Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Moderní techniky analýzy malware

Bakalářská práce

Autor: Radek Netolický
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Tomáš Svoboda, Ph.D.

Hradec Králové duben 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 12.4.2024 Radek Netolický

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Tomášovi Svobodovi, Ph.D. za metodické vedení práce a vstřícnost při konzultacích.

Anotace

Bakalářská práce se zaměřuje na moderní techniky analýzy malware pomocí statické a dynamické analýzy. V práci je popsána historie kybernetické bezpečnosti v České republice, také analýza zákona o kybernetické bezpečnosti a jsou zde vysvětleny různé modely informační bezpečnosti. Objasněny jsou také základní pojmy z oblasti analýzy rizik. Dalším klíčovým tématem této práce je malware, kde jsou popsány různé typy malwaru, jeho využití a techniky analýzy malware, konkrétně statická a dynamická analýza.

Klíčová slova: malware, hrozby, kybernetická bezpečnost, statická analýza, dynamická analýza

Annotation

Title: Malware analysis

This bachelor's thesis focuses on modern techniques of analysis malware using static and dynamic analysis. The thesis describes the history of cybersecurity in the Czech Republic, as well as an analysis of the Cyber Security Act and various models of information security are explained. The basic concepts of risk analysis are also explained. Another key topic of this thesis is malware, where the different types of malwares are described, their uses and malware analysis techniques, specifically static and dynamic analysis.

Key words: malware, threats, cybersecurity, static analysis, dynamic analysis

Obsah

Seznam obrázků

1 Úvod

V této době vyspělých digitálních technologií se kybernetická bezpečnost stává klíčovým prvkem pro ochranu jak jednotlivců, tak organizací před neustálými hrozbami. Ty představují kybernetičtí útočníci neboli hackeři. S každodenním propojením našich životů s internetem se tak zvyšuje i riziko útoků, které mohou mít ničivý dopad na soukromí, ekonomiku a společnost.

V rámci této problematiky je zaměřena pozornost na jednu z nejzávažnějších hrozeb, kterou je malware. Tento nebezpečný jev se neustále zdokonaluje a přizpůsobuje novým situacím, což vyžaduje neustálý vývoj metod a technik ochrany a analýzy.

Tato bakalářská práce si klade za cíl podrobně prozkoumat moderní techniky analýzy malware a jejich využití při ochraně před kybernetickými hrozbami. Zahrnuje jak teoretický pohled na historii kybernetické bezpečnosti v České republice a základní koncepty informační bezpečnosti, tak i praktické aplikace statické a dynamické analýzy malware.

V teoretické práci bude zaměřena pozornost na historii a vývoj kybernetické bezpečnosti v České republice, legislativní rámec v podobě zákona o kybernetické bezpečnosti a klíčové modely informační bezpečnosti. Dále je potřeba znát základní pojmy z oblasti analýzy rizik, které jsou zde také popsány a jsou důležité pro porozumění komplexnosti kybernetických hrozeb.

V praktické části bude řešena práce s konkrétními nástroji nebo programy, které jsou určeny pro statickou nebo dynamickou analýzu. Pro testování a analýzu bude využíváno virtuální prostředí, a to kvůli bezpečnému provádění experimentů a nenarušení našich osobních informací, či dat. Na základě těchto analýz budou odhaleny charakteristiky a chování různých typů malware.

2 Cíl práce

Cíle práce:

- popsat historii a vývoj kybernetické bezpečnosti v České republice
- popsat zákon o kybernetické bezpečnosti
- popsat základní modely informační bezpečnosti
- objasnit základní pojmy z oblasti analýzy rizik
- popsat kybernetické hrozby
- vysvětlit statickou a dynamickou analýzu
- vybudovat virtuální infrastrukturu pro analýzu malware
- otestovat a analyzovat vzorky malware v různých programech pro statickou a dynamickou analýzu

3 Teoretická část

3.1 Kybernetická bezpečnost

V dnešní uspěchané online době, kde se internet stal nedílnou součástí našeho života, si klademe otázku. Jak zabezpečit důležité osobní informace proti zneužití? Netýká se to jen osobních údajů, ale také zdravotních informací, kreditních karet či jiných citlivých údajů. S tímto úzce souvisí kybernetická bezpečnost. Tento pojem je definován jako *„souhrn právních, organizačních, technických a vzdělávacích prostředků směřujících k zajištění ochrany kybernetického prostoru*[1]*"* [13].

Kybernetickou bezpečnost je třeba nezanedbávat, protože se každým dnem zvyšuje digitalizace světa a tím i lepší možnosti pro kybernetické zločince. Zároveň se také rychle vyvíjí technika, což znamená neustálé aktualizování bezpečnostních pravidel a postupů.

3.1.1 Historie kybernetické bezpečnosti v České republice

První zmínka o kybernetické bezpečnosti v Evropě zazněla na pražském summitu NATO v roce 2002. Dne 19. října 2005 bylo vládou schváleno usnesení č. 1340 „o Národní strategii informační bezpečnosti České republiky a o zřízení Výboru pro informační bezpečnost České republiky". Hlavním cílem tohoto usnesení bylo zlepšení řízení informační bezpečnosti, rozvoj znalostí o informační bezpečnosti a podpora národní a mezinárodní spolupráce. Na toto usnesení bylo navázáno „Akčním plánem realizace opatření Národní strategie informační bezpečnosti České republiky a návrh nařízení vlády k realizaci úkolů stanovených Národní strategií informační bezpečnosti České republiky ze strany orgánů a organizací veřejné správy a subjektů kritické infrastruktury". V tomto dokumentu jsou popsány jednotlivé postupy k zajištění informační bezpečnosti v České republice. [1]

[1] Kybernetický prostor je definován jako *„digitální prostředí umožňující vznik, zpracování a výměnu informací, tvořené informačními systémy, a službami a sítěmi elektronických komunikací".* [13]

Až do roku 2007 mělo Ministerstvo informatiky České republiky na starosti řešení problematiky kybernetické bezpečnosti. V tomto roce došlo k jeho zrušení, a část mající na starost kybernetickou bezpečnost se sloučila s Ministerstvem vnitra. 15. března 2010 bylo přijato další usnesení č. 205 „o řešení problematiky kybernetické bezpečnosti České republiky“. Hlavním cílem bylo přenesení veškeré odpovědnosti pro oblast kybernetické bezpečnosti na Ministerstvo vnitra, které mělo mimo jiné na starost vytvoření Meziresortní koordinační rady pro oblast kybernetické bezpečnosti. Ta měla být hlavním koordinačním orgánem kybernetické bezpečnosti v České republice s cílem plnění řídících a koordinačních úkolů Ministerstva vnitra pomocí spolupráce a součinnosti se státními institucemi. [1]

