od dalších nových druhů nového malwaru, a sice vysokou modularitu, redundantní ovládání a minimální ukládání souborů. (9) Objevují se i případy použití strojového učení v oblasti malwaru, ačkoli toto se nejspíše Triady přímo netýká. (10) Triada existuje z velké části pouze v paměti a skládá se z několika procesů, které existují nebo neexistují v závislosti na podmínkách a příkazech ovládajících serverů. Je to multifunkční program, který nahrazuje kritické části systému, aby unikl detekci. (9)

Únik detekci je ze všech typů malwaru nejdůležitější pro ty, které musí vydržet dlouho v napadeném systému. To se týká také programů typu spyware, také známý jako stalkerware. (11) Spyware podle společnosti Kaspersky zaznamenal rovněž významný nárůst během roku 2019 (12), a to hlavně na mobilních zařízeních. (13) Příkladem může být například spyware zvaný FinSpy, který je určený rovněž pro mobilní telefony. Tento spyware je podobně jako bankovní trojský kůň Triada složen z několika různých modulů, z nichž každý má svůj specifický úkol. (14) Objevují se i další zprávy o nových anebo morfovaných typech malwaru, například ransomware KeyPass (15) nebo spyware Pegasus. (16) Podle záznamů společnosti Positive Technologies se převažující většina útoků spywaru, které tvořily cca 25 % všech útoků malwaru, zaměřovala hlavně na přihlašovací údaje, platební informace a osobní údaje. (17)

Útoky spywaru byly detekovány proti mobilním zařízením, jako výše zmíněný FinSpy, ale i na desktop zařízení, mezi nimiž se objevují se útoky primárně na Windows, pravděpodobně proto, že oproti Linuxu mají vyšší podíl méně zkušených uživatelů. A sice hlavně skrze makra balíku MS Office anebo PowerShell. Poslední jmenovaný je velice nebezpečný, protože je neoddělitelnou součástí každého Windowsu a jako systémový modul obvykle uniká pozornosti antivirů. (5) (6) Častým cílem malwaru se v současnosti stávají také již zmíněná zařízení IoT, tedy různé chytré elektrospotřebiče, vozidla, senzory a bezpočet dalších zařízení schopných připojení k síti, které se nedají označit jako počítač, mobil nebo tablet. (5) (6)

Tato zařízení společně se zařízeními mobilními mohou obsahovat o svých vlastnících a uživatelích obrovské množství informací, od osobních poznámek, hesel, a telefonních čísel až po údaje o zdravotním stavu. (16) (18) (19)Tyto informace jsou velmi cennou komoditou a na černém trhu mají velkou cenu. Dnešní hackeři jsou tedy často motivování spíše finančně, než aby se snažili o slávu nebo uznání. To podporuje i fakt, že existují frameworky jako MetaSploit a různé další nástroje, které umožňují i člověku bez výrazných znalostí o informatice vytvářet nebezpečný malware. A tím malwarem, který je určený k získávání informací, je právě spyware. Spyware se časem změnil ve velké nebezpečí pro informační systémy a přitáhl pozornost i specialistů v oboru, kteří začali analyzovat detekci spywaru a jeho chování. (20) (21) (22) (23) (24) (25) (26)Já jsem se rozhodl ve své práci také věnovat tomuto druhu malwaru. Používám specializovaný program pro správu hesel a zajímá mě, jak dobře mě vlastně dokáže ochránit před špehováním ze strany kybernetických útočníků.

3. Co je hrozba?

Kromě různých sociotechnických útoků, tedy „obelhávání“ lidí a různých skenů a přímých „hacků“ je velkou hrozbou v kyberprostoru malware. Slovo „malware“ je spojení slov „malicious“, tedy „škodlivý“ a „software“, tedy „program“. Neustále vznikají nové a nové, různých druhů a s různými funkcemi. Jedno je však vždy spojuje: vyvíjí nějakou negativní aktivitu, tj. kradou data, ničí data, mění data, šifrují data, kradou výkon anebo třeba instalují další malware. Také mohou například získat nad infikovaným systémem kontrolu. A většinou to dělají na pozadí, co nejskrytěji, aby pokud možno vydržely škodit co nejdéle. Pojem „infekce“ možná spíše připomíná medicínu. Nicméně právem – pojem „malware“ je často zaměňován s pojmem „počítačový virus“. To sice není totéž, počítačový virus je pouze konkrétní skupina malwaru, je však pravdou, že počítačové viry svým chováním vůči počítačům napodobují chování virů biologických vůči živým organismům. Malware však obsahuje mnoho dalších kategorií členění podle různých kritérií. (27)

3.1. Dělení dle způsobu šíření

Aby mohl jakýkoliv malware být úspěšný, musí se nejprve dostat na hostitelský stroj, tedy na místo, kde má škodit. Může tak činit s různou mírou autonomie, od programů šířených nevědomě nezkušenými uživateli a jejich stroji, až po plně autonomní sofistikované programy, které si samy volí a infikují svůj cíl. Vzhledem k obtížnosti překladu některých názvů jsem se rozhodl používat původní anglické názvy.

3.1.1. Virus

Počítačové viry jsou běžně zaměňovány se slovem malware. Faktem nicméně je, že počítačové viry jsou elektronickou kopií virů biologických. Biologický virus je nebuněčný organismus (tedy není buňkou), který nedokáže žít nebo množit se sám, nýbrž k tomu vyžaduje buňku cizího organismu. Jeho životní cyklus vypadá tak, že napadne hostitelskou buňku, přičemž ji může poškodit, a začne ji používat jako prostředek k vytvoření svých kopií. Počítačový virus se chová analogicky: jedná se o kód, který není samostatným souborem. Tak jako biologický virus napadá buňky, napadá počítačový virus počítačové soubory a používá je ke svému „rozmnožování“, tedy replikaci. Počítačový virus při spuštění infikovaného souboru napadne další soubory a stane se jejich součástí, popřípadě je úplně přepíše. Pokud jsou tyto soubory spuštěny, dojde k další replikaci atd.

3.1.2. Worm

Worm neboli počítačový červ je svým chováním velmi podobný počítačovému viru, ale na rozdíl od něj se jedná o samostatný soubor schopný autonomního šíření, a obvykle tak činí přes síť, čímž je obvykle o něco nebezpečnější, protože virus se na rozdíl od červa sám od sebe po síti šířit nemusí, a také červ nepotřebuje hostitele.

3.1.3. Trojan

Trojan nebo česky trojský kůň je druh malwaru, který se šíří za pomoci nezkušenosti nebo nepozornosti uživatelů. Trojský kůň získal své jméno podle dřevěného koně ze známého řeckého mýtu, který se tvářil jako dar, jako pozitivní věc, vůči svému cíli. Uvnitř však byli skrytí vojáci, tedy měl negativní obsah. Trojský kůň v oblasti malwaru se chová podobně: tváří se jako legitimní soubor, dokonce může mít legitimní funkcionalitu, tváří se tedy pozitivně – současně ale na pozadí vykonává nějaký škodlivý kód, a má tedy také negativní obsah. To sice může splňovat i počítačový virus. Zásadní rozdíl však je, že trojský kůň není schopen přímé sebereplikace. Může se však replikovat nepřímo, protože může např. generovat emaily se svojí kopií jako součást svého payloadu, tedy svého škodlivého kódu. Za jeho šířením jinak stojí obvykle emaily s napadenými přílohami, napadené odkazy a různé podobné věci, které se snaží uživatele přimět k tomu, aby na ně klikli. Zde přichází na řadu lidský faktor. Napadené

odkazy vedou např. na nějakou regulérní stránku, ale mezi tím je nějaké přesměrování, které např. odposlouchává provoz, nebo vedou na kopii regulérní stránky, která kromě své běžné funkcionality vykonává i jinou, škodlivou činnost. Nebo může hacker napadnout regulérní stránku, na které se něco stahuje, a stahovaný soubor nahradí svým infikovaným.

3.1.4. Drive-By

Útok drive-by malwaru probíhá přímo na webové stránce, obvykle je tedy třeba podvrhnout stránku nebo její část. V posledních letech byl zaznamenán nárůst počtu drive-by programů zabývajících se těžbou kryptoměn. Toto souviselo s nárůstem ceny BitCoinu. Drive-By malware je velmi „tichý“ kód, který vlastně nepotřebuje, aby bylo něco staženo. Stačí zobrazit stránku a malware napsaný obvykle v JavaScriptu začne fungovat.

3.2. Dělení dle payloadu

Payload neboli „nálož“, jednoduše obsah, je obecně datová část zprávy, programu apod. V případě malwaru se jedná o efektivní část jeho kódu, tedy tu, která vykonává jeho škodlivou činnost.

3.2.1. Ransomware

Ransomware je jedním z trendů v oblasti malwaru moderní doby. Jedním z typických příkladů útoku ransomwaru byl červ WannaCry. Ransomware se vyznačuje tím, že zašifruje data napadeného systému a za jeho odšifrování požaduje obvykle finanční částku. Nicméně není nikdy jisté, zda skutečně k odšifrování dojde. Často bývá jedinou možností, jak se jej zbavit, kompletní přeinstalace napadeného systému. Proto je velmi důležité pravidelně zálohovat data.

3.2.2. Miner

Minery, obecněji známé jako software na těžbu kryptoměn, mohou být i legitimním softwarem, ale existují i programy, které patří mezi malware, a sice proto, že těží kryptoměny na infikovaných strojích a tím fakticky kradou jeho výkon.

3.2.3. Adware

Adware je tak trochu specifický druh softwaru. Malware je v nejjednodušší verzi definovatelný jako program, který provádí škodlivou či nebezpečnou činnost. Je však adware škodlivý? To je otázka. Adware, jehož název je složením „advertisement“ (reklama) a „software“ (program), není totiž nic jiného než program, který uživateli neustále nabízí různé reklamy, čímž mu znepříjemňuje práci. Takže adware je sice hrozně otravná věc, ale je skutečně nebezpečný? To je diskutabilní.

3.2.4. Exploit

Exploity jsou programy, které nejsou závislé tak moc na nezkušenosti uživatele napadeného systému. Malware fungující jako exploit potřebuje k infekci cílového systému často jediné: neaktualizovaný systém nebo i jen konkrétní program. Nepotřebuje další kliknutí uživatelem nebo něco podobného, obvykle bývá také náročný na technické znalosti autora. Exploit je tedy program, který je speciálně napsán tak, aby zneužil konkrétní zranitelnost konkrétního systému, programu, modulu, nebo dokonce i jedné konkrétní verze dříve zmíněných komponent, a získal tak nějakou výhodu. To obvykle znamená získání zvýšených oprávnění (root, administrátor), otevření vzdáleného přístupu pro útočníka anebo pád služby (DOS/DDOS útoky). Jinak řečeno exploit je program, který využívá nějakou nezamýšlenou funkcionalitu jiného softwaru. Termín „exploitace“ je použitelný jako synonymum ke slovu „hacking“ v přímém smyslu útoku na stroj, s minimálním nebo žádným použitím lidského faktoru na straně cíle.

3.2.5. Backdoor

Backdoory jsou programy, které otevírají v napadeném systému vstupní bod pro útočníka, aby mohl vzdáleně ovládat napadený systém, popř. třeba i jen svůj malware. Většinou se jedná o otevřený síťový port anebo program, který naslouchá na běžně otevřeném portu (80/443 u webserveru, 20/21 FTP serveru apod.).

3.2.6. Destruktivní malware

Destrukce dat byla pravděpodobně jedna z prvních funkcí opravdu nebezpečného malwaru. DO této kategorie se dá zařadit také některý ransomware, konkrétně ten, který po zaplacení neodšifruje data. Obecně sem však patří každý malware, který má za cíl své škodlivé činnosti zlikvidovat nějaká data. Tyto programy bývají z malwaru nejjednodušší na tvorbu, protože jejich payload se může skládat i jen z jediného příkazu příkazové řádky.

3.2.7. Rootkit

Rootkit je takový program, který pomocí modifikace napadeného programu/systému skrývá svou přítomnost. Tím ztěžuje uživateli, aby jej lokalizoval a odstranil.

3.2.8. Dropper

Dropper je škodlivý software, který sám o sobě nemá žádnou nebezpečnou funkcionalitu, ale vyznačuje se tím, že do systému instaluje další malware.

3.2.9. Spyware

Spyware je také jedním z nejnebezpečnějších programů moderní doby, protože data jsou dnes velice cenná věc. Spyware je spolu s cryptominery a backdoory kategorií malwaru, která se musí umět dobře skrýt, aby vůbec k něčemu byl. Spyware je název vzniklý od slova „spy“ tedy špehovat. A přesně to tento druh malwaru dělá: špehuje, shrabuje informace. Speciální kategorií spywaru jsou keyloggery, programy, které zachytávají stisky kláves. Existují dokonce i jako hardwarové moduly, které se dají připojit mezi klávesnici a počítač. Nebezpečnost spywaru ale tímto nekončí: schopný spyware může monitorovat obrázky plochy, odposlouchávat síťový provoz, přeposílat útočníkovi soubory z napadeného stroje a různá další data. Ještě větší sílu má spyware mířený na mobilní telefony, do kterých lidé dávají podstatně víc informací o sobě skrze různé aplikací typu fitness aplikace, do kterých uživatelé zadávají údaje jako výška, váha, které měří tep, zjišťují lokaci atd. Nebo mohou využít senzory mobilu a snímat tak své okolí v takovém rozsahu, že po dostatečně dlouhé činnosti mohou třeba zmapovat byt oběti. (18) (19)V počítači, který je položený na stole, je tato funkcionalita omezena. Počítač však může být zdrojem jiných důležitých dat – a sice hesel. Do mobilu lidé často zadají své heslo jednou a nechají zařízení, aby si zapamatovalo přihlášení. I toto spojení lze samozřejmě napadnout a přerušit. Na počítači však lidé spíše heslo zadají. Ti nejméně zkušení dokonce nechají napsané heslo na viditelném místě, „nejlépe“ přímo na papírku na monitoru. Paradoxně tam se spyware v počítači pochopitelně nedostane, leda by na papír mířila k počítači připojená kamera. Spyware na počítači se tedy zaměřuje na hesla, která jsou do počítače zadávána nebo jsou v něm uložena. Nejzákladnější obranou proti spywaru, nebo obecně proti každému malwaru, je kromě obezřetnosti spočívající v tom, že uživatel nekliká bez rozmyslu na všechno, co vidí, hlavně v emailu, také udržování veškeré softwarové výbavy aktuální, hlavně operačního systému a antivirových, antispywarových a ostatních anti- programů. Když už je ale spyware uvnitř, je pro něj snadné zachytávat stisky kláves a snímky obrazovky. Tím vyvstává otázka, nakolik je vlastně nutné mít bezpečné heslo. (28)

4. Bezpečné heslo

Je Vaše heslo bezpečné? Dnes možná, otázka je, zda bude v budoucnu. Mnoho lidí myslí způsobem „heslo musí být co nejjednodušší, abych si ho mohl pamatovat“ anebo si ho někam napíše. To způsobuje, že na vrcholu žebříčku nejužívanějších hesel se neustále opakují hesla jako 12345, password, heslo, 123456, qwertz, 111111, iloveyou, qwerty, letmein, secret, admin, abc123 atd. (29) (30) (31) Anebo se jedná o jméno, datum narození, nebo jinou podobnou informaci. Každopádně jsou to jedna z prvních hesel, která každý útočník otestuje jako první, a pokud to udělá strojově, prolomí heslo velmi rychle. Bezpečné heslo by mělo v první řadě obsahovat znaky ze čtyř znakových sad: malá písmena, velká písmena, číslice a speciální znaky, jako je otazník, vykřičník, procento, matematické symboly apod. To by mohlo svádět na heslo typu Password123#.

Nicméně ani toto ještě není bezpečné. Toto heslo totiž zaprvé stále připomíná existující slovo a zadruhé obsahuje postupnou řadu číslic. I toto může představovat potenciální riziko, protože útočníci obvykle nejsou úplní hlupáci a používají také tzv. slovníkový útok. Slovníkový útok je generování hesel obsahujících nějaké konkrétní existující slovo nebo logickou posloupnost (číslic apod.). Dobrou obranou proti slovníkovému útoku je také nepoužívání jazykově specifických znaků, jako jsou například písmena s diakritikou. Jejich užití naopak paradoxně zvětšují počet možností, které by musel otestovat útočník užívající útok brutální sílou, tedy generování všech možných hesel o určité délce z určitého rozsahu znaků. Ne právě bezpečné je také umístění speciálního znaku na konec nebo začátek hesla. Konečně, heslo by v dnešní době mělo být alespoň 16 znaků dlouhé. Před pár desítkami let mohlo být bezpečné třeba i jen osmimístné heslo. Za dalších deset let nemusí být třeba třicetiznakové heslo dost bezpečné – kdoví. Toto je dáno rostoucí výpočetní silou současné výpočetní techniky.