Dne 9. prosince 2010 Ministerstvo vnitra České republiky a sdružení CZ.NIC[2] podepsaly „Memorandum o Computer Security Incident Response Team České republiky“. Do dubna následujícího roku mělo sdružení CZ.NIC za úkol zahájit plný technický provoz CSIRT.CZ (Computer Security Incident Response Team České republiky). Ze strany ministerstva šlo zejména o podporu CZ.NIC, hlavně potvrzením statutu CSIRT.CZ jako národního CSIRT týmu. [1, 3]

20. července 2010 vláda schválila usnesení č. 564 „Strategie pro oblast kybernetické bezpečnosti České republiky na období 2011–2015“ a „Akční plán opatření ke Strategii pro oblast kybernetické bezpečnosti České republiky pro období 2011–2015“. Toto usnesení bylo následně o rok později, dne 19. října, aktualizováno Národním bezpečnostním úřadem (NBÚ), protože Ministerstvo vnitra svým velkým finančním zatížením nezvládalo plnění úkolů. Neschopnost obsazení pracovních pozic zkušenými zaměstnanci se projevila jako dalším důvodem. Následkem přenesení veškeré odpovědnosti na NBÚ byla Meziresortní koordinační rada dne 19. října 2011 usnesením vlády zrušena. Tento krok hrál důležitou roli v budoucnosti kybernetické bezpečnosti ČR, a to zejména kvůli vzniku Národního

[2] CZ.NIC je zájmové sdružení právnických osob, které se stará především o provoz registru CZ domén a zabezpečování jejich provozu [4].

centra kybernetické bezpečnosti (NCKB). Toto centrum, sídlící v Brně, bylo součástí NBÚ. Hlavní úloha NCKB spočívala v koordinaci spolupráce jak na domácí, tak na mezinárodní scéně. Dále mělo aktivně přispívat k vytváření opatření pro řešení konfliktů a potlačování kybernetickým útokům. [1, 5]

NBÚ byl postaven před svoji počáteční výzvu – vypracovat návrh zákona o kybernetické bezpečnosti, jenž měl být přijat koncem roku 2013. Veřejnost měla možnost vyjádřit se k „Návrhu věcného záměru zákona o kybernetické bezpečnosti“, který byl volně dostupný na internetových stránkách NBÚ. 30. května 2012 byl návrh schválen vládou. [1, 2]

Určitou roli v historii kybernetické bezpečnosti v ČR hrálo také Ministerstvo obrany (MO) schválením dokumentu „Koncepce kybernetické obrany resortu Ministerstva obrany“. Resort začal spolupracovat jak s NBÚ, tak s NCKB s cílem provést úpravy v legislativě. Během této spolupráce byla rozeznána kritická obranná komunikační infrastruktura resortu. Kybernetickou bezpečnost na MO má na starost Odbor bezpečnosti MO (OB MO) s hlavním úkolem zabezpečit komunikaci a informace MO před kybernetickými útoky. Používána byla vysoce efektivní metoda „plánuj-dělej-kontroluj-jednej“ (Plan-Do-Check-Act – PDCA). [1, 6]

1. srpna 2017 vznikl Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB), a to oddělením NCKB od NBÚ. V tento datum se NÚKIB stává hlavním gestorem problematiky kybernetické bezpečnosti a přebírá tuto úlohu od již výše zmiňovaného NBÚ. Ten však stále zůstává ústředním správním orgánem pro ochranu utajovaných informací v různých oblastech bezpečnosti. Hlavní činností NÚKIB je předcházet zejména kybernetickým bezpečnostním incidentům, popřípadě následná reakce na ně. Dále řeší průzkum, vývoj nebo mezinárodní spolupráce v oblasti kybernetické bezpečnosti. NÚKIB vytvořil projekt nazvaný BIVOJ, který má za cíl zlepšit kybernetickou bezpečnost veřejného sektoru ČR. Název BIVOJ je odvozen od pěti slov, a to bezpečný, inovativní, pro veřejnou správu, odolný a jednotný. Projekt je zaměřen na vytvoření jednotné, funkční a bezpečné sdílené platformy. Ta bude poskytovat bezpečnostní a komunikační služby institucím veřejného sektoru, prvkům kritické informační sktruktury a dalším

základním službám ČR. Další důležitou činností je zveřejňování zranitelností, kde se jedná o proces, ve kterém pracovníci odkrývají jednotlivé zranitelnosti v informačních systémech vlastníků. Následně pak vytvoří report možné zranitelnosti a odesílají vlastníkovi s cílem opravit možné bezpečnostní riziko. [31, 32, 35]

3.1.2 181/2014 Sb. Zákon o kybernetické bezpečnosti

Ačkoliv se uživatel přihlásí do internetového bankovnictví, tak jeho provozovatel nesmí jeho přihlašovací údaje nikde zveřejňovat, ani je šířit. Ano, přesně tento příklad ilustruje, proč je důležitý Zákon o kybernetické bezpečnosti, který *„upravuje práva a povinnosti osob, jakož i pravomoc a působnost orgánů veřejné moci v oblasti kybernetické bezpečnosti.“* (NÚKIB, 2023) Platnosti nabyl 1. ledna roku 2015 a aktuální znění je z 6. srpna 2022 (verze 8). Původně se skládal ze šesti částí, z toho dvě byly zrušeny. Dnes je tedy pouze první, třetí, pátá a šestá část. Primárním cílem je zavedení nejdůležitějších opatření bezpečnosti, zdokonalení odhalení bezpečnostních událostí, upravení činnosti dohledových pracovišť a také nastolení hlášení bezpečnostních incidentů v kybernetice. [7, 8]

S tímto zákonem souvisí také 82/2018 Sb. Vyhláška o kybernetické bezpečnosti. Celkem se skládá z pěti částí. První část je pouze úvodní ustanovení, druhá se věnuje bezpečnostním opatřením, třetí se zaměřuje na kybernetické bezpečnostní incidenty a čtvrtá část pojednává o reaktivních opatřeních. V poslední části se pak nachází závěrečná ustanovení a datum nabytí účinnosti. Druhá část se dále dělí na dvě hlavy. V hlavě první jsou uvedeny organizační opatření:

- Systém řízení bezpečnostních informací
- Řízení aktiv
- Organizační bezpečnost
- Bezpečnostní role
- Řízení dodavatelů
- Bezpečnost lidských zdrojů
- Řízení provozu a komunikací

- Řízení změn
- Řízení přístupu
- Akvizice, vývoj a údržba
- Zvládání kybernetických bezpečnostních událostí a incidentů
- Řízení kontinuity činností
- Audit kybernetické bezpečnosti

Hlava druhá se pak zabývá technickými opatřeními, jako jsou:

- Fyzická bezpečnost
- Bezpečnost komunikačních sítí
- Správa a ověřování identit
- Řízení přístupových oprávnění
- Ochrana před škodlivým kódem
- Zaznamenávání událostí informačního a komunikačního systému, jeho uživatelů a administrátorů
- Detekce kybernetických bezpečnostních událostí
- Sběr a vyhodnocování kybernetických bezpečnostních událostí
- Aplikační bezpečnost
- Kryptografické prostředky
- Zajišťování úrovně dostupnosti informací
- Průmyslové, řídicí a obdobné specifické systémy
- Digitální služby

Součástí této vyhlášky je celkem osm příloh, ve kterých je uvedeno např. jakým způsobem mazat data, jaké jsou hrozby a zranitelnosti nebo obsah bezpečnostní dokumentace. [30]

Razantní změnu v oblasti kybernetické bezpečnosti přinese nová směrnice Evropské Unie o kybernetické bezpečnosti NIS2 a návrh nového zákona o kybernetické bezpečnosti. Tato platnost změn je naplánována na rok 2024. NÚKIB se rozhodl vydat cestou úplně nového zákona, namísto obnovy toho starého. Nově se rozšíří povinný okruh subjektů, touto směrnicí se totiž bude řídit až 6000

subjektů, což je 15x více oproti současné verzi zákona. Firmy se budou rozdělovat podle velikosti na režim vyšších a nižších povinností. Jednotlivé režimy definují úroveň opatření, který subjekty musí plnit. [36]

Jedná se o poskytovatele, kteří poskytují služby v těchto oblastech:

- Energetika
- Doprava
- Bankovnictví
- Zdravotnictví
- Infrastruktura finančních trhů.
- Pitná voda
- Odpadní voda
- Digitální infrastruktura
- Veřejná správa
- Poštovní služby
- Potravinářství
- Chemická a farmaceutická výroba
- Poskytovatelé digitální infrastruktury a digitálních služeb
- Letectví

Celkově má nová směrnice NIS2 posílit bezpečnostní požadavky na jednotlivé subjekty. Podle původního zákona měly členské státy možnost přizpůsobení bezpečnostních požadavků, a to vedlo k větším zranitelnostem. NIS2 eliminuje riziko narušení kritických služeb, tedy stanovením jasných pravidel, které musí dodržovat všechny členské státy. Subjekty lze také rozdělit na kritické a klíčové. Kritické subjekty budou pod neustálým dohledem, protože narušení jejich služeb by mohlo vést k ekonomickým ztrátám státu, či ohrožení státní bezpečnosti. Zatímco klíčové subjekty budou pod dohledem až tehdy, když poruší předepsané bezpečnostní předpisy. Směrnice NIS2 nově zavádí povinnost hlásit všechny kybernetické incidenty, včetně těch, které nemají okamžitý dopad na provoz subjektu. Tato povinnost je důležitá proto, aby kontrolní orgány mohly lépe monitorovat potenciální hrozby a následně na ně reagovat. Subjekty tedy musí dbát

na bezpečnost zejména v těchto oblastech: hodnocení a řízení rizik, pravidelné školení, šifrování dat nebo řešení a hlášení zranitelností a incidentů. [37, 38]

3.2 Modely informační bezpečnosti

3.2.1 CIA triáda

Tento model je nejstručnější a nejuniverzálnější metodou, které se snaží dosáhnout každý v oblasti informační bezpečnosti. Pojem CIA vznikl složením tří anglických slov – Confidentality (Důvěrnost), Integrity (Integrita), Availability (Dostupnost). [9]

Pokud není uvedeno jinak, následující informace vychází ze zdroje [10].

Důvěrnost je princip bezpečnosti, který má za úkol, aby se informace nedostaly k neoprávněným osobám, procesům nebo zařízením. Do toho spadá i komunikace mezi odesílatelem a příjemcem, včetně uložených dat.

Integrita je princip bezpečnosti zajišťující přesnost, úplnost a ochranu před neoprávněnými modifikacemi. Udržuje se během zpracování, přenosu a ukládání, aby se předešlo neoprávněným změnám. Velký důraz na integritu dat klade zejména finanční průmysl, kvůli velmi citlivým informacím, se kterými je zde pracováno.

Poslední princip bezpečnosti, nazývaný **dostupnost**, se stará o včasný a spolehlivý přístup k informacím, se kterými je potřeba pracovat. Dostupnost je z těchto tří elementů považována za nejdůležitější.

Každý prvek CIA triády je nějak propojený s jiným, všechny na sebe totiž navazují. Důvěrnost údajů není jen o bezpečnosti, ale také o poskytování údajů osobám k tomu oprávněným. Správná autentizace závisí na dostupnosti autentizačních a autorizačních služeb.

Obrázek 1: CIA triáda. Zdroj: vlastní, překresleno podle [33]

3.2.2 Parkerian Hexad

Pokud není uvedeno jinak, následující informace vychází ze zdroje [11].

Donn B. Parker nyní pracuje jako konzultant a výzkumník informační bezpečnosti. Do historie informační bezpečnosti se zapsal v roce 2002, kdy představil novou verzi CIA modelu. Jeho model obsahoval tři nové elementy, celkem tedy šest – Confidentality, Integrity, Availability, Utility (Užitečnost), Possession/Control (Majetek/Kontrola), Authenticity (Autenticita). Lze si tedy všimnout, že původní tři elementy CIA modelu zůstaly opravdu beze změny. Díky vynálezu tohoto modelu je Parker uznáván jako velký průkopník v oblasti informační bezpečnosti. Mimo jiné o této oblasti napsal i spoustu prodávaných knih.

A teď tedy odpověď na otázku „Proč se vlastně Parker rozhodl vymyslet novou verzi CIA modelu?“ Domníval se totiž toho názoru, že původní verze modelu je příliš prostá a zranitelná pro větší aplikace. Dále se zabývá hlavně technologií, která chrání informační majetek, ale klade málo důrazu na lidi. To byla jeho hlavní myšlenka. Chtěl, aby se na jeho prvky dívalo v těchto skupinách: důvěrnost a vlastnictví, integrita a autenticita, dostupnost a užitečnost. Domníval se, že jedině takto je model správně chápán a realizován.