Příklad tvorby bezpečné hesla by tedy aktuálně mohl vypadat nějak takto:

Vezmu si obyčejnou větu „Studuju vysokou v Ostravě“.

Odstraním jazykově specifické znaky: „Studuju vysokou v Ostrave“.

Přidám pár dalších velkých písmen, a to tak, že jím začnu každé slovo: „Studuju Vysokou V Ostrave“. Rozdělím každé slovo na dvě poloviny, první polovinu převrátím a druhou polovinu smažu (všechna slova mají lichý počet písmen, takže první polovina bude ta delší: „Studuju Vysokou V Ostrave“ → „dutS osyV V rtsO“.

Potřebujeme dostat do hesla také nějaká čísla. Nejsnazší je nahradit číslicemi písmena, která je připomínají, takže například 0 za O, 4 za A, 6 za b, 2 za S apod: „dut2 0syV V rts0“.

Posledním krokem je dostat do hesla nějaké speciální znaky. To jsem už udělal čistě náhodným nahrazením mezer různými znaky: „dut2#0syV&V@rts0“.

Toto byl trochu extrémní příklad, ale byla to ukázka, jak se dají logikou vytvářet celkem dobře zapamatovatelná hesla. Nicméně běžní uživatelé mají často problém si i zapamatovat heslo připomínající svou délkou i znakovou skladbou státní poznávací značku. Jak tento problém vyřešit? Zde přichází na řadu speciální programy spadající do kategorie „password manager“ neboli správce hesel.

5. Password manager

Existuje několik druhů password managerů neboli správců hesel. Některé z nich, jako například KeePass, vytvářejí zašifrovaný soubor, který je fakticky databází hesel, ale navenek vypadá jako normální soubor, který má svou příponu, dá se kopírovat, vložit, přejmenovat. Tento program umožňuje skrze uživatelem zvolenou klávesovou zkratku vložit uživatelské jméno a heslo do aktuální pozice kurzoru. Jiné password manager programy existují jako pluginy, jako např LastPass. Vytváří podobnou databázi v prohlížeči. Naopak například PwdHash, starší program, který už dnes není aktivně vyvíjen, se chová tak, že použije uživatelem zadané heslo a doménové jméno cílové stránky k vygenerování nového, mnohem silnějšího hesla, a ve skutečnosti použije to. V této práci otestuji některé ze zdarma šířených programů pro správu hesel a jejich odolnost proti aktivnímu základnímu spywaru.

5.1. KeePass

KeePass je opensource program pro správu hesel, primárně vyvíjený jako desktop aplikace pro platformu Windows, ale není na ni omezen. Je neoficiálně portován pod různými názvy i na spoustu dalších platforem, včetně operačních systémů mobilních zařízení, zařízení firmy Apple, dokonce i na poněkud vzácné systémy typu BlackBerry, Chrome OS, Palm OS a konečně existují i porty do CLI verze nebo do verze hned několika rozšíření pro prohlížeč. KeePass tvoří databázi ve formátu KDB, resp. verze 2 ve formátu KDBX. Tato databáze je soubor zašifrovaný algoritmem AES-256 a Twofish (verze 1) resp. ChaCha20 (verze 2). Jeho integritu navíc pojišťuje SHA-256 checksum plaintextové verze databáze ve verzi 1 resp. HMAC-SHA-256 zašifrované verze databáze, dochází tedy k šifrování a současně k hashování. Přístup k databázi lze ochránit několika způsoby: Zaprvé je tu možnost ochránit přístup pomocí Master hesla. To je sice pořád heslo, ale přinejmenším si uživatel musí pamatovat jedno heslo a ne desítky. Je velmi důležité si jej pamatovat, protože KeePass nepodporuje vůbec žádnou možnost jeho obnovení. Ztratíte-li Master heslo, databáze je pro vás ztracena, a to navždy. Master heslo si KeePass pamatuje jen jako jeho SHA-256 hash.

Druhou možností je používat tak řečený klíčový soubor. Jako klíčový soubor může sloužit zcela jakýkoliv soubor. Na rozdíl od hesla, kdy uživatel musí něco znát, v tomto případě uživatel musí něco mít. To může pro někoho být jednodušší. Důležitý je v tomto případě obsah souboru, to on je tou tajnou informací. Proto logicky tento soubor nesmí být přepsán, jinak je přístup k databázi ztracen. Stejně jako když zapomenete heslo, není ani zde žádná možnost, jak přístup k databázi obnovit – leda když znáte na bit přesně obsah souboru a jste schopni ho regenerovat. Tento soubor by však neměl být úplně tak snadno zapamatovatelný. V podstatě se jedná o ekvivalent autentizace pomocí USB klíčenky nebo podobného zařízení – přihlašovací klíč je příliš velký a složitý, než aby si jej uživatel zapamatoval, ale pamatuje si ho zařízení, které uživatel má u sebe. KeePass umožňuje přímo i tuto funkci: užití souboru, který je na USB. Není pro něj vůbec důležité, kde je soubor. Důležité je jen, co je v něm, přesněji řečeno zajímá ho hash souboru.

Jako třetí autentizační možnost je propojení s Windows účtem. V tomto módu KeePass poskytuje přístup k databázi tehdy, pokud je uživatel přihlášen pod správným profilem. Není to závislost na hesle, KeePass bere v potaz celý uživatelský profil. To znamená, že pokud je účet z nějakého důvodu poškozen, uzamčen nebo zničen, nestačí jen vytvořit profil se stejným jménem a heslem. Heslo Windows účtu jako takové KeePass nezajímá. Abyste se dostali k takto chráněné databázi KeePassu, museli byste zkopírovat celý účet se vším, co k němu patří, to znamená s ID, přístupovými právy, typem účtu, přístupovými údaji apod. Toto jsou tři možnosti, jak přistupovat k databázi KeePassu. To ale ještě není všechno: KeePass totiž umožňuje tyto přístupové cesty kombinovat. To znamená, že například můžete nastavit přístup přes Master heslo a klíčový soubor. V takovém případě musíte při autentizaci

poskytnout oba uvedené údaje, takže se vlastně jedná o dvoufaktorovou autentizaci. Třífaktorovou, pokud uživatel použije všechny tři možnosti autentizace. To však podporuje pouze KeePass verze 2. x v prostředí Windows.

Z uživatelského pohledu KeePass vloží uživatelské jméno a/nebo heslo tam, kde je aktuálně kurzor, na povel klávesové zkratky. Pokud pro danou doménu nebo okno existuje více přihlašovacích údajů, dá KeePass na výběr. To znamená, že může vložit přístupové údaje naprosto do čehokoliv, co akceptuje vstup textového typu. To může znamenat potenciální bezpečnostní riziko.

5.2. LastPass

LastPass se pravidelně umisťuje na různých žebříčcích TOP password managerů. (32) (33) (34) Na rozdíl od KeePassu jde o prohlížečový plugin, který ale stále ukládá hesla hlavně lokálně. Ukládá je mezi daty prohlížeče a dohledat je zvenku je poněkud obtížnější, protože nepoužívá člověkem snadno čitelné názvy. K zašifrování hesel používá algoritmus AES-256 a jejich integritu zajišťuje PBKDF2 algoritmus spolu s SHA-256.

LastPass defaultně používá SHA-256 chráněné Master heslo. Na rozdíl od KeePassu má schopnost jeho obnovy v případě ztráty. Není to ale jeho jediná možnost. Umožnuje používat jednorázové kódy odeslané na mobilní telefon nebo potvrzení přes aplikaci. LastPass je v kontaktu s celou plejádou podobných aplikací, které mu umožnují toto realizovat. Dále dokáže používat i autentizaci skrze USB klíčenku skrze aplikaci Yubico, což je aplikace instalovaná na USB, která vygeneruje jednorázové heslo navíc k Master heslu. A konečně, LastPass podporuje i autentizaci pomocí biometrických údajů či identifikační karty. Poslední tři možnosti jsou však dostupné pouze pro Premium verzi.

Free verze může kromě kódů do mobilu použít ještě tzv. grid autentizaci. Ta spočívá v tom, že program vygeneruje tabulku písmen a číslic oindexovanou na způsob známé hry „lodě“, tedy řádky jsou označené čísly a sloupce písmeny. LastPass pak při přihlašování vygeneruje pokaždé 4 sety souřadnic a uživatel musí podle těchto souřadnic doplnit znaky z tabulky, které se na daných souřadnicích nacházejí. Samotná tabulka zůstává pro jednoho uživatele stejná, ale program nejpozději po 100 přihlášeních vynutí její regeneraci. Tabulku lze zobrazit na obrazovce, stáhnout ve formě CSV nebo přímo tisknout. Nakonec LastPass se může synchronizovat s existujícími cloud drive službami, jako je např Google Drive nebo Dropbox. Premium verze má dokonce svůj cloud. (35)Proti tomu KeePass něco takového nedokáže, ale protože to je desktop aplikace, vytváří standardní soubor, který se tam dá nahrát.

5.3. DashLane

DashLane je konkurent LastPassu původně míněný pro enterprise sektor a funkcionálně je mu velmi blízký. I on pravidelně figuruje na vysokých místech na žebříčcích. (32) (33) (34) Funguje jako desktopová aplikace napojená na plugin a skladuje data lokálně s cloudovou synchronizací. DashLane disponuje plně svým vlastním cloudem. I on šifruje hesla algoritmem AES-256. Svou databázi chrání klasicky Master heslem, ale umožňuje ve všech verzích aktivovat i druhou úroveň autentizace, a sice jednorázové heslo.

Podobně jako LastPass monitoruje webové stránky a když detekuje nějaký přihlašovací formulář, zobrazí v něm svou ikonu. Kliknutí na ni zobrazí seznam všech již uložených přihlašovacích údajů pro tuto doménu. Zde uživatel vybere konkrétní kombinaci přihlašovacích údajů a plugin je přímo vloží do správných polí. Oba pluginy jsou si funkcionálně velmi podobné, co se týče samotného vkládání a ukládání hesel. Síla DashLanu leží spíše v jeho specializaci na enterprise prostředí. Umožňuje různá zabezpečení sdílení přihlašovacích údajů, například skupinové heslo do nějaké pracovní skupiny apod.

Jeho bezplatná verze je docela limitovaná, například umožňuje uložení pouze 50 přihlašovacích údajů, umožňuje používat jen 1 zařízení. Tj. nepoužívá cloud, sdílet může jen s 5 dalšími účty apod. Naproti tomu Premium placená verze pravidelně skenuje uživatelská hesla proti údajům o únicích hesel a údajně prohledává také DarkWeb. (36)

Také používá svou vlastní VPN. Proti tomu LastPass většinu z těchto funkcí podporuje už ve Free verzi, Premium verze navyšuje funkce jen o live support, sdílení údajů a přidané možnosti autentizace (viz výše). LastPass nedisponuje VPN, naproti tomu však má malé šifrované úložiště na soubory.

5.4. PwdHash

PwdHash je mrtvý projekt, který se choval zcela odlišně oproti ostatním zmíněným password managerům. Uvádím jej zde pro úplnost, protože si myslím, že vzhledem ke svému chování za to stojí, ale testován nebude, protože není se současnými prohlížeči kompatibilní, a tudíž se mi jej nepodařilo spustit. Projekt byl zastaven zřejmě v roce 2009 nebo 2010, protože dle oficiálních stránek projektu je původně kompatibilní pouze s prohlížečem Mozilla Firefox 3. 5 a starší. Podle reportů některých uživatelů však Firefox plugin stejně podporoval až někde do verze 40. x, kde definitivně Heslo zachyceno na nějaké zásadní změně v prohlížeči týkajícím se paralelizace jeho chodu. Ačkoli existují i porty do prohlížečů Internet Explorer 7 a starší a Opera verze cca 11 a starší, nepodařilo se mi jej rozchodit. Pro srovnání, v době psaní této práce má prohlížeč Firefox aktuální verzi 74, Opera 67 a Internet Explorer dokonce jako projekt úplně umřel a byl nahrazen prohlížečem MS Edge, o jehož kompatibilitě s pluginem PwdHash jsem nenašel žádné informace, ale našel jsem nějaká fóra a skupiny, u kterých si nejsem jistý, jak moc jsou pravá, ale pokud ano, lze z nich usoudit, že plugin byl ve své době poměrně oblíbený a že jednotliví vývojáři z řad uživatelů se snaží projekt obnovit a portovat na nové prohlížeče. Samotný původní projekt je dnes veřejný. Nicméně jeho součástí je i webová stránka schopná generovat hesla. Později se dovíme proč.

PwdHash poskytuje určitou obranu proti podvrženým stránkám. LastPass a DashLane evidují databázi známých podvržení podobným způsobem jako antiviry a také součástí každého jejich přihlašovacího záznamu je URL, které záznam patří. PwdHash na to jde trochu jinak. Ale od začátku: Zadá-li uživatel heslo do formuláře, PwdHash jej za použití cílové domény převede na mnohem silnější heslo a použije toto vygenerované heslo. To jinými slovy znamená, že cílová stránka se uživatelovo heslo nikdy nedoví, doví se jen heslo vygenerované PwdHashem. Protože ale každá mince má dvě strany, plugin samotný uživateli také nijak neřekne, jaké heslo vygeneroval, a tak se uživatel stává na pluginu závislým, bez něj se nepřihlásí. Právě proto vznikla i webová stránka. Zaprvé se uživatel zde může vygenerované heslo dovědět, a zadruhé se s její pomoci může přihlásit i tam, kde nemá plugin. PwdHash potřebuje manuální trigger, aby detekoval heslo. Tímto triggerem je stisknutí klávesy F2 před a po zadání hesla anebo napsání „@@“ před samotné heslo. (37)

Tabulka 1 – Srovnání funkcionality password managerů

	KeePass	**LastPass**	**DashLane**
Cloud	Neumí používat.	Umí se spojit s cizím, Premium verze má svůj.	Má vlastní cloud, ale jen Premium.
Šifrování hesel	AES-256	AES-256	AES-256
Primární ochrana přístupu	Master heslo	Master heslo	Master heslo
Obnova ztraceného přístupu	nemožná	možná	nemožná
Vícefázová ochrana přístupu	Klíčový soubor, sloučení s Windows účtem	Autentizace mobilem, USB klíčenkou, dokonce biometrií	Jednorázové heslo
Primární způsob tvorby záznamu	Manuální zadání přímo do programu	Automaticky čtením z formuláře při prvním zadání údajů	Automaticky čtením z formuláře při prvním zadání údajů
Dovoluje zobrazit Master heslo	Ano	Ano	Ano
Dovoluje zobrazit heslo přihlašovacího záznamu	Ano	Ano	Ano

6. Implementace

Jednoduchý špehovací program bude psán v PowerShellu, skriptovacím jazyce postaveném na platformě .NET. Tento jazyk a jeho interpret je vestavěnou součástí systémů Microsoft Windows od roku 2006. Poslední verze v době psaní tohoto textu je 6.2.2. PowerShell je vhodným adeptem pro škodlivý kód proti platformě Windows, protože vzhledem k tomu, že to je jeho nativní prostředí, Windows jej pravděpodobně nebude považovat za nevyžádanou akci. PowerShell totiž slouží jako „nový příkazový řádek" této platformy, a existují plány, podle kterých by měl PowerShell v brzké době zcela nahradit standardní příkazový řádek platformy Windows. Je to ekvivalent bashe v Linuxu. A pozor, dokonce i Linux je schopen PowerShell nainstalovat a používat, takže tento kód by potenciálně mohl v Linuxu pracovat. Ve Windowsu je však už delší dobu součástí samotného jádra. Proto by měl uniknout pozornosti Windows Defenderu i různých dalších antivirů a jim podobných programů. Kód využije opět vestavěné vlastnosti Windowsu, a sice Task Scheduler, komponentu, která má na svědomí automatizaci procesů. Spyware vytvoří tři samostatné procesy na pozadí, z nichž každý má na svědomí zachytávání konkrétního informačního zdroje: klávesnice, schránka a monitor. Přísně vzato se jedná pouze o payload, který by měl být dále vložen do nějakého mechanismu, který by ho šířil, například do binárního souboru vyhlížejícího jako regulérní instalační soubor.