Prvky Parkerian Hexad modelu, které jsou stejné jako v CIA modelu, jsou už objasněné o kapitolu výše. Následující obsah této kapitoly tedy popisuje ty prvky, které Parker přidal k původnímu CIA modelu.

Majetek/Kontrola je prvek sloužící k ochraně před nápadem, že osoby, které k tomu nemají oprávnění, mohou vlastnit a kontrolovat citlivá data, aniž by porušili důvěrnost. Mimo jiné se zabývá i ochranou veřejných dat, která mohou být vlastněna. Podle Parkera je tento prvek zásadní pro život, protože obsahuje porušení, kde je důvěrnost podstatná a neexistující.

Autenticita se týká jistoty, že zpráva, transakce či jiná výměna informací je opravdu ze zdroje, o kterém tvrdí, že je. Autenticita jako taková je obvykle ověření uživatele pomocí jeho uživatelského jména a hesla neboli prokázání totožnosti [12]. Tento prvek je v dnešní době určitě velice důležitý, protože je nutné vědět, s kým si na internetu či sociálních sítích uživatel sděluje informace, vyměňuje data apod. Zabráněním vstupu na neověřenou webovou stránku je užitečné používat digitální certifikáty, které prokáží identitu společnosti, která je dostupná na svých webových stránkách.

Užitečnost je posledním přidaným prvkem Parkerian Hexadu. Nelze ji opomenout jako nedílnou součást tohoto modelu. Data jsou nepoužitelná, pokud nejsou v použitelném stavu či formě.

Obrázek 2: Parkerian Hexad. Zdroj: vlastní, překresleno podle [34]

3.3 Analýza rizik – základní pojmy

Mějme firemní síť s vlastní databází, která obsahuje zranitelná a velmi citlivá firemní data. Pracovníci, kteří mají na starost zabezpečení, například nedostatečně proškolili ostatní zaměstnance, aby si nastavili silné heslo nebo mají slabé zabezpečení firewallu. Toho může jednoduše využít kyberzločinec neboli hacker, který bude chtít ukrást citlivá firemní data a je to pro něj mnohem jednodušší než u nějaké jiné firmy, která má velmi silné zabezpečení. Když se hacker dostane do firemní sítě a získá přístup k citlivým datům, firmu může dostihnout finanční deficit či ztráta jejího dobrého jména. Tomu se dá samozřejmě předejít dvou faktorovým ověřením, pravidelnou aktualizací softwaru, šifrováním dat nebo častým školením zaměstnanců.

Na tomto příkladu firemní síť představuje tzv. **aktivum**, což je všechno, co nese cenu pro jednotlivce, instituci nebo veřejnou správu. Slabá hesla nebo slabé zabezpečení firewallu jsou **zranitelnosti**. Tento pojem lze definovat jako *„slabé místo aktiva nebo opatření, které může být využito jednou nebo více hrozbami“.* Hacker, který se snaží prolomit firemní síť a ukrást data, je to tedy **hrozba** neboli potenciální původ nechtěných událostí, které mohou způsobit poškození systému nebo organizace. **Riziko** je v našem případě ztráta pověsti firmy, finanční deficit a další důsledky po ukradení citlivých dat. Riziko představuje *„Nebezpečí, možnost škody, ztráty, nezdaru. Účinek nejistoty na dosažení cílů. Možnost, že určitá hrozba využije zranitelnosti aktiva nebo skupiny aktiv a způsobí organizaci škodu.“* Předejitím hrozby výše uvedenými příklady se nazývá **opatření** neboli *„prostředky modifikující riziko, včetně politik, strategií, postupů, směrnic, obvyklých postupů (praktik) nebo organizačních struktur, které mohou být administrativní, technické, řídící nebo právní povahy“*. [13]

3.4 Malware

Jedná se o závažnou hrozbu pro jednotlivce i organizace všech velikostí. Tyto škodlivé programy mají schopnost zneužít zranitelnosti v bezpečnostním systému a způsobit závažné následky jako například ztrátu dat nebo finanční škodu. Přítomnost malwaru tedy představuje spoustu rizik, a proto je třeba proti němu

provádět různá opatření. Pojem malware vznikl z dvou anglických slov – malicious (škodlivý) a software [28]. Uživateli určitě pomůže informovanost v této oblasti, proto jsou v této kapitole objasněny jednotlivé typy malwarů, na co si dát převážně pozor a jaký je vlastně důvod pro hackery vykonávat tuto nelegální činnost.

3.4.1 Typy malwaru

Dnes už je opravdu hodně druhů různých malwarů. Pro analytika je mnohem jednodušší odhadnout jednotlivé chování škodlivých programů, pokud zná jednotlivé typy. Pomocí nich je možné alespoň trochu odhadnout, jak se daný malware bude chovat. [14]

3.4.1.1 Adware

Tento typ malwaru dokáže být velmi dotěrný, zejména pokud člověk nutně potřebuje pracovat na počítači či ho jinak využívat. Jedná se o neustálé vyskakování reklam. Jejich cílem je následné kliknutí uživatele na reklamu, která často přesměrovává na stránku, kde se nachází malware. Tyto reklamy mohou vypadat velmi důvěryhodně díky sledování aktivity uživatele a následné vytvoření stejné nebo podobné reklamy. Na rozdíl od spywaru se instaluje do počítače s povolením uživatele. Adware není tak složitý rozeznat, mezi identifikátory patří zpomalení webového prohlížeče, přesměrovávání na podezřelé stránky, vyskakovací reklamy na známých stránkách či změna domovské stránky. [19, 20]

3.4.1.2 Botnet

Bot je předem naprogramovaný software, který má kontrolu nad cílovým zařízením. Cílem Botnetu je konat nežádoucí aktivitu na infikovaném zařízení, aniž by o tom vlastník věděl, např. krádež dat či rozesílání nevyžádaných zpráv. [19]

V případě Botnetu se rozlišují dvě architektury, a to model klient-server, který botům povoluje ovládat vše na dálku a maskovat provoz. Nic netušící klienti jsou na předem určené lokalitě, kam jsou pomocí botů posílány příkazy ze serveru. Když je příkaz proveden, je automaticky odeslán zpět botům. Častým cílem jsou právě IoT zařízení, která jsou velmi slabě zabezpečená a často nemají aktualizovaný systém. Do druhé architektury se řadí model peer-to-peer. Centrální server většinou dokáže

zachytit botnetové útoky, který právě v tomto modelu chybí. Za server se vydávají boti. [22]