6.1. Skener schránky

Jednou z věcí, kterou se na počítači běžného uživatele vyplatí sledovat, je schránka, tedy vestavěná funkce Kopírovat a Vložit. Někteří uživatelé mohou heslo kopírovat ze souboru či password manageru. Můj kód kontinuálně sleduje schránku a v pravidelných intervalech kontroluje, zda má obsah. Nevýhodou tohoto algoritmu je to, že má tendenci zapisovat mnohokrát za sebou jeden a týž obsah schránky. Bylo by bývalo snadné tomu zabránit i bez událostí, ale to by muselo platit, že do schránky lze vložit jen jediný druh dat. Protože tam ale lze vložit soubor, text, obrázek a podle dokumentace i audio stopu, docházelo by k potřebě porovnávat obrázek s textem apod. anebo ke složitému porovnávání, co vlastně bylo ve schránce před pár sekundami, co je tam teď apod. Proto můj skener každých deset vteřin zkontroluje, zda schránka obsahuje text, obrázek, souborový strom anebo audio stopu. Pokud jedno z toho skript ve schránce najde, uloží to do odpovídajícího souboru, s výjimkou souborového stromu. Evidentně se schránkou nepřenáší samotné soubory, ale pouze cesty. I to ale může být pro hackera užitečná informace. Kód je primárně tvořen funkcí SaveClip.

- **Funkce SaveClip** je jádrem skriptu na skenování schránky. Jako argument tato funkce požaduje cestu ke složce, do které má ukládat výsledky své aktivity. Tato cesta by v ideálním případě měla vést na vzdálený server. Já vzdálený server nemám, proto jen pro demonstraci ukládám data do lokální složky. Jak jsem již uvedl, funkce nejprve zkontroluje, zda schránka obsahuje text, obrázek, zvukovou stopu anebo seznam souborů. V dalším průběhu se funkce rozvětví podle toho, zda našla nějaký obsah ve schránce. Pokud neexistuje cílový adresář, stvoří ho a skryje jej před uživatelem. Pak exportuje případný nalezený text do textového souboru, obrázek uloží ve formátu jpg, zvuk ve formátu wma a seznam souborů zapíše opět do textového souboru, ale jiného než toho, kam zapisuje kopie textu.

Hlavní tělo tohoto skeneru tvoří jen nekonečný cyklus, který v pravidelných intervalech volá funkci SaveClip. Toto by bylo možné rozšířit na event-driven program, který místo pravidelného intervalu bude reagovat na změnu obsahu.

6.2. Skener monitoru

Pro skenování obrazovky jsem zvolil postup, kdy kontroluji název aktuálně aktivního okna. Skener neustále kontroluje, který proces právě drží „fokus“, tedy aktuálně používané okno, a zabývá se názvem samotného okna. Protože nesleduje změnu procesu, ale změnu názvu okna, dokáže detekovat i akce jako je překliknutí na jiný panel prohlížeče, které by detektor na úrovni názvu procesu pravděpodobně nedetekoval, protože by se stále jednalo o proces se stejným názvem. Jednoduše dokáže rozlišit od sebe více oken téhož programu. Jádrem skeneru je funkce SaveScreenshot.

- **Funkce SaveScreenshot** projde všechny připojené monitory a vytvoří pro každý z nich objekt typu System.Drawing.Graphics, do nějž zkopíruje bitmapu s obsahem daného monitoru a pak ji uloží do souboru na definované umístění. Sama o sobě je to celkem jednoduchá funkce.

Skener je tvořen nekonečným cyklem, který neustále kontroluje, zda nedošlo ke změně názvu okna. Před cyklem se načte název okna aktivní v čase spuštění. Pak se nastartuje cyklus, který v každé iteraci srovná aktuálně otevřené okno s tím předchozím. Narazí-li na rozdíl, zapamatuje si okno, aby mohl v další iteraci opět posoudit rozdíl, a aktivuje zachytávání obrazovek. Skener obsahuje seznam řetězců ve formě regulárního výrazu, díky kterým je do jisté míry schopen detekovat, zda otevřené okno je přihlašovací, registrační anebo nějaké podobné okno, které by mohlo obsahovat nějaké přihlašovací nebo třeba i platební údaje. Pro tato okna bude skener mnohem aktivnější, pořídí více snímků a bude je pořizovat vyšší rychlostí. Přepisem této proměnné je možné skript adaptovat na zvýšenou aktivitu pro určitý typ oken.

6.3. Skener klávesnice

Keylogger je třetím modulem špehovacího skriptu. Je implementován v jazyce C#, který je následně volán a ovládán PowerShellem.

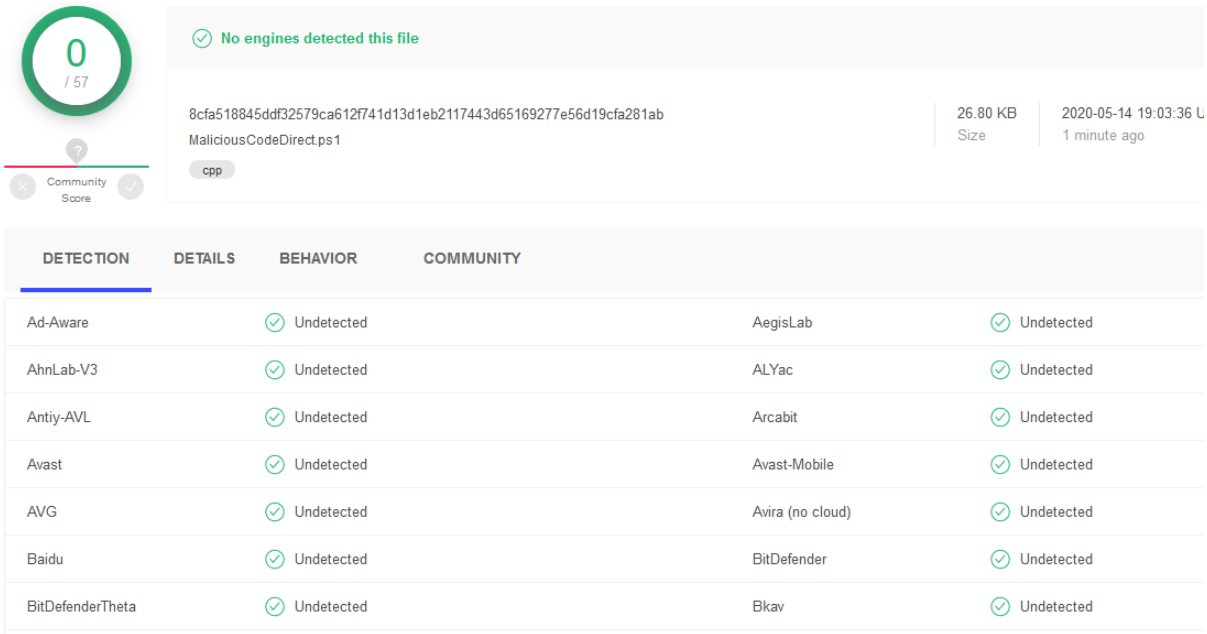
Hlavní funkční část tvoří samotný C# kód, který importuje skupinu metod z knihovny user.dll. Funkční část je tvořená metodou Record, která je zodpovědná za samotný keylogging. Obsahuje téměř nekonečný cyklus, přesněji řečeno cyklus, který zastavuje jen konkrétně definovaná kombinace kláves. Zde konkrétně kombinace CTRL+SHIFT+ALT. Stručný popis jednotlivých kroků v cyklu:

- Nejprve se přečte klávesnice pomocí metody GetAsyncKeyState. Async, tedy asynchronní metoda je použita, aby přenos informace o stisku klávesy zablokoval vlákno a zkreslil výsledky. Tato funkce vrací číselnou hodnotu 0x8000, je-li stisknuta klávesa, anebo 0x8001, byla-li klávesa nedávno ve stisknutém stavu. Skener spoléhá na hodnotu 0x8000, protože volání metody GetAsyncKeyState jinou aplikací by mohlo ovlivnit výsledek, kdyby se spoléhal na 0x8001.
- Pokud má klávesa hodnotu 0x8000, nastaví se jednotlivé prvky pole stavů kláves na 0xFF (stisknuto) nebo 0x00 (nestisknuto). Tím se fakticky zbaví čísel a v podstatě převede číselné konstanty na pole logických hodnot.
- Kód zkontroluje, zda se jednalo o stisk a ne držení. Porovná aktuální pole stavů proti poli stavů z předchozí iterace, kde jej zajímají pouze klávesy, které mají nyní hodnotu 0xFF a které současně v předchozí iteraci měly hodnotu 0x00.
- Aby keylogger mohl zaznamenávat stisky všech kláves, a ne pouze alfanumerických znaků, musí proběhnout ještě konverze netisknutelných znaků a stisků kláves jako CTRL, SHIFT, ALT, DELETE, WIN, Num Lock, Caps Lock, tabulátor apod. To je zařízeno pomocí switche, který jednotlivé netisknutelné klávesy nahradí tisknutelným textem, tedy vypíše název této klávesy uzavřený ve složených závorkách.

- Tisknutelné klávesy jsou přeloženy pomocí metody ToUnicodeEx. Ta na rozdíl od velmi podobné metody ToUnicode dokáže rozlišovat rozložení klávesnice a zapojovat jej do překladu. Metoda ToUnicodeEx přeloží stisky kláves s tisknutelnými znaky z jejich virtuálních kódů na samotné znaky.
- Záměrně jsem nechal nahrazovat i klávesy se speciálními znaky jejich názvy, protože jsem zjistil, že některé z těchto znaků při aktivním keyloggeru nejdou napsat, což by mohlo vzbudit podezření. Primárně se jedná o symboly ´ a ˇ, ale nevylučuji, že by se to mohlo týkat i jiných znaků, které mě nenapadlo otestovat.
- Pokud byl zaznamenán znak nebo název klávesy, vypíše se tento znak nebo název klávesy do souboru definovaného parametrem funkce. Pokud je to třeba, připíše pak ještě i nový řádek. To, po které klávese se má přidat nový znak, jsem nevolil podle nějakého specifického klíče, jen jsem vypnul odřádkování za známými klávesami typu numpad klávesy, CTRL, ALT, DEL, WIN apod., u kterých jsem neviděl důvod řádkovat.

6.4. Hlavní kód

V předchozích částech jsme si popsali jednotlivé části špehovacího kódu. Jsou to však stále ještě jen definice funkcionality, ještě je nutné vyřešit jejich spouštění. K tomu využiji dříve zmíněnou vestavěnou funkci systému Microsoft Windows, a sice Task Scheduler. Hlavní skript použije vestavěný příkaz Register-ScheduledJob s parametry povolujícími spouštění i při napájení z baterie a bez sítě a nastaví spouštění při startu systému. Současně jej skryje před Task Managerem pomocí dalšího parametru. Pokud toto selže, pokusí se alespoň zaregistrovat procesy tak, aby se spustily každý den krátce po hodině a minutě prvního spuštění skriptu. Použil jsem timeout dvě minuty, abych předešel nespuštění skriptu, dojde-li ke spuštění přesně na přelomu minut. Pro zajímavost jsem také nechal výsledný kód zkontrolovat webem VirusTotal. Žádný z jeho skenerů nepovažoval kód za škodlivý.



Obrázek 1 – Výpis VirusTotalu

7. Popis experimentu

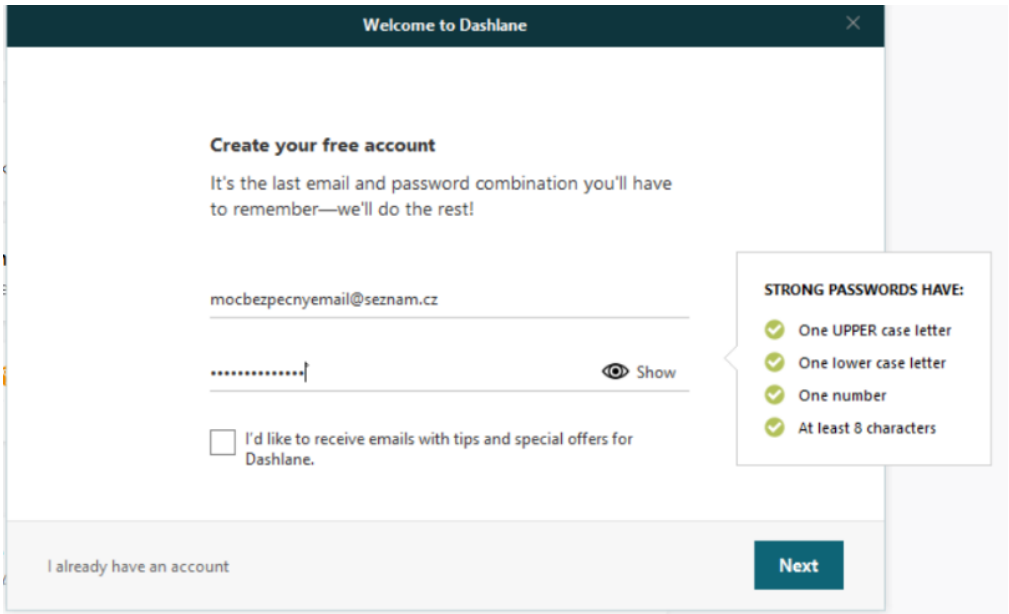
V experimentu provedeném v souvislosti s touto prací provedu několik základních akcí pomocí všech tří password managerů, a to včetně samotné tvorby a užití Master hesel. Bohužel nemůžu otestovat vícefázové autentizace, protože se často jedná o placené funkce. Poté budu analyzovat, jak dobře a pomocí kterého ze tří modulů můj spyware dokázal zachytit alespoň částečně přihlašovací údaje. Dále otestuji schopnost jednotlivých password managerů zabránit odcizení hesla při samotné tvorbě přihlašovacího záznamu a samozřejmě otestuji útok na samotné přihlášení do nějaké stránky pomocí těchto password managerů a provedu ještě několik dalších testů. Testování bude probíhat na platformě Microsoft Windows. Pro každý test nejprve spustím celý skript a pak se pokusím provést danou akci s každým password managerem tak, jak bych očekával od „běžného, nezkušeného uživatele“. Snímky skeneru obrazovky jsou oříznuty na obrázky zachycující pouze podstatné informace. Například okno DashLane je poměrně malé a na záznamu celé obrazovky by nebylo příliš na výřezu vidět.

7.1. Experiment č. 1: Tvorba Master hesla

Master heslo je prvním prvkem zabezpečení každého z testovaných password managerů. LastPass a DashLane se bez něj neobejdou, KeePass ano. Otestuji tedy pouze jej, abych měl proti čemu porovnávat. Master heslo je jediné, které uživatel bezpodmínečně musí zadat ručně anebo zkopírovat, password manager je nedokáže vyplnit sám. V tomto testu se pokusíme simulovat nezkušeného uživatele a vytvořit nový účet nebo databázi s každým z testovaných programů. Cílem testu je pokusit se zachytit Master heslo během jeho prvního zadávání, a to zadávání z klávesnice.

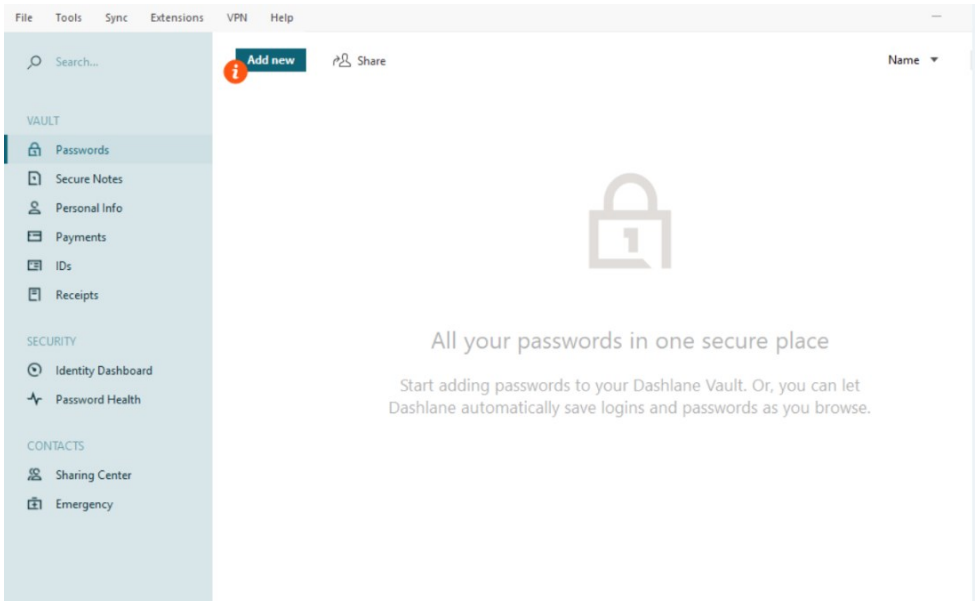
DashLane

Tvorbu Master hesla v DashLane doprovází tvorba samotného účtu. Obrázek 2 zachycuje částečně napsané heslo a celé přihlašovací jméno.



Obrázek 2 – Tvorba Master hesla v DashLane

Obrázek 3 mimo jiné ukazuje, že DashLane informuje uživatele, co by mělo splňovat alespoň trochu bezpečné heslo. Je vidět i tlačítko na zobrazení hesla. To náš „uživatel“ použil, ale následující snímek zachycený skenerem monitoru už ukazuje obrazovku DashLane po přihlášení.



Obrázek 3 – DashLane po přihlášení

Skener monitoru tedy sice odhalil přihlašovací email, ale zobrazení hesla minul. Vyšší frekvence snímkování by však mohla heslo takto zachytit. Nicméně i když skener monitoru zachycení hesla při tvorbě Heslo zachyceno a skener schránky dosud nevytvořil nic, co by se dalo zobrazit, skener klávesnice uspěl:

Výpis 1 – zachycené Master heslo DashLane

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

mocbz{BACKSPACE}

ezpecnyemail{ALT}{LALT}

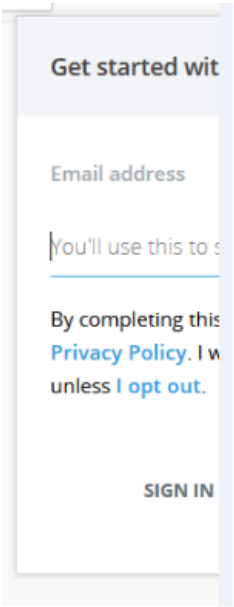
{CTRL}{LCTRL}@seznam{OEM_PERIOD}

cz{TAB}{SHIFT}{LSHIFT}Bezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD1}{LEFTMOUSE}

Tento výpis odhaluje nejen to, že uživatel skutečně napsal přihlašovací email mocbezpecnyemail@seznam.cz, udělal u toho navíc překlep, ale když znovu napíšeme zaznamenanou sekvenci kláves, zjistíme, že „uživatel“ použil heslo „BezpecneHeslo1“ a použil k tomu numerickou klávesnici. Mimo jiné jsme se například dověděli, že uživatel je dost znalý na to, aby věděl, že klávesa tabulátor přepíná mezi vstupy. DashLane tedy nedokázal zabránit zachycení Master hesla při jeho vytváření. Zachytil ho skener klávesnice a při vyšší frekvenci snímkování by jej pravděpodobně zachytil i skener monitoru.

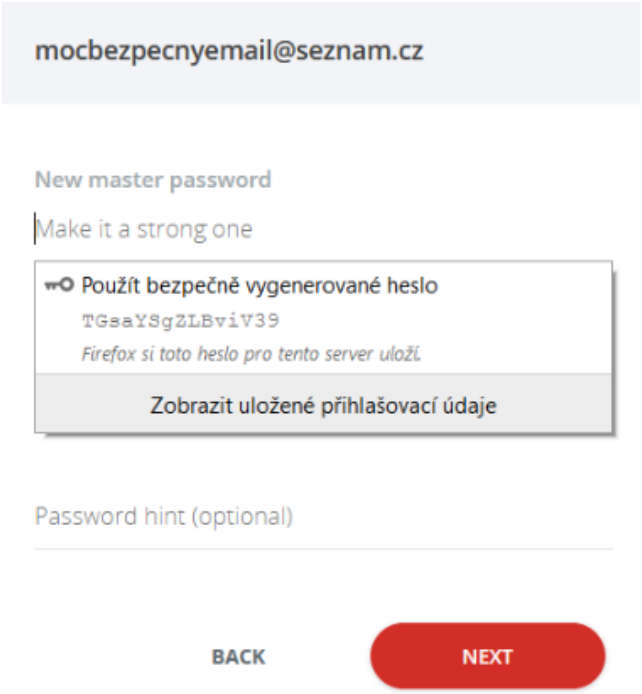
LastPass

LastPass se s DashLanem shoduje v tom, že i u něj je třeba tvořit současně účet. Na obrázku 4 je vidět, že skener monitoru nedetekuje úplně správně rozměr obrazovky a malé okýnko LastPassu zachytil pouze částečně:



Obrázek 4 – Dialog pro volbu uživatelského jména LastPass

Kód čte rozměry připojeného monitoru pomocí nativních vlastností Width a Height objektu System.Windows.Forms.Screen, který používá. Pole pro přihlašovací email je na obrázku prázdné. LastPass netvoří vlastní okno, takže nespustil zrychlené snímkování skeneru monitoru a na dalším snímku již proto vidíme dialog pro tvorbu hesla:

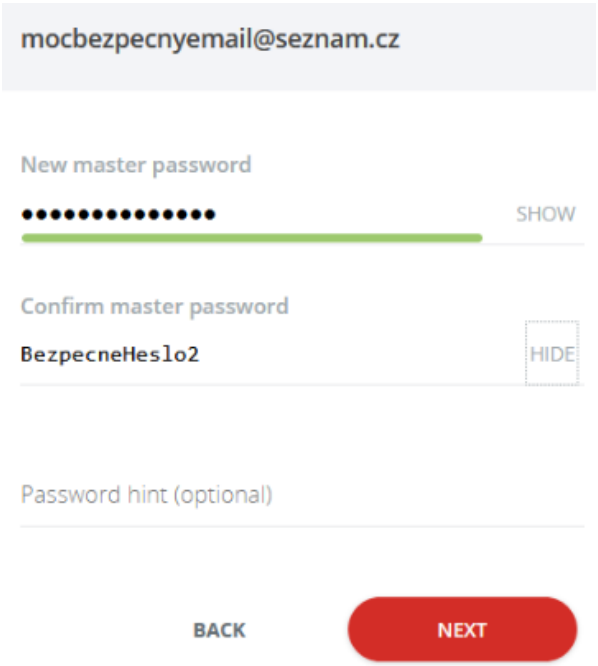


Obrázek 5 – Dialog tvorby hesla LastPassu

Zadávání přihlašovacího jména samo o sobě skeneru monitoru uniklo díky pomalé snímkovací frekvenci. Na tomto výřezu však nahoře vidíme, jaké uživatel zadal. LastPass nabízí uživateli k použití

automaticky vygenerované heslo jako Master heslo. Avšak ho neskryl, a proto jej skener monitoru zachytil. Náš uživatel však není schopen si takové heslo zapamatovat, a použije tedy své.

Zatím to vypadá, že LastPass se brání stejně dobře jako DashLane. Nicméně obrázek 6 ukazuje, že skener měl v tomto případě trochu lepší načasování.



Obrázek 6 – Zobrazené Master heslo LastPassu

Skener klávesnice tyto údaje potvrdil.

Výpis 2 – zachycené přihlašovací údaje LastPass

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

mocbezpecnyemail{ALT}{LCTRL}{LALT}

{CTRL}@seznam{OEM_PERIOD}

cz{LEFTMOUSE}

{SHIFT}{LSHIFT}Bezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD2}{TAB}{TAB}{SHIFT}{LSHIFT}Bezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD2}{LEFTMOUSE}

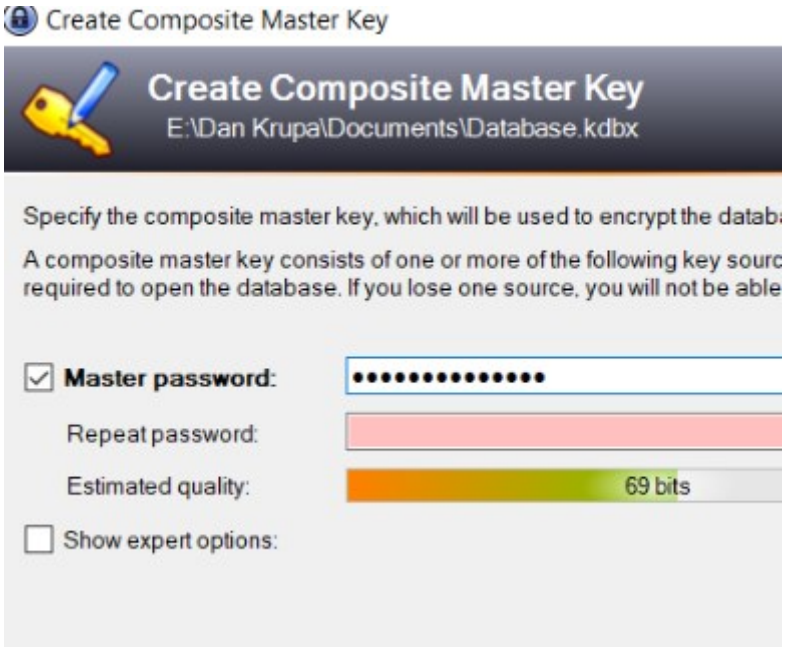
Výpis mimo jiné také ukazuje, že způsob, jakým keylogger zaznamenává speciální znaky, by si zasloužil trochu víc pozornosti. V každém případě ale LastPass dopadl víceméně stejně jako DashLane a přihlašovací údaje během tvorby účtu neochránil.

KeePass

Na rozdíl od předchozích password managerů nemá KeePass žádný účet. V podstatě jen formuje zkomprimovaný a zašifrovaný soubor. Není tedy třeba chránit uživatelské jméno, ale pouze Master heslo.

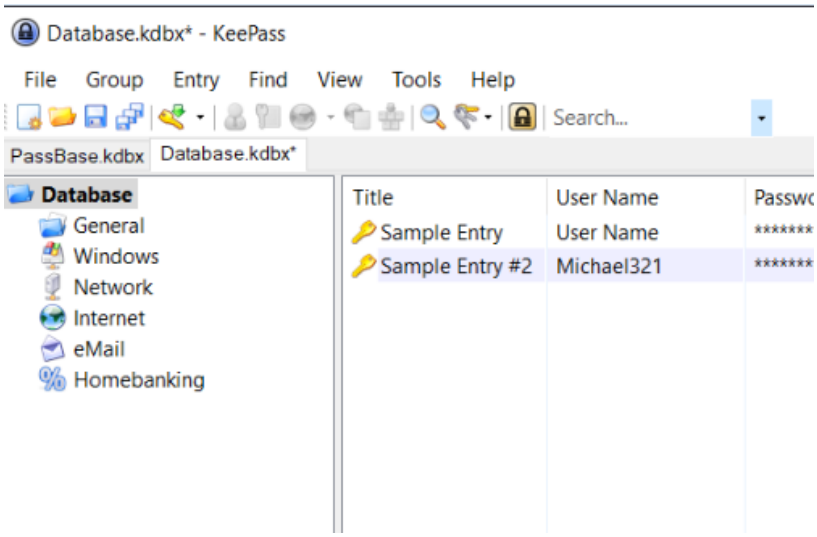
KeePass má samostatné okno, aktivoval tedy zrychlené snímání obrazovky, takže skener zachytil mnoho snímků z volby umístění souborů. První zajímavý snímek zachycuje již tvorbu Master hesla (obrázek 7) a druhý již napsané Master heslo (obrázek 8)

Obrázek 7 – Tvorba Master hesla KeePassu



Obrázek 8 – Vyplněné Master heslo KeePassu

Potom však zachytil až vytvořenou databázi s několika vzorovými záznamy, jak je vidět na obrázku 9.



Obrázek 9 – Databáze KeePassu

I když „uživatel“ i u KeePassu zobrazil Master heslo, skener obrazovky ho minul díky nízké frekvenci snímkování. Skener klávesnice byl úspěšnější.

Výpis 3 – Zachycené Master heslo KeePassu

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{SHIFT}{LSHIFT}Bezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD3}{LEFTMOUSE}

{SHIFT}{LSHIFT}Bezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD3}{LEFTMOUSE}

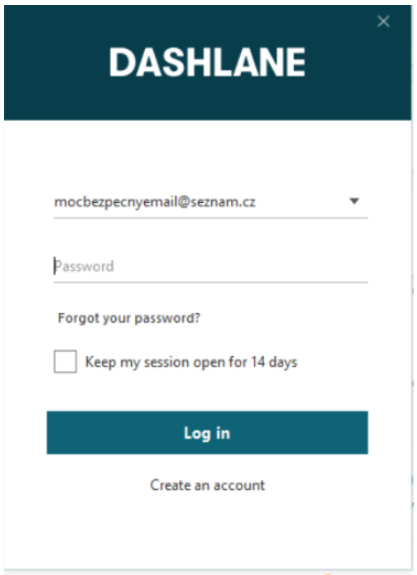
Skeneru obrazovky heslo sice uniklo, ale ve výpisu z keyloggeru je jasně vidět, že uživatel použil heslo „BezpecneHeslo3“. V konečném důsledku tedy ani KeePass hlavní heslo před krádeží neuchránil.

7.2. Experiment č. 2: Přihlášení pomocí Master hesla

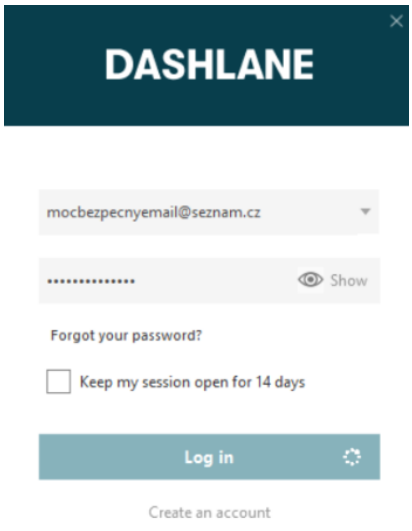
Ve druhém testu se pokusíme provést přihlášení pomocí hesla, které bylo vytvořeno před spuštěním skenerů. Cílem testu je zachytit Master heslo při přihlášení.

DashLane

V případě DashLanu skener nejprve zachytil přihlašovací okno aplikace, dokonce už s předvyplněným přihlašovacím jménem. Je to samostatné okno, takže skener začal snímat zvýšenou rychlostí. Zachytil několik kroků z vyplňování Master hesla, poslední z nich zobrazuje obrázek 10, ale samotné heslo nezjistil, tentokrát ho uživatel nezobrazil.



Obrázek 10 – Přihlašovací obrazovka DashLane



Obrázek 11 – Vyplněné údaje při přihlášení

Skener klávesnice vyprodukoval následující výpis:

Výpis 4 – Zachycené přihlašovací údaje DashLane

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

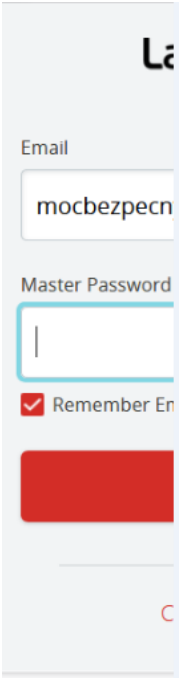
{LSHIFT}{SHIFT}Bezpecny{BACKSPACE}

e{SHIFT}{LSHIFT}Heslo{NUMPAD1}{LEFTMOUSE}

Skener klávesnice byl tedy úspěšný a heslo zachytil, navíc odhalil, že uživatel při jeho zadávání udělal překlep. I při zadávání Master hesla vytvořeného před spuštěním měl nakonec poslední slovo keylogger a heslo získal.

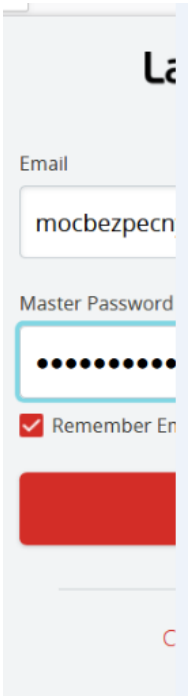
LastPass

Skener obrazovky při skenování přihlášení do LastPassu opět zradila chybná detekce rozměrů obrazovky, takže jak lze vidět na obrázku 12, zachytil pouze část už předvyplněného uživatelského jména.