3.4.1.3 Červ

Tento druh vytváří a šíří své duplikáty v původním nebo jiném formátu. Jedná se o jeden z prvních vynalezených škodlivých kódů vůbec, který nemusí být ničivý. Jakmile se ale šíří duplikací do jiných systémů bez připojení k ostatním souborům, pak lze snadno určit, že je škodlivý. Záměr červa může být velmi různorodý, záleží totiž k čemu ho útočník naprogramoval. Většinou se však jedná o šíření nevyžádané pošty nebo odepření jednotlivých služeb. [24]

Způsob šíření červů je identický s virusem, který bude popsán v pozdější kapitole. Tedy pomocí sítě nebo paměťových médií. Většinou jimi uživatel nakazí počítač pomocí infikované přílohy e-mailu, odkazu na nežádoucí webovou stránku nebo stáhnutím nakaženého souboru. To, že se v počítači nachází červ, lze poměrně jednoduše poznat. Zařízení je hodně zpomalené, někdy nereaguje vůbec. To je způsobeno velkým vytížením systémových prostředků. Odhalení a následné smazání červa dokáže antivirový program. Předejít infekci je možné například ostražitostí při otevírání/stahování podezřelých příloh, odkazů nebo souborů, aktualizací antivirového programu a používáním firewallu. [25]

3.4.1.4 Downloader

Tento škodlivý program je v infikovaném zařízení určen pouze pro stahování a instalování dalších malwarů. Často se instalují, když útočník poprvé získá přístup do systému. Mohou být součástí dalších malwarů. [14]

3.4.1.5 Ransomware

Ransomware je dnes nejrozšířenější hrozbou, se kterou se lze setkat. Jedná se o velmi agresivní druh škodlivého programu, zabrání totiž uživateli zcela používat počítač zašifrováním dat či zastavením některých aplikací. Hackeři poté chtějí po napadených uživatelích výkupné ve formě zaplacení určité částky. Avšak ani po

zaplacení výkupného není stále jisté, že s tím hackeři skoncují, proto se doporučuje výkupné neplatit. [19]

Odhalit, zda je váš počítač nakažený ransomwarem, není žádná věda. Většinou se jedná o vyskakovací okna, případně textové soubory se zprávou o výkupném. Další vlastností je měnění přípony souborů v zařízení. Možností, jak se dostane ransomware do počítače, je poměrně hodně. Mezi ty nejčastější patří např. virem napadené přílohy e-mailu nebo vstup na zavirovanou webovou stránku. Ke znemožnění přístupu do vašeho zařízení využívají útočníci třeba Screen locker (uzamknutí obrazovky, blokování přístupu) nebo PIN locker, který dokáže změnit stávající PIN kód, používaný na povolení přístupu do zařízení. [23]

Proti ransomwaru se lze chránit mnoha způsoby, zejména častým zálohováním dat, aktualizováním operačního systému a antivirového programu nebo používáním VPN. [23]

3.4.1.6 Rootkit

Další z malwarů, který vnikne do počítače bez vědomí vlastníka se nazývá Rootkit. Je naprogramovaný velmi chytře, jeho vlastností je totiž skvělé skrývání před antivirovým programem, který ho pak považuje za neškodný program. Z toho vyplývá, že Rootkit je velmi těžký, co se týče odhalení. Po vniknutí do počítače převezme kontrolu nad operačním systémem, to mu umožňuje již zmíněné skrývání či vytvoření bezpečného prostředí pro jiný škodlivý program. [19]

Do zařízení se může dostat několika způsoby, např. pomocí rozšíření aplikací třetích stran či spolu s bezpečnostními programy. Rootkity nemají vlastnost reprodukce samy sebe. Jak už bylo zmíněno, tento typ malwaru je složité odhalit, i přes to ho ale antivirový program dokáže najít. Je třeba, aby důkladně prověřil všechny systémové procesy a jejich závislosti. Odstranění se pak provádí ručně, nikoliv pomocí antivirového programu. Vniknutí Rootkitu do počítače lze zabránit nainstalovaným antivirovým programem s funkcí antirootkit. [29]

3.4.1.7 Spyware

Jeho hlavním cílem je sledování aktivity na infikovaném zařízení a krádež citlivých osobních informací a dat. Bez vědomí vlastníka se shromažďují informace, které jsou následně posílány útočníkovi. S tímto malwarem souvisí soubory cookies, obsahující informace, jež se uchovávají ve webovém prohlížeči na budoucí použití. [19]

Toto je problém, jestliže spyware je natolik „chytrý“, že dokáže mít přístup i k těmto informacím. Následkem přítomnosti spywaru v počítači bývá obvykle zpomalení rychlosti internetu. Tento malware má několik druhů, jako např. keylogger, který sleduje aktivitu stisknutých znaků na klávesnici a následně může zjistit třeba přihlašovací údaje. Sniffer je dalším druhem, sleduje veškerou síťovou aktivitu a ukládá ji. Password stealer potom stahuje přihlašovací údaje, které uživatel použil ve webových prohlížečích. [21]

Tento typ malwaru má schopnost shromažďovat širokou škálu informací či údajů, jako jsou informace o aktivitě na zařízení, osobní data, bankovní údaje nebo přihlašovací údaje. Tomuto lze předejít výběrem antivirového programu, který obsahuje modul antispyware, detekující, zda se v počítači nachází nějaký sledovací program, v kladném případě pak jeho odstranění. [21]

3.4.1.8 Trojský kůň

Trojský kůň neboli také „Trojan“ svým chováním vypadá neškodně, jako užitečný program. Ve skutečnosti má ale škodlivý záměr. Na rozdíl od červů a virů se neduplikuje, do počítače se dostane stáhnutím z internetu. [19]

To, že vypadá neškodně, je způsobeno ukrytím v jiném programu. Proto také většinou uživatel spustí program bez vědomí, že může obsahovat nějaký škodlivý program. Trojský kůň patří do kategorie jménem Backdoor. Jak z názvu vyplývá, viry v této kategorii využívají tzv. „zadní vrátka“ k přístupu do zařízení. Trojský kůň následně většinou převezme kontrolu nad zařízením, stáhne uživatelská data nebo jiný malware. Některé antivirové programy mají funkci antitrojan, která dokáže upozornit na Trojský kůň. [26, 27]

3.4.1.9 Virus

Hlavním znakem viru je jeho závislost na ostatních souborech nebo aplikacích, nedokáže totiž existovat samostatně, a tak se k nim připojuje a následně duplikuje. [19]

Může v sobě ukrývat části, které se aktivují při vyhovění některé podmínky na zařízení, kde je provozován. Přenáší se jak po internetu – v e-mailu, stáhnutím infikovaného programu z webové stránky, tak i pomocí přenosných paměťových médií. Jeho hlavním cílem je získání dat, zcizení identity či poškození počítače.
Jelikož využívá spoustu systémových procesů, počítač se velmi zpomalí. Odstranění je možné opět pomocí antivirového programu. Lze to uskutečnit i ručně, je to ale velmi náročný proces vyžadující obsáhlé znalosti a zkušenosti. [13, 19]

3.4.2 Využití malwaru

Dříve, když počítače teprve začínaly, tak malware nepůsobil tolik škody jako v dnešní době. Na svém začátku počítače nebyly tolik výkonné, to zabraňovalo rychlému šíření. Lidé neměli tolik zkušeností vytvořit nějaký škodlivý program a prolomit tím systém. Cílem nainstalování škodlivého programu do počítače je mnoho, ale všechny mají podobný záměr. Nejčastěji se jedná o odcizení uživatelských jmen, hesel, údajů o platebních kartách či bankovních údajů. Toto má za následek odcizení peněžní částky, krádež identity a další trestné činy. Hackeři, s vidinou vysokého finančního obnosu, nedbají na vysoké tresty za tuto činnost a počet obětí tak stále roste.

3.5 Techniky analýzy malware

Analýza malwaru se dá definovat jako způsob rozebírání škodlivého softwaru tak, abychom ho mohli následně pochopit a nejlépe odstranit. Práce na pozici malware analytika může být velice složitá, díky hackerům, kteří pořád vymýšlejí nové a nové techniky, kterými se snaží překazit analýzu. Je tedy důležité, aby se analytici malwaru snažili tyto techniky pochopit, porazit je a jít dále s vývojem v oblasti analýzy malwaru. M. Sikorski a A. Honig doporučují nesnažit se pochopit každý detail malwaru, ale spíše ho chápat jako celek, protože se většinou jedná o velké

a složité programy. Každý škodlivý program je také něčím specifický a jednoznačný. Proto, když se nebude dařit ho odhalit, je vhodné ho začít zkoumat v jiném nástroji, přesunout se k odlišnému problému či ho začít zkoumat z různého pohledu. M. Sikorski a A. Honig naopak nedoporučují zůstávat příliš dlouho na jednom problému. [14]

Hlavním cílem analýzy škodlivého softwaru je zjištění, kdy došlo k napadení, co přesně se stalo a najití všech nakažených souborů. Je třeba také zjistit, jaké škody může podezřelý binární soubor napáchat, jak ho odhalit v počítačové síti, následně měřit a zamezit zhoršení stavu. Způsoby, jak analyzovat malware se rozdělují na statickou a dynamickou analýzu. [14]

3.5.1 Statická analýza

Analýza malwaru, která se provádí, aniž by byl škodlivý program spuštěn, se nazývá statická analýza. Týká se procesu rozboru kódu nebo struktury programu za účelem určení jeho funkčnosti [14]. Spustitelný soubor musí být před analýzou extrahován a dešifrován. Používají se zde tzv. techniky binární obfuskace, které dělají binární kód složitějším na pochopení. Také se využívají k zašifrování nebo dešifrování dat. Tyto techniky jsou velmi drahé a nespolehlivé, a to je jedna z hlavních nevýhod statické analýzy. Další problém při využití binárních souborů je ztráta informací – velikost proměnných nebo datových struktur. Tyto nedostatky statické analýzy hackeři brzy odhalili a prolomili. Proto vznikla dynamická analýza, která je popsána v kapitole 4.5.2. Statickou analýzu můžeme rozdělit podle složitosti rozboru, konkrétně na základní a pokročilou. [15, 16]

3.5.1.1 Základní statická analýza

Pokud není uvedeno jinak, následující informace vychází ze zdroje [14].

Základní statická analýza je obvykle první krok, který se používá k odhalení viru zejména proto, že je jednoduchá a může být i rychlá. Jedná-li se o jednodušší malware, tato technika ho dokáže odhalit a potvrdit škodlivost souboru. Naopak, když se jedná o složitější a dobře promyšlený malware, tak základní statickou analýzou můžeme přehlédnout jeho důležité chování.

Jako první krok této metody je samozřejmě to nejjednodušší řešení, a to skenování zařízení antivirovým programem. Je doporučeno provést skenování více než jedním programem, protože co jeden přehlédne, druhý může označit za škodlivý soubor. Hlavní důraz je kladen na databázi identifikovatelných částí podezřelého kódu, také nazýváno jako signatury souborů. Ty ale dnešní hackeři umí „obejít", a proto po skenování antivirovým programem nemusí být škodlivý soubor vůbec odhalen. Dalším důvodem neodhalení malwaru antivirovým programem je možnost, že není veden v databázi, tudíž je nový nebo vzácný. Existuje osvědčená bezplatná webová stránka VirusTotal[3], kam lze nahrát podezřelý soubor, URL adresu, hash kód, IP adresu a další. Následně je provedeno skenování cca 40 antivirovými programy a vypsání možných hrozeb. Je zde třeba zmínit hrozbu, protože všechny soubory na této webové stránce zůstávají uloženy a kdokoliv si je může stáhnout do svého zařízení.

Další technikou základní statické analýzy je tzv. hashování. Jedná se o vytvoření jedinečného kódu (hashe), který je následně vložen do již výše zmiňované webové stránky VirusTotal a je vytvořen výpis možných hrozeb.
Poněkud jednoduchým způsobem je prohledávání řetězců (Stringů). Lze tak odhalit několik základních způsobů hrozeb jako připojení k URL, lokace přístupové cesty či výpis zprávy. Například, když program přistupuje k URL, zobrazí se právě jako řetězec.

Všechny tyto techniky se nezajímaly o formát souboru. Ten nám ale může ukázat mnohé o možnostech programu. Spustitelné soubory operačního systému Windows používají formát souborů PE (Portable Executable). *„Formát souboru PE je datová struktura, která obsahuje informace, které potřebuje zavaděč operačního systému Windows ke správě zabaleného spustitelného kódu."* Důležité je, že tyto soubory začínají hlavičkou, ve které je obsaženo spoustu důležitých informací, jako jsou informace o kódu nebo typ aplikace.