Obrázek 12. – Přihlašovací dialog LastPass

Na několika dalších snímcích skener obrazovky zachytil postupné psaní Master hesla, ale ani teď jej uživatel nezobrazil. Na obrázku 13 je vidět jeden z těchto snímků.



Obrázek 13 – Vyplnění hesla při přihlášení

Skener obrazovky heslo neodchytil, ale opět jej odchytil skener klávesnice.

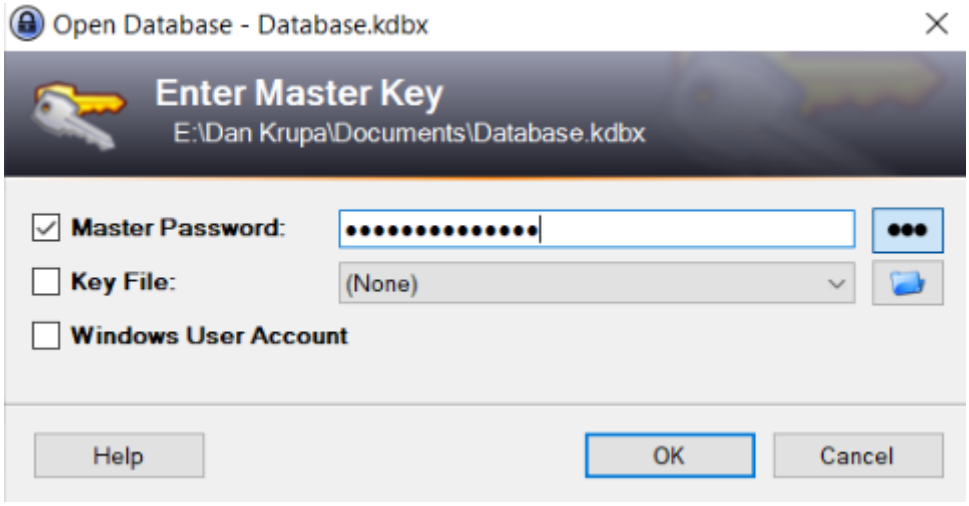
Výpis 5 – Zachycené heslo LastPassu

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{LSHIFT}{SHIFT}Bezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD2}{ENTER}

KeePass

V případě KeePassu skener obrazovky minul objevení přihlašovacího dialogu, pravděpodobně proto, že detekuje zrychlení podle přítomnosti slov jako „hesl“, „pass“, „login“ apod. Toto okno má však název „Open Database“. Následující obrázek však ukazuje, že skener obrazovky zachytil zadávání hesla



Obrázek 14 – Přihlášení do KeePassu15

Skener klávesnice se stále ukazuje být nejúspěšnější.

Výpis 6 – Zachycené Master heslo KeePassu

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{SHIFT}{LSHIFT}B{SHIFT}{LSHIFT}ezpecne{SHIFT}{LSHIFT}Heslo{NUMPAD3}{ENTER}

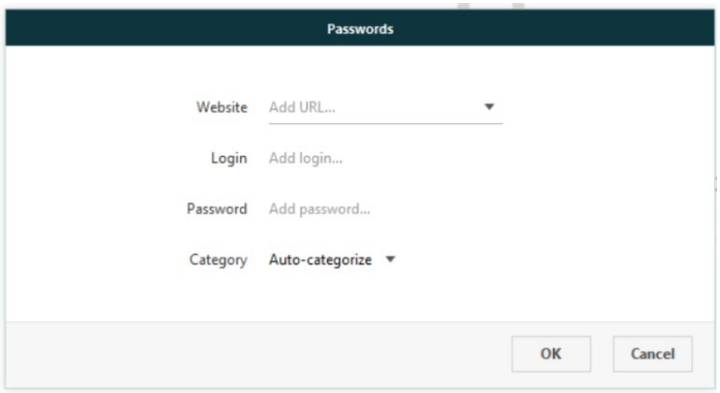
V dalším testu vyzkoušíme použít již otevřené password managery k jejich účelu. To znamená, uložíme do něj heslo.

7.3. Experiment č. 3: Tvorba přihlašovacího záznamu

K čemu by byla silná ochrana databáze a její šifrování, kdyby byla prázdná. V tomto testu se pokusíme vytvořit přihlašovací záznam pro přihlášení do emailu od společnosti Seznam.cz. Původně jsem zamýšlel použít Gmail, ale vypadá to, že password manager programy mají obvykle potíže s jeho přihlašováním v tom stylu, že vstup pro přihlašovací okno a vstup pro heslo jsou na různých stránkách. Cílem testu je zachytit přihlašovací údaje psané ručně uživatelem během tvorby zabezpečeného záznamu.

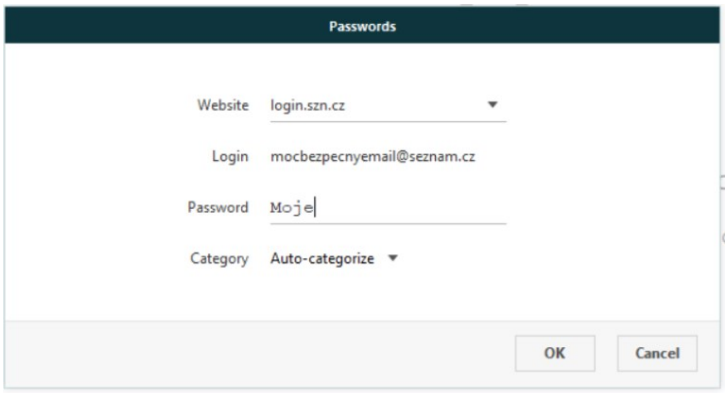
DashLane

DashLane dokáže sám detekovat přihlašovací formuláře a jemu dosud neznámé přihlašovací údaje ukládá. My však použijeme manuální tvorbu záznamu, protože pokud bychom zadali přihlašovací údaje do stránky a čekali, až DashLane zaznamená neznámé přihlášení, jedná se vlastně o analogii testu č. 2, bezmála stejná situace, jen jiný vzhled formuláře. Tvorba záznamu manuálně by však mohla být chráněna jinak. Na obrázku 16 vidíme úspěšně zachycený dialog pro tvorbu nového záznamu aplikace DashLane.



Obrázek 16 – Dialog pro tvorbu přihlašovacího záznamu DashLane

Zatím není nic vyplněno. Je vidět, že DashLane vyžaduje uživatelské jméno, heslo a webovou stránku, které patří tyto údaje. Další snímek odhaluje důležitou slabinu aplikace DashLane: zadávané heslo je nejen že zobrazeno v podobě čistého textu, ale DashLane dokonce ani nenabízí možnost jej zakrýt.



Obrázek 17 – Částečně zachycené heslo

Heslo by tedy snadno dokázal odhalit skener monitoru, kdyby tvořil snímky rychleji. Kompletní informace o hesle nakonec získal skener klávesnice:

Výpis 7 – Zachycené údaje přihlašovacího záznamu DashLane

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

login{OEM_PERIOD}

szn{OEM_PERIOD}

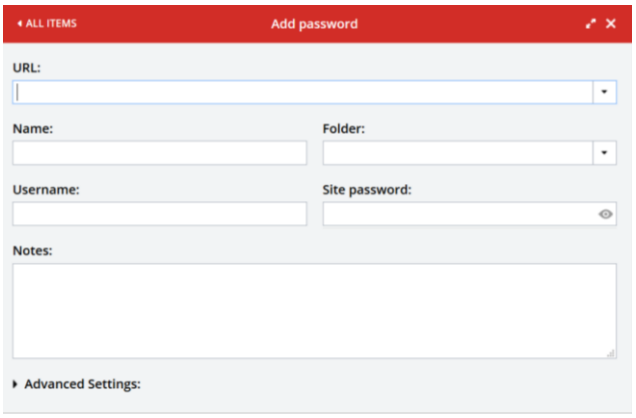
cz{TAB}mocbezpecnyemail{ALT}{LALT}

{CTRL}{LCTRL}@seznam{OEM_PERIOD}

cz{TAB}{SHIFT}{LSHIFT}Moje{SHIFT}{LSHIFT}Heslo{LEFTMOUSE}

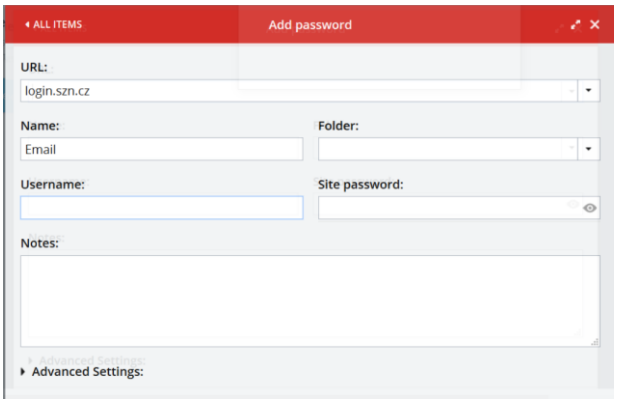
LastPass

LastPass se v otázce přihlašovacích záznamů chová velmi podobně, ne-li identicky jako DashLane, protože je jeho přímý konkurent. Skener monitoru zachytil dialog pro tvorbu záznamu:



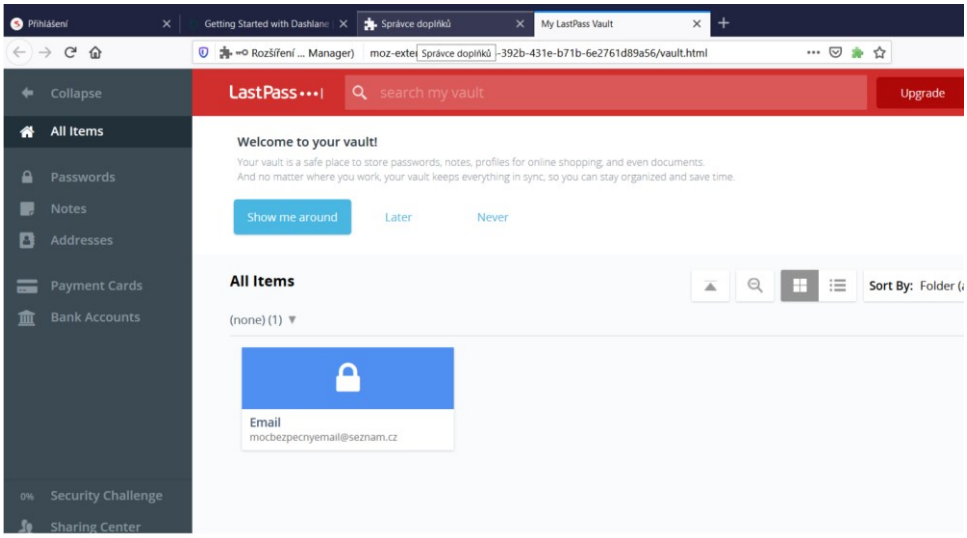
Obrázek 18 – Dialog pro tvorbu přihlašovacího záznamu LastPass

Také LastPass vyžaduje URL cílové stránky, uživatelské jméno a heslo, mimo jiné. Poslední relevantní snímek však zachytil jen napsanou URL:



Obrázek 19 – Dialog LastPass s vyplněnou URL

Na následujícím snímku (obrázek 20) je už vidět vytvořený záznam. Uživatel tedy pravděpodobně psal hodně rychle a současně skeneru vypršel čas zrychleného snímání. Okno má název „LastPass Vault“, což zrychlené snímání aktivovalo.



Obrázek 20 – Databáze LastPassu

Skener klávesnice neúspěch skeneru obrazovky vyrovnal a zachytil následující:

Výpis 8 – Zachycené údaje přihlašovacího záznamu LastPass

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

login{OEM_PERIOD}

szn{OEM_PERIOD}

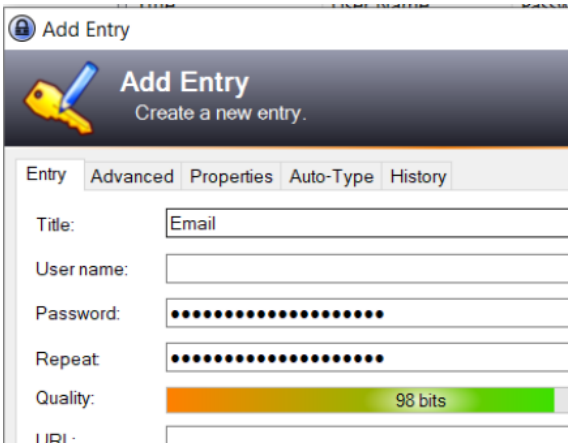
cz{TAB}{SHIFT}{LSHIFT}Email{TAB}{TAB}mocbezpecnyemail{ALT}{LALT}

{CTRL}{LCTRL}@seznam{OEM_PERIOD}

cz{TAB}{LSHIFT}{SHIFT}Moje{SHIFT}{LSHIFT}Heslo{LEFTMOUSE}

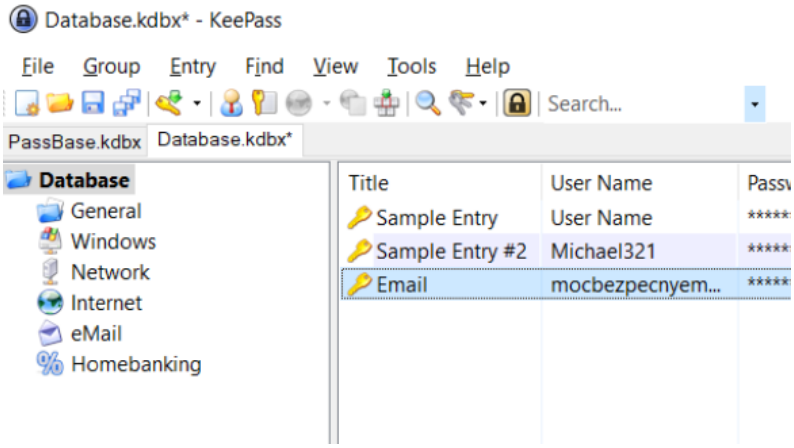
KeePass

KeePass na rozdíl od předchozích nedetekuje nic sám, jeho databáze je jen soubor, který sám od sebe nic nedělá. Skener monitoru jako první zajímavý snímek pořídil toto:



Obrázek 21 – Dialog pro tvorbu přihlašovacího záznamu KeePassu

Uživatel nejprve vyplnil titulek, heslo bylo již předgenerováno, KeePass je ale sám nezobrazil, ačkoliv mohl. Následující snímek už ale ukazuje až hotový záznam:



Obrázek 22 – Nový záznam v KeePassu

Skener monitoru opět minul většinu tvorby záznamu. Ochrana údajů doposud končila na skeneru klávesnice, nejinak je tomu i nyní:

Výpis 9 – Zachycené údaje přihlašovacího záznamu KeePassu

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{SHIFT}{LSHIFT}Email{TAB}{TAB}mocbezpecnyemail{ALT}{LCTRL}{LALT}

{CTRL}@seznam{OEM_PERIOD}

cz{TAB}{SHIFT}{LSHIFT}Moje{LSHIFT}{SHIFT}Heslo{TAB}{TAB}{LSHIFT}{SHIFT}Moje{SHIFT}{LSHIFT}Heslo{LEFTMOUSE}

V první třech testech dosud všechny přihlašovací údaje byly odhaleny. Test č. 4 spočívá v použití tohoto záznamu k přihlášení.

7.4. Experiment č. 4: Přihlášení pomocí password manageru

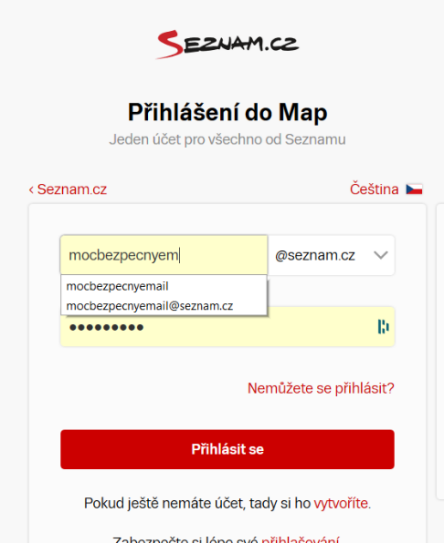
Ve čtvrtém testu použijeme přihlašovací údaje vytvořené v minulém kroku pro přihlášení do samotné stránky login.szn.cz, tedy na přihlašovací stránky emailu společnosti Seznam.cz. Tento test bude mít dvě části. V první části se pokusíme překopírovat heslo z databáze password manageru do přihlašovacího okna. V části druhé použijeme vestavěnou doplňovací schopnost daného password manageru. Cílem tohoto testu bude zachytit přihlašovací údaje při jejich použití pro přihlášení.

DashLane

DashLane má stejně jako všechny testované password managery schopnost zkopírovat heslo, aniž by ho musel zobrazit na obrazovku. To zachycuje tento snímek:

Obrázek 23 – Kopírování hesla z DashLane

Přesněji řečeno, toto je obrazovka, odkud lze heslo kopírovat. Kopírovací tlačítko je mimo záběr vedle tlačítka zobrazení. Plusem pro DashLane je, že bez ohledu na skutečnou délku hesla vždy zobrazuje šest teček. Při vkládání do přihlašovacího formuláře se heslo nezobrazuje, viz obrázek 24:



Obrázek 24 – Vkládání hesla do Seznam Email

Skener obrazovky se tedy nic nového nedozvěděl. Podíváme se, co zachytil skener klávesnice:

Výpis 10 – Kopírování hesla z DashLane nebylo zachyceno

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{CTRL}{LCTRL}{LEFTMOUSE}

mocbezpecnyemail{LEFTMOUSE}

Tentokrát skener klávesnice přihlašovací údaje nezískal. Získal jen uživatelské jméno. Místo hesla zaznamenal jen stisknutí Ctrl+V, které navíc vyhodnotil jako znak o neznámé hodnotě. Ale poprvé zareagoval skener schránky:

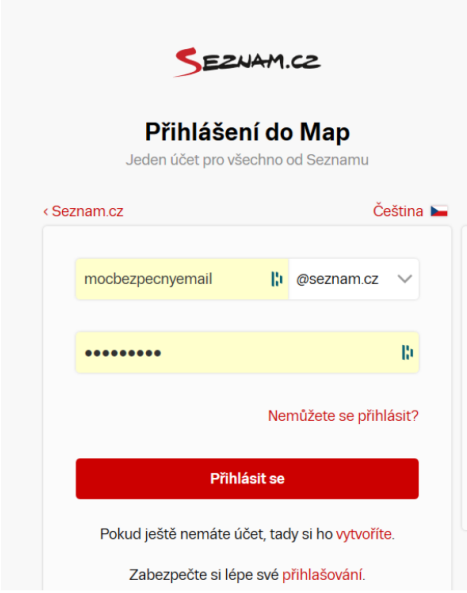
Výpis 11 – Zachycení hesla DashLane ve schránce

cliptext-DESKTOP-MS4ABNQ-09_05_2020.txt:

09_05_2020_10_36_42 MojeHeslo

Skener schránky zachytává schránku každých 10 vteřin, pokud v ní něco je. V tomto případě úspěšně zachytil obsah schránky a získal heslo. Řetězec číslic a podtržítek je přeformátované datum a čas odchycení.

Druhá část testu spočívá v použití vkládacího mechanismu samotného DashLanu.
Po odhlášení a opětovném zobrazení přihlašovací stránky skener monitoru zachytil toto:



Obrázek 25 – Automatické vyplnění přihlašovacích údajů programem DashLane

DashLane funguje na úrovni prohlížeče a při načítání stránky už údaje předvyplnil. Skener obrazovky se tedy heslo nedověděl, jen přihlašovací jméno. Skener klávesnice zachytil následující výpis:

Výpis 12 – Automatické vyplnění DashLane nezachycen

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{LEFTMOUSE}

{F5}

{LEFTMOUSE}

Takže jsme se dověděli, že uživatel je dost znalý na to, aby používal klávesu F5 na obnovení stránky. To je ale vše, heslo jsme se nedověděli. Zde je výpis skeneru schránky:

Výpis 13 – Zachycené heslo ve schránce

cliptext-DESKTOP-MS4ABNQ-09_05_2020.txt:

09_05_2020_10_38_12 MojeHeslo

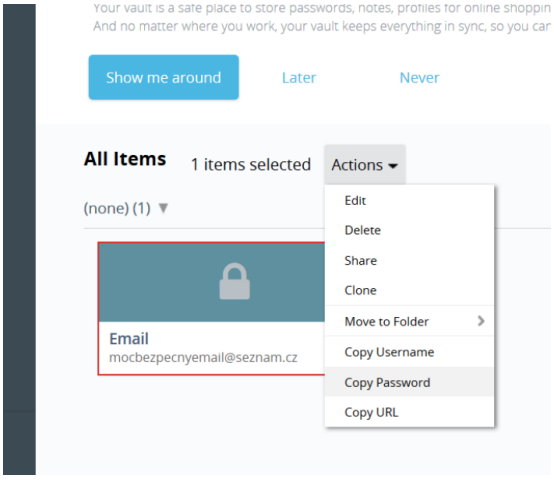
09_05_2020_10_38_22 MojeHeslo

09_05_2020_10_38_32 MojeHeslo

Heslo tedy bylo zachyceno. Z výpisu však není jasné, zda se jedná o nově zachycené heslo anebo to, které bylo zachyceno při předchozím kopírování, protože skener schránky není zřejmě dostatečně propracovaný, aby si s takovou situací poradil. Heslo tedy sice máme, ale skener schránky nedokáže říct jeho zdroj.

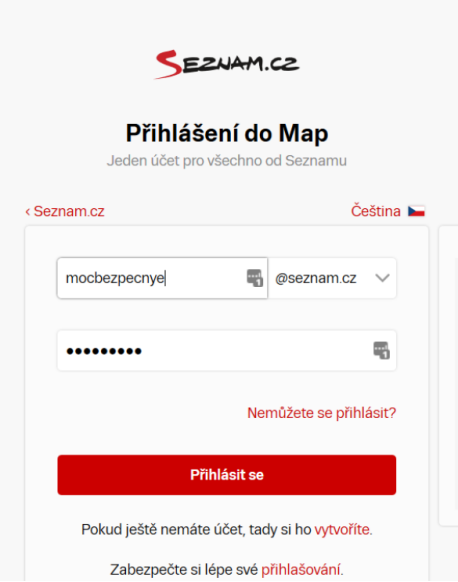
LastPass

LastPass velmi podobně jako DashLane dokáže kopírovat heslo, aniž by ho zobrazil, jak ukazuje tento snímek:



Obrázek 26 – Kopírování hesla z LastPass

Ten následující je už ze série zachycující vyplňování uživatelského jména po tom, co bylo zkopírováno heslo:



Obrázek 27 – Vložení hesla do Seznam Email

Skener obrazovky tedy heslo nezískal, jen odhalil malý rozdíl mezi DashLanem a LastPassem v uživatelské přívětivosti, a sice ten, že LastPass u své ikonky v přihlašovacím poli ukazuje číslo. Toto číslo znamená počet dvojic uživatelské jméno-heslo, které LastPass pro toto okno zná. Skener klávesnice vyrobil prakticky stejný výpis jako v případě DashLanu:

Výpis 14 – Kopírování hesla z LastPass nezachyceno

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{CTRL}{LCTRL}{LEFTMOUSE}

mocbezpecnyemail{LEFTMOUSE}

Opět neznámý znak jako Ctrl+V a napsané uživatelské jméno. Heslo není. Skener schránky neustále vypisuje, že ve schránce je stále totéž heslo:

Výpis 15 – Zachycené heslo ve schránce

cliptext-DESKTOP-MS4ABNQ-09_05_2020.txt:

09_05_2020_10_39_32 MojeHeslo

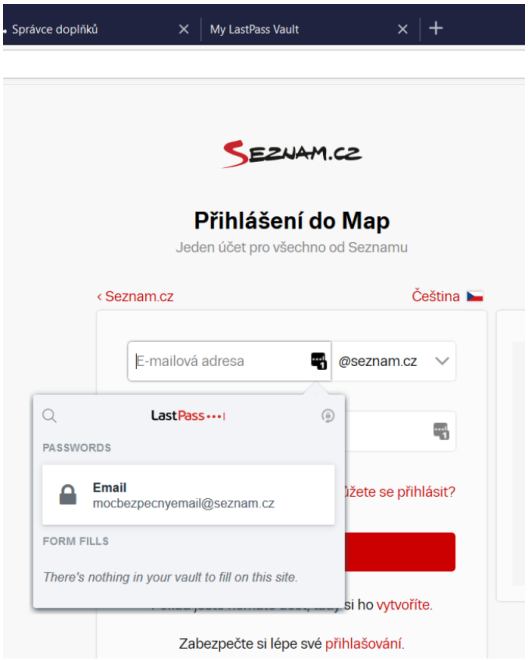
09_05_2020_10_39_42 MojeHeslo

09_05_2020_10_39_52 MojeHeslo

09_05_2020_10_40_02 MojeHeslo

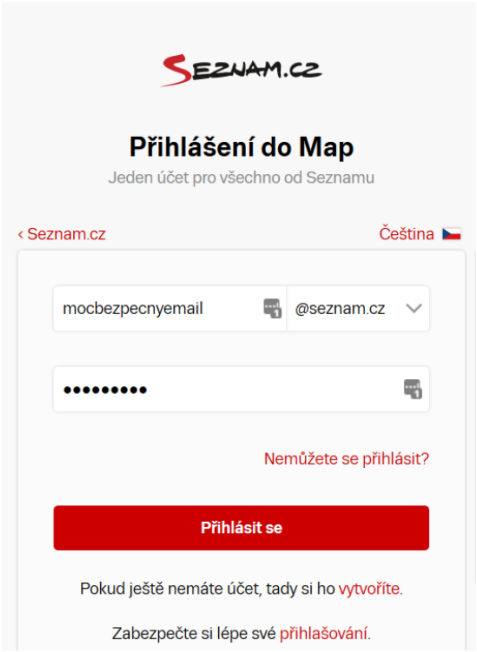
Teď už však ani nevíme, zda heslo vůbec pochází z LastPassu, nebo si schránka stále ještě pamatuje heslo z DashLanu, protože se jedná o přihlašovací údaje ke stejnému emailu. Schránka je sama o sobě nekonečná, pamatuje si svou hodnotu tak dlouho, dokud není změněna. Vzhledem k tomu, jak moc jsou si LastPass a DashLane podobné, je pravděpodobné, že při kopírování z LastPassu bylo heslo také zachyceno, ale žádný ze tří skenerů to nedokázal prokázat.

Nyní zkusíme ještě vložit heslo vestavěným vkládacím procesem LastPassu:



Obrázek 28 – Automatické vkládání hesla LastPass

Snímek ukazuje, že LastPass je o něco opatrnější a nevyplňuje údaje sám hned napoprvé ve výchozím stavu. To se však dá změnit v jeho nastavení. Následující snímek u ukazuje vyplněné údaje.



Obrázek 29 – Vložené přihlašovací údaje

Skener klávesnice zachytil zase jen znovunačtení stránky:

Výpis 16 – Automatické vyplnění LastPass nezachyceno

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{LEFTMOUSE}

{F5}

{LEFTMOUSE}

Skener schránky se evidentně stal při provádění těchto testů nepoužitelným, aktuálně nelze rozlišit, odkud heslo pochází, protože je stále stejné:

Výpis 17 – Zachycené heslo ve schránce

cliptext-DESKTOP-MS4ABNQ-09_05_2020.txt:

09_05_2020_10_41_02 MojeHeslo

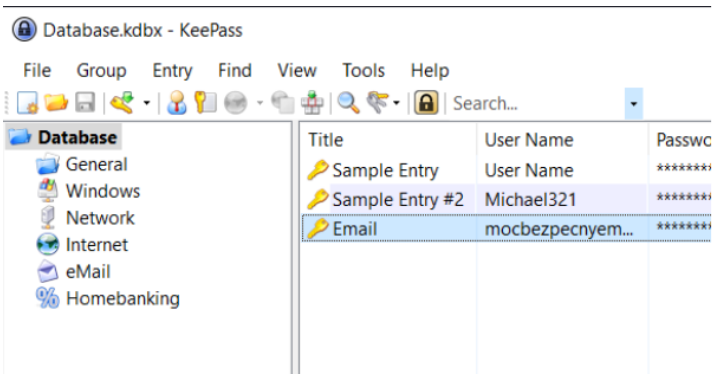
09_05_2020_10_41_12 MojeHeslo

09_05_2020_10_41_22 MojeHeslo

09_05_2020_10_41_32 MojeHeslo

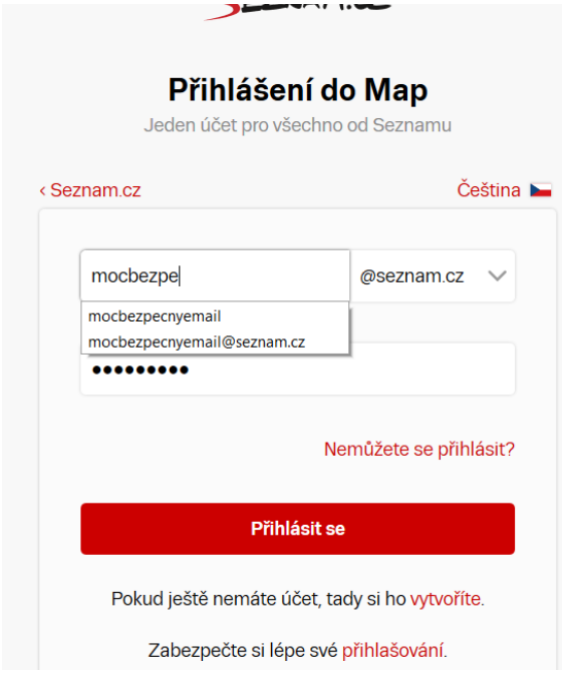
KeePass

KeePass automaticky čistí schránku, pokud je použita na zkopírování hesla. U ostatních password managerů předpokládám totéž chování, ale nepodařilo se mi ověřit, že by to tak skutečně bylo. Viz skener schránky, který i přes minutu heslo evidentně stále ve schránce vidí. Buď tedy LastPass a DashLane schránku nečistí, nebo mají moc dlouhý časový interval.



Obrázek 30 – Kopírování hesla z KeePass

Tento snímek zachycuje část okna KeePassu v průběhu kopírování hesla. Bohužel špatné vnímání rozměrů obrazovky nedovoluje zahlédnout, že v pravém dolním rohu okna začala běžet časomíra, po které KeePass schránku pročistí. Následující snímek už je opět ze série zachycující psaní přihlašovacího jména



Obrázek 31 – Vkládání hesla z KeePass do Seznam Email

Skener klávesnice zachytil tuto akci velmi podobně jako u předchozích programů:

Výpis 18 – Zachycení automatického vyplnění KeePass

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{TAB}{CTRL}{LCTRL}{LEFTMOUSE}

{CTRL}{LCTRL}{LEFTMOUSE}

mocbezpecnyemail{ENTER}

Zachytil tedy nejspíše stisk kláves Ctrl+C a Ctrl+V a ruční napsání uživatelského jména. Heslo opět neznáme. Skener schránky mezitím vytvořil následující výpis:

Výpis 19 – Zachycené heslo ve schránce a promazání schránky

cliptext-DESKTOP-MS4ABNQ-09_05_2020.txt:

09_05_2020_10_42_12 MojeHeslo

09_05_2020_10_42_23 MojeHeslo

09_05_2020_10_42_33 MojeHeslo

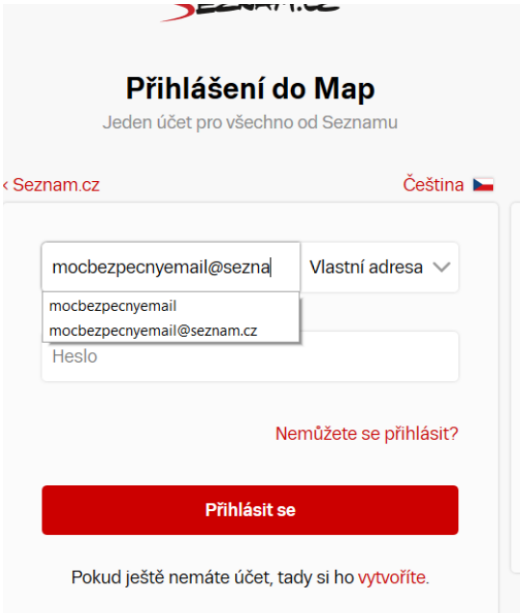
09_05_2020_10_42_43

Zde vidíme, že ačkoliv skener heslo zachytil, KeePass schránku pročistil tak, že do ní vložil prázdný znak.