[3] https://www.virustotal.com/gui/home/upload

S formátem PE také souvisí pojem linkování, což je metoda, pomocí níž se hledá výpis funkcí, které importuje. Importy jsou funkce, které se používají v jednom programu, ale ve skutečnosti jsou uloženy v jiném programu, jako jsou kódy knihoven. Ty mají funkce, které jsou společné mnoha programům. Název linkování odvozen kvůli následnému propojení těchto knihoven ke spustitelnému souboru. Typy linkování kódu knihoven se rozdělují na statické a dynamické. Pro pochopení malwaru je důležité vědět, jakým typem je kód knihovny propojen. Závisí na tom totiž informace obsažené v hlavičce PE souboru.

3.5.1.2 Pokročilá statická analýza

Tato technika, jak už z názvu vyplývá, je složitější než základní statická analýza. Pokročilá statická analýza využívá disassembler, což můžeme označit jako nástroj, který převádí zkompilovaný program zpět do zdrojového kódu [17]. Do disassembleru se načte spustitelný soubor, instrukce programu se následně prohlížejí, aby se zjistilo, co program dělá. Jelikož instrukce provádí procesor, pokročilá statická analýza umožní přesně vědět, co daný program dělá. Podobně jako u základní statické analýzy i zde existují různé techniky pro odhalení škodlivého programu. [14]

Počítačové systémy lze reprezentovat více úrovněmi abstrakce, které poskytují způsoby, jak skrýt detaily implementace. V následujících bodech budou stručně představeny jednotlivé úrovně abstrakce.

- **Hardware** – Tato úroveň je jediná fyzická. Tvoří ji elektrické obvody, které jsou implementovány složitými kombinacemi logických operátorů (XOR, AND, OR a NOT).
- **Firmware** lze chápat jako program, který je funkční pouze na takovém elektrickém obvodu, pro který byl vyprojektován.
- **Strojový kód** – V této úrovni jsou obsaženy hexadecimální číslice, podle kterých procesor vykonává instrukce. Strojový kód se vytváří až po následném zkompilování programu, napsaného ve vyšším programovacím jazyce.

- **Nižší programovací jazyky** – Na začátku této kapitoly byl zmíněn disassembler, který je přesným opakem assembleru. Tedy nejznámějšího nižšího programovacího jazyku. Ten povoluje psát přímo ve strojovém kódu procesoru. Disassembler zde hraje svou roli k vytvoření textu.
- **Vyšší programovací jazyky** – Patří sem např. C, C++, Pascal a další. Struktura je logická a je následně převedena kompilací do strojového kódu.
- **Interpretované jazyky** – Oproti vyšším programovacím jazykům je zde kód přeložen do bajtového kódu. Nejznámějšími jsou např. Java nebo C#. Je zde využíván interpret, který lze chápat jako program, který za běhu překládá bajtový kód do strojového.

Reverzní inženýrství je dalším důležitým pojmem pokročilé statické analýzy. Jakmile se už malware nachází v našem počítači, vyskytuje se obvykle v binární podobě na úrovni strojového kódu. Při rozboru malwaru je na vstupu použit jeho binární kód a na výstupu vygenerovaný kód v assembleru. [14]

Von Neumannova architektura je základem většiny současných počítačových architektur. Skládá se ze tří částí hardwaru, kde kód vykonává procesor a paměť RAM ukládá všechna data a kód. Další zařízení jako například myš, klávesnice nebo monitor, se řadí do takzvaných vstupních a výstupních zařízení systému. Paměť RAM lze dále rozdělit na čtyři jednotlivé mikro části:

- **Data** jsou základním stavebním prvkem v datové sekci paměti. Nacházejí se zde hodnoty, které jsou používány při načítání programu. Říká se jim statické hodnoty, protože se během chodu programu nemusí měnit, nebo také globální hodnoty, a to kvůli přístupnosti všem součástem programu.
- **Kód** obsahuje instrukce, které jsou třeba k provedení úkolů programu. Načítá je procesor. Hlavní funkcí kódu je kontrola chování programu a organizace pořadí, ve kterém budou úlohy spuštěny.
- **Halda** souvisí s dynamickou pamětí během provádění programu, protože obsah této paměti se může měnit během chodu programu. Vytváří nové hodnoty, a naopak odstraňuje ty, které program již nepotřebuje.

- **Zásobník** slouží jako paměť pro funkce, řízení toku a lokální proměnné. Jedná se o datovou strukturu mající dvě funkce – push a pop. Funkce push přidává prvky do zásobníku, a to vždy na jeho vrchol. Funkce pop se pak řídí pravidlem LIFO (Last In First Out) a odebírá prvky, které byly vloženy jako poslední, tedy na vrcholu zásobníku. [14]

3.5.2 Dynamická analýza

Na rozdíl od statické analýzy se v dynamické dělá rozbor malwaru přímo za chodu aplikace. Toto se doporučuje dělat ve virtuálním prostředí, aby zařízení a data v něm nebyla nijak poškozena. Má to ale jednu nevýhodu, ve virtuálním prostředí může mít malware jiné chování než ve skutečném. Znamená to teda, že ve virtuálním prostředí nelze identifikovat za jakých podmínek se spouští ve skutečném. Dnes je mnoho nástrojů, pomocí kterých se provádí dynamická analýza. Některé z nich jsou představeny v praktické části. Mezi nejznámější techniky patří sledování hovorů, toku informací nebo analýza funkčních parametrů. Zatímco u statické analýzy je třeba rozebrání PE souboru, u dynamické tomu tak není. To je jeden z důvodů, proč je tato analýza mnohem efektivnější, ale o to mnohem náročnější, co se času týče. [15]

Dynamická analýza se provádí až po následném zkrachování základní statické analýzy. Je zde totiž riziko v podobě ohrožení sítě a systému. Na rozdíl od statické analýzy může dynamická analýza sledovat škodlivý program a jeho vývoj. Dynamická analýza je také účinný způsob, jak identifikovat funkčnost malwaru. [14]

3.5.2.1 Základní dynamická analýza

První technikou základní dynamické analýzy jsou tzv. sandboxy. Lze je nazvat jako „bezpečné prostředí“. Umožňují tedy spustit škodlivý program bez obavy následné infekce zařízení. Velkou výhodou těchto sandboxů je, že modelují síťové služby, což znamená správné fungování testovaného malwaru. Dále umožňují odeslání malwaru na jejich webové stránky, a to je velmi skvělé řešení, jak mít přehled o co nejvíce typech škodlivých programů. Sandboxových programů existuje opravdu mnoho, mezi nejznámější patří např. FortiSandbox, Kaspersky Sandbox, ESET