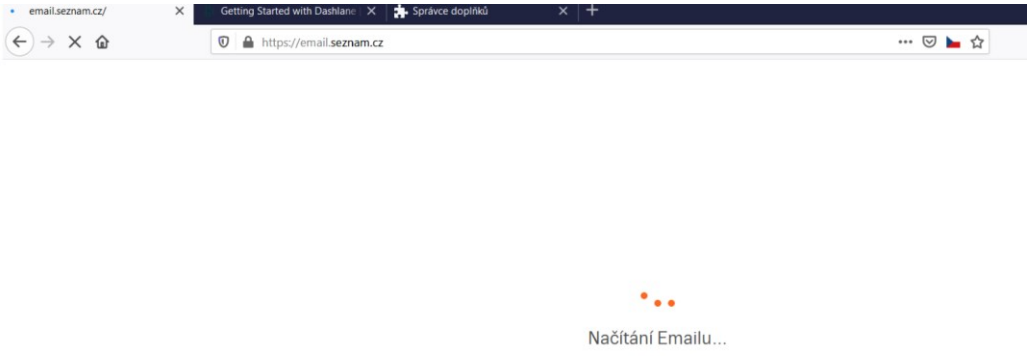
Posledním testem přihlášení je použití vkládacího algoritmu KeePassu. Na rozdíl od LastPassu a DashLanu, které jednorázově vkládají údaje do správných polí, pravděpodobně pomocí svých vnitřních proměnných, KeePass používá simulované stisky kláves. KeePass také nepoužívá žádnou automatiku,

která by aktivovala doplňování, to musí udělat uživatel sám, a sice tak, že klikne na první přihlašovací pole, pak otevře KeePass, vybere správný záznam a stiskne Ctrl+V. Tu použijeme v našem testu. Druhou možností je detekce pomocí názvu okna a URL. Na tohle lze v KeePassu nadefinovat klávesovou zkratku, která aktivuje vyplnění, aniž by se musel KeePass dostat do popředí.

Následující snímek už ukazuje průběh zadávání. KeePass píše velmi rychle, a proto i zrychlený skener zachytil jen jeden snímek. Na druhém je vidět už průběh přihlašování:



Obrázek 32 – Automatické vkládání KeePassu



Obrázek 33 – Načítání Seznam Email po přihlášení

Heslo tedy skener monitoru nezaznamenal. Skener klávesnice zachytil toto:

Výpis 20 – Zachycení automatického vyplňování KeePassem

keylog-DESKTOP-MS4ABNQ-09052020_103117-00000405.txt:

{CTRL}{LCTRL}zpem{LCTRL}{RALT}

{CTRL}{ALT}zm{LSHIFT}{SHIFT}{LSHIFT}{SHIFT}{LEFTMOUSE}

To znamená, že KeePass používá klávesnici, aby odeslal pouze některé znaky přihlašovacích údajů. Pro ostatní znaky mohl použít schránku anebo vlastní paměť. Skener schránky zapsal stále jen prázdná data.

Výpis 21 – Zachycení prázdné schránky

cliptext-DESKTOP-MS4ABNQ-09_05_2020.txt:

09_05_2020_10_46_03

09_05_2020_10_46_13

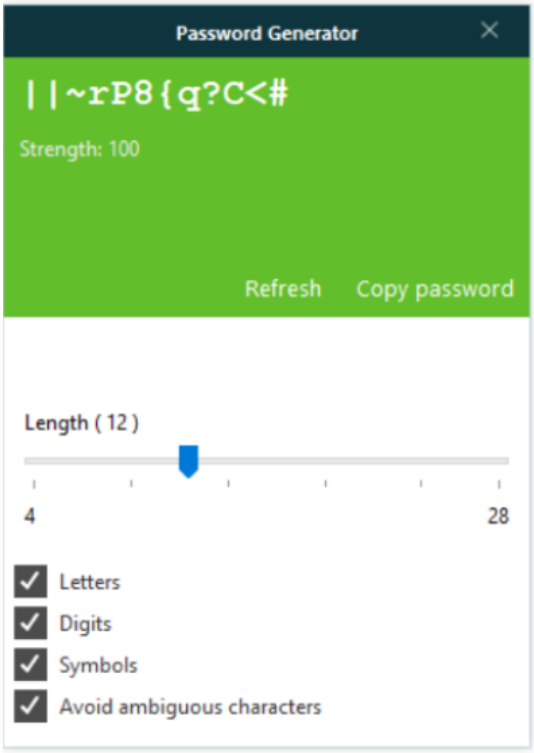
To může mít dva důvody. Prvním z nich je, že KeePass schránku nepoužil a použil vlastní paměť. Druhým z nich je to, že KeePass schránku sice použil, ale zaprvé ji pak vyčistil, a také interval mezi dvěma skeny schránky byl příliš dlouhý. Interval je 10 sekund, ale vyplnění formuláře zabralo KeePassu jen asi dvě sekundy. KeePass tedy zjištění hesla zabránil.

7.5. Experiment č. 5: Generování bezpečného hesla pomocí utilit password manageru

Všechny testované password managery jsou schopné samy vygenerovat bezpečné heslo z uživatelsky definované znakové sady. V tomto testu vygenerujeme s jejich pomocí silné heslo a vyzkoušíme, zda špehovací skript toto heslo dokáže zachytit z generátoru.

DashLane

DashLane prokázal, že silná hesla generovat umí. Avšak vygenerované heslo implicitně zobrazil na obrazovce:

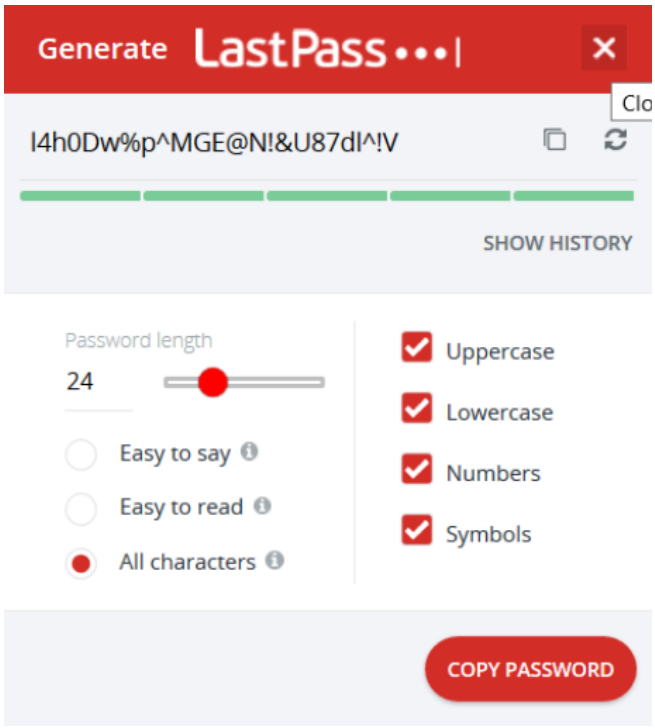


Obrázek 34 – DashLanem vygenerované silné heslo

Skener obrazovky snadno heslo zaznamenal. Tlačítko Copy password by také lákalo k otestování. Ale heslo máme a při testu přihlášení přes kopírované heslo skener schránky heslo zachytil, takže je pravděpodobné, že výsledek zde by byl stejný jako u experimentu č.4.

LastPass

LastPass napoprvé zvolil trochu delší heslo, jinak se ale od DashLanu v otázce tohoto testu nijak neodlišil. I on vygenerované heslo okamžitě zobrazil:

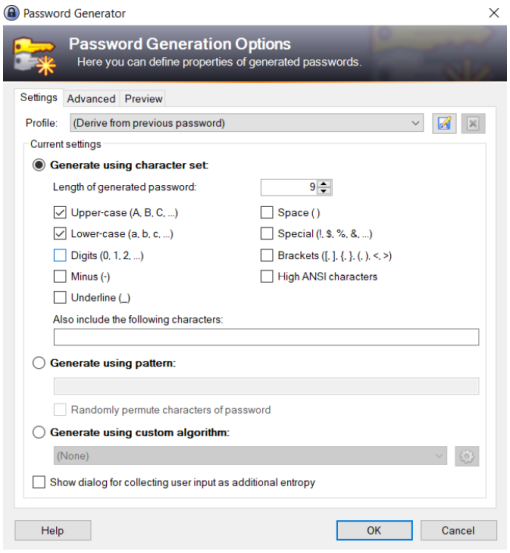


Obrázek 35 – LastPassem vygenerované silné heslo

I jeho heslo bylo skenerem obrazovky okamžitě zachyceno.

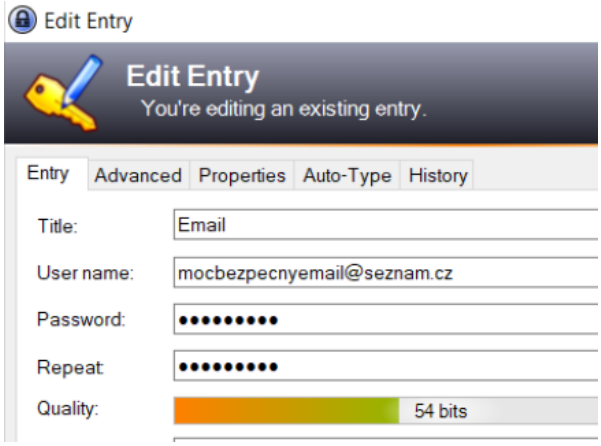
KeePass

KeePass jako jediný heslo okamžitě neukázal. Vložil vygenerované heslo okamžitě do přihlašovacího záznamu, jehož se generování týkalo. Na rozdíl od předchozích negeneruje heslo jen tak pro uživatele, ale určí ho konkrétnímu záznamu. Skener obrazovky zachytil následující dialog:



Obrázek 36 – Dialog pro generování hesla KeePass

Ten ukazuje, že KeePass má mnohem jemnější kontrolu nad tím, jak heslo vygenerovat, a nabízí uživateli mnohem více možnosti celý proces ovlivnit. U LastPassu a DashLanu si uživatel mohl zvolit jen několik základních sad znaků a délku. Nemohl zasáhnout do algoritmu. U KeePassu je mnoho podstatně rozsáhlejších možností. Vygenerované heslo bylo vloženo přímo do přihlašovacího záznamu a skener ho tedy nezachytil:



Obrázek 37 – Vygenerované heslo v záznamu KeePass

Uživatel sice posléze nové heslo zobrazil, ale skener ho minul a další snímek jím pořízený je identický s obrázkem 37. Skener monitoru tedy heslo zachytit nestihl.

8. Souhrn výsledků testů

Password managery prokázaly v testech, že mají mnoho společného. Žádný z nich nedokázal uchránit své Master heslo ani heslo manuálně vytvořeného přihlašovacího záznamu před skenerem klávesnice, který se ukázal být nejúspěšnějším ze tří modulů. Skener monitoru však Heslo zachyceno jen díky tomu, že netrefoval úplně všechny správné okamžiky a tomu, že nezachytával úplně správně celý rozsah obrazovky. „Uživatel“ mu tak trochu „pomohl“ tím, že zadávaná hesla zobrazoval, přesto skener obrazovky nezachytil všechna. Je zřejmé, že jeho schopnost zachytit heslo zobrazené na obrazovce je spíše věcí náhody.

Skener schránky byl dlouhou dobu spící hrozbou, ale nakonec i on zaznamenal jistý úspěch. Při testování odhalování stále téhož hesla však nebylo možné rozlišit, co vložilo do schránky obsah, který právě vidí. Při čtvrtém testu se ale ukázal být jediným modulem, který dokázal heslo získat. Tyto skutečnosti dávají trochu možnost předpovědět, jak by dopadl v testu program PwdHash popsaný dříve v této práci, pokud by byl do testů zapojen. PwdHash nepoužívá žádné Master heslo a netvoří žádnou databázi, což považuji za výhodu, protože by u něj většinu testů provedených v této práci nebylo možné vůbec provést. Soudě podle výsledků testů u ostatních password managerů by přihlašování pomocí jeho generovací stránky pravděpodobně bylo zachyceno nejméně dvěma skenery, protože je heslo zobrazeno v čistém textu (to by mohl zachytit skener obrazovky) a je nutné ho přepsat (to by mohl zachytit skener klávesnice) anebo zkopírovat (to by zase mohl zachytit skener schránky). Navíc heslo, které uživatel zná, které zadává do PwdHashe, aby z něj vygeneroval bezpečné heslo, musí být také napsáno/vloženo. Obrázek 38 je již vytvořený manuálně a ukazuje generátor PwdHashe přístupný na jeho stránkách. (38)

Obrázek 38 – Generátor hesel PwdHash

Všechny testované password managery naopak prokázaly, že si velmi dobře umí poradit se skenery, co se týče vkládání hesel pomocí svých vnitřních funkcí. DashLane a LastPass zafungovaly na úrovni prohlížeče a skenery jejich aktivitu vůbec nezaznamenaly, a KeePass, který působí, že pravděpodobně používá nejméně dvě různé cesty, kterými dostává znaky do cílových polí, dokázal většinu znaků uchránit. Skener klávesnice některé z nich přečetl, ale nebylo možné z nich vyčíst, které vůbec patří do hesla a které do přihlašovacího jména. Navíc skener zaznamenal stisky speciálních kláves, které uživatel vůbec nestiskl. KeePass má tedy evidentně podstatně komplexnější způsob vkládání údajů. Tabulka č. 2 ukazuje porovnání, jak si vedly jednotlivé password managery v jednotlivých testech, včetně odhadů, jak by dopadl PwdHash, vytvořených na základě výsledků ostatních password managerů.

Tabulka 2 – Srovnání výsledků testu a odhad výsledků pro PwdHash

	KeePass	LastPass	DashLane	PwdHash
Experiment č. 1 – tvorba Master hesla	Heslo zachyceno	Heslo zachyceno	Heslo zachyceno	N/A[1]
Experiment č. 2 – přihlášení do password manageru	Heslo zachyceno	Heslo zachyceno	Heslo zachyceno	N/A[1]
Experiment č. 3 – tvorba přihlašovacího záznamu	Heslo zachyceno	Heslo zachyceno	Heslo zachyceno	N/A[2]
Experiment č. 4 část A – přihlášení pomocí password manageru za užití kopírování	Heslo pravděpodobně zachyceno[3]	Heslo pravděpodobně zachyceno[3]	Heslo zachyceno	Pravděpodobně by bylo heslo zachyceno
Experiment č. 4 část B – přihlášení pomocí password manageru za užití jejich vkládacích funkcí	Heslo prakticky nezachyceno[4]	Heslo nezachyceno	Heslo nezachyceno	Efektivní heslo by uniklo, ale bylo by zachyceno základní heslo.
Experiment č. 5 – vygenerování bezpečného hesla generátorem	Heslo nezachyceno[5]	Heslo zachyceno	Heslo zachyceno	Heslo by bylo zachyceno[6]

[1] PwdHash nepoužívá Master heslo.
[2] PwdHash neukládá záznamy.
[3] Skener schránky nebyl dost dobře navržený a nedokázal rozlišit, čí heslo vidí.
[4] Některé znaky byly zachyceny keyloggerem, ačkoli by hledání hesla pravděpodobně příliš nezjednodušily.
[5] Prošel jen proto, že skener monitoru neudělal snímek ve správný okamžik. Prošel by zcela, pokud by uživatel byl dost zkušený a nezobrazil heslo vůbec.
[6] Webová stránka zobrazuje efektivní, tedy vygenerované heslo v čistém textu a je třeba ho přepsat nebo zkopírovat a tím být zachycen skenerem klávesnice nebo schránky. Plugin zase vyžaduje napsání základního hesla, což činí efektivní heslo bezpředmětným, protože ho lze při znalosti základního hesla a cílové URL snadno vygenerovat.

Tabulka č. 3 shrnuje, který skener byl úspěšný v daném testu.

Tabulka 3 – Úspěšnost skenerů

	Skener obrazovky	**Skener klávesnice**	**Skener schránky**
Experiment č. 1 – tvorba Master hesla	Částečně uspěl díky nezkušenosti uživatele[7]	Uspěl	Neuspěl
Experiment č. 2 – přihlášení do password manageru	Částečně uspěl díky nezkušenosti uživatele[7]	Uspěl	Neuspěl
Experiment č. 3 – tvorba přihlašovacího záznamu	Částečně uspěl díky nezkušenosti uživatele[7]	Uspěl	Neuspěl
Experiment č. 4 část A – přihlášení pomocí password manageru za užití kopírování	Neuspěl	Neuspěl	Pravděpodobně uspěl[8]
Experiment č. 4 část B – přihlášení pomocí password manageru za užití jejich vkládacích funkcí	Neuspěl	Prakticky neuspěl[9]	Neuspěl
Experiment č. 5 – vygenerování bezpečného hesla generátorem	Částečně uspěl[10]	Neuspěl	Neuspěl

Jednotlivé skenery tedy prokázaly, že společně jsou poměrně účinné, v každém testu kromě testu č. 4 alespoň jeden uspěl. V zhruba polovině případů se však projevilo také nezkušené chování uživatele. Některé z testů byly nebo dostaly šanci být úspěšné jen díky tomu.

[7] Uživatel zobrazil heslo, ačkoli nemusel. Pokud by ho nezobrazil, skener monitoru by neuspěl. Úspěch tohoto skeneru byl také závislý na náhodě a načasování pořizování snímků.
[8] Není rozpoznatelné, z jakého zdroje má zachycená schránka heslo
[9] Zachytil několik málo znaků, ale nebylo možné poznat, zda patří přihlašovacímu jménu nebo heslu ani kde jsou v něm umístěny
[10] Skener monitoru minul jen díky nedokonalému časování

9. Závěr

V rámci této bakalářské práce jsem provedl pokus ověřit efektivitu spywaru proti programům z kategorie password manager, jejichž cílem je ochrana hesel a procesu zadávání přihlašovacích údajů. Provedl jsem průzkum, jak vypadá aktuální stav spywaru. Soudě podle výš zmíněných nalezených příkladů se nejen spyware, ale malware jako celek stávají modulárními programy, které se skrývají hluboko v systému, jsou ovládány z dálky a objevují se ve velkém množství. Na základě toho jsem se rozhodl vytvořit jednoduchý špehovací kód, který se skládá z několika modulů. V rámci práce bylo provedeno 5 experimentů, jejichž cílem bylo zjistit efektivitu tohoto spywaru proti programům z kategorie password managerů. Pro účely testování byl spyware spuštěn ve viditelných oknech. Experimenty prokázaly, že testované password managery mohou být náchylné na únik hesel.

Pozornost je třeba věnovat také faktu, že některé části spywaru nedělaly svou práci úplně dobře. To se týká špatné detekci hranic obrazovky u skeneru monitoru, který detekoval jen část obrazovky. Dále také příliš pravidelného zachytávání v případě skeneru schránky.

Keylogger fungoval velmi dobře, jen je třeba zapracovat na tom, aby byl jeho výstup více čitelný. Všechny tři skenery, obzvláště skener schránky, by bylo vhodné implementovat odlišným způsobem, a sice pomocí takzvaných hooků, tedy přípojných bodů, kde lze instalovat různé monitorovací procesy ve Windows. Jedná se o přístup řízený událostmi, který nepotřebuje žádné časovače, jaké používají skenery z této práce. Takový spyware by mnohem méně zatěžoval paměť. Během testu spywaru během této práce bylo na testovaném zařízení znát výrazné zpomalení. Jinak ale nebyla na první pohled znát žádná další indicie, která by aktivitu spywaru odhalovala. Ani Avast antivirus přítomný v systému na jeho aktivitu nereagoval.

Ukládací cesty by měly vést na nějaké místo na síti, anebo by měl existovat další modul, který bude zajišťovat odstraňování souborů z hostitelského systému a jejich přenos k útočníkovi.

Password managery selhaly hlavně při přijímání vstupu od uživatele. Spyware vždy získal přinejmenším hlavní heslo. Je třeba si však uvědomit, že jsme zdaleka netestovali plnou sílu těchto password managerů. Například jsme netestovali schopnost KeePassu používat k autentizaci klíčový soubor a Windows účet. LastPass a DashLane totiž ve své základní verzi nemají nic podobného, co by proti tomu mohli postavit. Jejich druhou úrovní zabezpečení je jednorázové heslo, nebo v případě LastPassu mnoho dalších možností, jejichž dostupnost se liší napříč jeho různými Premium a Business verzemi. LastPass v průběhu psaní této práce změnil své plány a uvolnil synchronizaci vlastním cloudem i pro Free verze. (39)

Testy také prokázaly, že i když password manager dokáže hodně pomoci, dokáže to pouze tehdy, pokud je uživatel zkušený a rozumí tomu, jak ho používat, jako například že by neměl zbytečně zobrazovat hesla na obrazovce. Úplně vyhnout se tomu pravděpodobně nedá, ale je dobré tuto akci omezit na minimum. Rozhodně by také uživatel neměl používat schránku na přenos hesla. Vkládací algoritmy jednotlivých password managerů se ukázaly být jejich nejsilnější schopností a hlavním důvodem, proč je vhodné přemýšlet o jejich užívání. Žádný skener v této práci nedokázal zachytit nic relevantního, dokonce i přesto, že jsme u KeePassu nepoužili jeho zesílenou formu vkládání, a sice dvoucestnou obfuskaci, která vkládá jednotlivé znaky na přeskáčku, na rozdíl od základního vkládání, které probíhá lineárně. Je ale možné, že spyware fungující na úrovni webové stránky, například v JavaScriptu, by to mohl zvládnout.

Další věcí je verzatilita: LastPass a DashLane dokážou detekovat i různé platební formuláře a různé další podobné uživatelské vstupy. Proti tomu KeePass je ochoten vyplnit naprosto cokoliv, co akceptuje vstupy z klávesnice. Na rozdíl od ostatních je u něj použití URL dobrovolné a vždy se dá obejít, jak jsme ostatně v testu prokázali. Pro zajímavost, způsob zaznamenávání speciálních kláves skenerem klávesnice byl částečně upraven tak, aby KeePass mohl replikovat aktivitu uživatele na klávesnici. Používání kláves jako Enter, Esc, Ctrl, Shift, Alt mu nedělá problém, i když právě kombinaci Ctrl, Shift, Alt nemůže vygenerovat ze záznamu našeho keylogu. Používá pro ně totiž znaky +, ^ a %, ne názvy kláves. Podle dokumentace by měl být schopen nejen vygenerovat všechny kódy virtuálních kláves, ale dokonce také spouštět jiné aplikace, což by také mohlo být bezpečnostním rizikem. (40) Uživatel KeePassu tedy musí být mnohem opatrnější než v případě DashLanu nebo LastPassu.

Obecně se ale Password managery ukázaly být příliš slabé, aby zabránily již spuštěnému spywaru dostat se k heslům. Pokaždé se našla aspoň jedna cesta, jak se dostat buď do databáze, nebo k samotnému záznamu. Je možné, že by si s tím poradil prototyp password manageru Amnesia čínských autorů Luren Wang a Yue Li (41), který se skládá z více separovaných částí a používá více druhů tajných informací, aby poskytoval hesla. Je to generativní password manager podobně jako PwdHash, nic neskladuje, ale používá Master heslo k autentizaci uživatele. Jedná se o projekt ve fázi výzkumu, možná se dostal do fáze, kdy je někde používán neveřejně. Nepodařilo se mi najít žádné informace, které by potvrzovaly, že by byl někde nasazen. Nejdůležitější proto stále zůstává udržovat svůj operační systém a jeho antivirové, antispywarové a ostatní anti- programy aktuální a být obezřetný, neklikat na podezřelé odkazy, dávat si pozor, zda nejsme na podvržené stránce apod., protože když je malware v systému jednou aktivní, je to problém.

Zdroje

1. *Desktop Operating System Market Share Worldwide* [online]. [cit. 2020-05-14]. Dostupné z: https://gs.statcounter.com/os-market-share/desktop/worldwide.

2. *Mobile Operating System Market Share Worldwide* [online]. [cit. 2020-05-14]. Dostupné z: https://gs.statcounter.com/os-market-share/mobile/worldwide.

3. *History of computer viruses: Creeper and Reaper.* [online]. [cit. 2020-05-14]. Dostupné z: https://pandorafms.com/blog/creeper-and-reaper.

4. *First Computer Virus* [online]. [cit. 2020-05-14]. Dostupné z: https://history-computer.com/Internet/Maturing/Thomas.html.

5. *SOPHOSLABS 2019 THREAT REPORT* [online]. [cit. 2020-05-14]. Dostupné z: https://www.sophos.com/en-us/medialibrary/pdfs/technical-papers/sophoslabs-2019-threat-report.pdf.

6. *AVTEST Security Report 2018-2019* [online]. [cit. 2020-05-14]. Dostupné z: https://www.av-test.org/fileadmin/pdf/security_report/AV-TEST_Security_Report_2018-2019.pdf.

7. *Ransomware now targeting back-up data, IT Threat Evolution Report Q3 2019 shows* [online]. [cit. 2020-05-14]. Dostupné z: https://www.kaspersky.com/about/press-releases/2019_ransomware-now-targeting-back-up-data-it-threat-evolution-report-q3-2019-shows.

8. *Ransomware WannaCry - co to je a jak ochránit vaše PC* [online]. [cit. 2020-05-14]. Dostupné z: https://www.avast.com/cs-cz/c-wannacry.

9. *Triada - truly scary mawlare for Android* [online]. [cit. 2020-05-14]. Dostupné z: https://www.kaspersky.com/blog/triada-trojan/11481/.

10. Qamar, Attia, Ahmad Karim a Victor Chang. *Mobile malware attacks: Review, taxonomy & future directions. Future Generation Computer Systems* [online]. 2019, 97, 887-909 [cit. 2020-05-14]. DOI: 10.1016/j.future.2019.03.007. ISSN 0167739X. Dostupné z: https:/.

11. *All about spyware* [online]. [cit. 2020-05-14]. Dostupné z: https://www.malwarebytes.com/spyware/.

12. *The State of Stalkerware in 2019* [online]. [cit. 2020-05-14]. Dostupné z: https://securelist.com/the-state-of-stalkerware-in-2019/93634/.

13. *Mobile malware evolution 2019* [online]. [cit. 2020-05-14]. Dostupné z: https://securelist.com/mobile-malware-evolution-2019/96280/.

14. *New FinSpy iOS and Android implants revealed ITW* [online]. [cit. 2020-05-14]. Dostupné z: https://www.kaspersky.com/blog/triada-trojan/11481/.

15. *KeyPass ransomware* [online]. [cit. 2020-05-14]. Dostupné z: https://securelist.com/keypass-ransomware/87412/.

16. Marczak, Bill, John Scott-Railton, Sarah McKune, Ron Deibert a Bahr Abdulrazzak, 2018. *HIDE AND SEEK Tracking NSO Group's Pegasus Spyware to Operations in 45 Countries*. 271 [online]. B.m.: 271. Dostupné z: doi:10.13140/RG.2.2.33325.95204.

17. *Cybersecurity Threatscape 2019* [online]. [cit. 2020-05-14]. Dostupné z: https://www.ptsecurity.com/ww-en/analytics/cybersecurity-threatscape-2019/.

18. Templeman, Robert & Rahman, Zahid & Crandall, David & Kapadia, Apu. (2012). *PlaceRaider: Virtual Theft in Physical Spaces with Smartphones.* .

19. S. Elmalaki, B. Ho, M. Alzantot, Y. Shoukry and M. Srivastava, *"SpyCon: Adaptation Based Spyware in Human-in-the-Loop IoT*," 2019 *IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, 2019, pp. 163-168, doi: 10.1109/SPW.2019.00039.

20. K. M. E. N. Mallikarajunan, S. R. Preethi, S. Selvalakshmi and N. Nithish, "*Detection of Spyware in Software Using Virtual Environment*," 2019 3rd *International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019, pp. 1138.

21. Hutchinson, Shinelle and Karabiyik, Umit, "*Forensic Analysis of Spy Applications in Android Devices*" (2019). *Annual ADFSL Conference on Digital Forensics, Security and Law*. 3.

22. Nagar, Sonali. *WhatsApp Hacking Wake-up Call for Users* [online]. [cit. 2020-05-14]. Dostupné z: http://nopr.niscair.res.in/handle/123456789/48957.

23. Zhang, Rongjunchen & Chen, Xiao & Lu, Jianchao & Wen, Sheng & Nepal, Surya & Xiang, Yang. (2018). *Using AI to Hack IA: A New Stealthy Spyware Against Voice Assistance Functions in Smart Phones.* .

24. D. Javaheri, M. Hosseinzadeh and A. M. Rahmani, "*Detection and Elimination of Spyware and Ransomware by Intercepting Kernel-Level System Routines*," in IEEE Access, vol. 6, pp. 78321-78332, 2018, doi: 10.1109/ACCESS.2018.2884964.

25. Pierazzi, Fabio, Ghita Mezzour, Qian Han, Michele Colajanni a V. S. Subramahian. *A Data-driven Characterization of Modern Android Spyware. ACM Transactions on Management Information Systems* [online]. 2020, 11(1), 1-38 [cit. 2020-05-14]. DOI: 10.1145/3382.

26. Lakhno, Valerii & Касаткін, Дмитро & Kozlovskyi, V. & Petrovska, S. & Boiko, Y. & Kravchuk, P. & Lishchynovska, N.. (2019). *A model and algorithm for detecting spyware in medical information systems. International Journal of Mechanical Engineering and Tec*.

27. *A definition of malware* [online]. [cit. 2020-05-14]. Dostupné z: https://www.bullguard.com/bullguard-security-center/pc-security/computer-threats/malware-definition,-history-and-classification.aspx.

28. D. Sukhram and T. Hayajneh, "*KeyStroke logs: Are strong passwords enough?*," 2017 *IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, New York, NY, 2017, pp. 619-625, doi: 10.1109/UEMCON.2017.8249051.

29. *The most common passwords 2020* [online]. [cit. 2020-05-14]. Dostupné z: https://rockit.cloud/2020/03/18/the-most-commonly-used-password-in-2020-is/.

30. *The Top 50 Worst Passwords of 2019* [online]. [cit. 2020-05-14]. Dostupné z: https://www.teamsid.com/1-50-worst-passwords-2019/.

31. *Most common passwords list* [online]. [cit. 2020-05-14]. Dostupné z: https://www.passwordrandom.com/most-popular-passwords,.

32. *The Best Password Managers for 2020* [online]. [cit. 2020-05-14]. Dostupné z: https://www.pcmag.com/picks/the-best-password-managers.

33. *The best password managers in 2020* [online]. [cit. 2020-05-14]. Dostupné z: https://www.tomsguide.com/us/best-password-managers,review-3785.html.

34. *The Best Password Managers to Secure Your Digital Life* [online]. [cit. 2020-05-14]. Dostupné z: https://www.wired.com/story/best-password-managers/.

35. *Pricing by Plan | LastPass* [online]. [cit. 2020-05-14]. Dostupné z: https://www.lastpass.com/pricing.

36. *What are security alerts and Dark Web Alerts, and what to do when I get one* [online]. [cit. 2020-05-14]. Dostupné z: https://support.dashlane.com/hc/en-us/articles/360000038180-What-are-security-alerts-and-Dark-Web-Alerts-and-what-to-do-when-I-get-one.

37. Chiasson, Sonia & Oorschot, P & Biddle, Robert. (2006). *A usability study and critique of two password managers. 15th USENIX Security Symposium.* .

38. *Stanford PwdHash* [online]. [cit. 2020-05-14]. Dostupné z: https://pwdhash.github.io/website/.

39. *LastPass Now Free On All Devices* [online]. [cit. 2020-05-14]. Dostupné z: https://helpdesk.lastpass.com/cs/lastpass-now-free-on-all-devices/.

40. *Auto-Type - KeePass* [online]. [cit. 2020-05-14]. Dostupné z: https://keepass.info/help/base/autotype.html.

41. L. Wang, Y. Li and K. Sun, "*Amnesia: A Bilateral Generative Password Manager,*" 2016 *IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Nara, 2016, pp. 313-322, doi: 10.1109/ICDCS.2016.90.

Příloha v IS EDISON

- Zdrojový kód skeneru (MaliciousCodeDirect.ps1)
- Záznam skeneru klávesnice (Spyware Results/Keylogs/)
- Záznam skeneru schránky (Spyware Results/Cliplogs/)
- Záznam skeneru monitoru (Spyware Results/Screens/)