PROTECT Advanced a GFI Sandbox. Pro domácí použití jsou ale velmi drahé. I když předchozí tvrzení mohou znít jako že sandboxy jsou naprosto bezchybné, bohužel tomu tak není. Mezi největší nevýhody patří např. pokud je malware nastaven na noční režim spánku. Sandbox to nepozná a na aktivitu škodlivého programu se bude muset čekat. Dnešní malwary dokáží rozeznat, jestli jsou spuštěné ve virtuálním prostředí, mohou zde mít tedy odlišné chování. Klíče registrů také obvykle nejsou ve virtuálním prostředí, přestože ho některé malwary vyžadují. Sandboxy mají předem nainstalovaný daný operační systém, který nemusí být shodný s operační systémem, na kterém má být malware spuštěn. Toto lze opět označit za nežádoucí chování. [14]

Monitorování pomocí nástroje Process Monitor je další technikou. Tento nástroj, určený pro operační systém Windows, dokáže sledovat určité registry, síť, procesy a aktivity vláken. Neumí ale zachytit např. chování ovladače zařízení, který komunikuje s rootkitem pomocí ovládacích prvků vstupu nebo výstupu. Problémem tohoto nástroje je velké využití paměti RAM, díky zachycení všech systémových volání, což může mít za následek pád virtuálního počítače. [14]

Další možností základní statické analýzy je technika nazývaná falešná síť. Pro vytvoření falešné sítě je důležité, aby malware nezjistil, že je spouštěn přes virtuální prostředí. Tato technika spočívá v odhalení síťových indikátorů – název DNS nebo IP adresy, bez připojení zařízení ke skutečné síti. Správným nastavením sítě virtuálního počítače bude mnohem pravděpodobnější zvýšení možností dosažení úspěchu. [14]

3.5.2.2 Pokročilá dynamická analýza

Tento typ analýzy je opět použit po nedostatečném odhalení malwaru ve výše uvedených typech. Je třeba objasnit pojem „dynamic disassembly“ (dynamická demontáž). Lze ji definovat jako *„analýzu binárního kódu jeho spuštěním v kontrolovaném prostředí, například ve virtuálním stroji nebo ladicím programu.“* Dynamickou demontáž lze realizovat pomocí debuggerů či emulátorů. Pokročilá dynamická analýza je vysoce efektivní a spolehlivá cesta pro odhalení škodlivého programu, navzdory tomu ale vyžaduje rozsáhlé znalosti a při nesprávném postupu

může analytik vystavit zařízení riziku poškozením. Patří sem, podobně jako u statické analýzy, úrovně abstrakce a reverzní inženýrství. [14, 19]

4 Praktická část

V této části práce budou použity vzorky testovaného malwaru, které jsou volně dostupné a pochází z těchto zdrojů:

- https://www.virustotal.com/gui/user/malware1
- https://github.com/ytisf/theZoo/tree/master/malware/Binaries/PotaoExpress
- https://github.com/pan-unit42/Wireshark-quizzes/
- https://github.com/HuskyHacks/PMAT-labs/blob/main/labs/2-3.Challenge-SikoMode/unknown.exe.7z

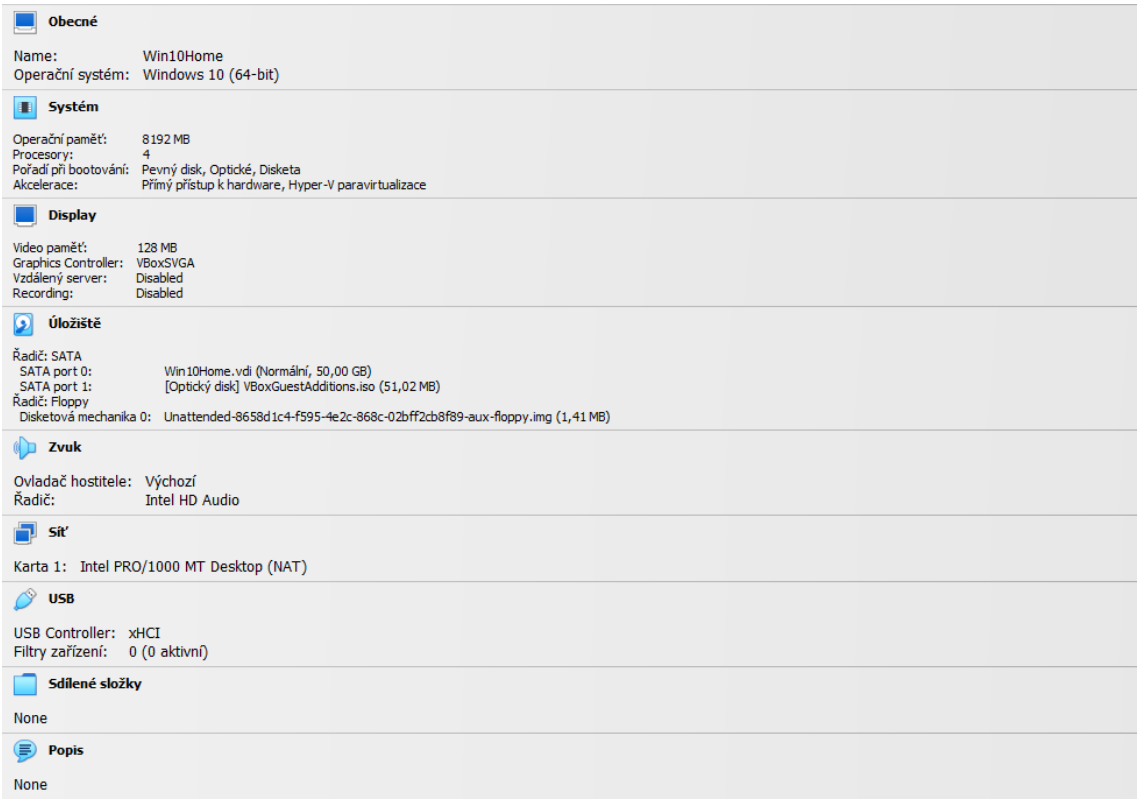
4.1 Příprava virtuálního prostředí

V této části práce se zaměříme na přípravu virtuálního prostředí jako prostředku pro analýzu a testování malware. Jelikož malware je škodlivý soubor či kód, bylo nezbytné dbát na ochranu souborů a dat v počítači. Analýza byla provedena ve virtuálním prostředí s cílem identifikovat jeho charakteristiky a možné hrozby. V našem případě konkrétně pomocí virtualizačního nástroje Oracle VM VirtualBox, který je k dispozici zcela zdarma na oficiálních webových stránkách pro operační systémy Linux, macOS a Windows.

Obrázek 3: Stažení VirtualBoxu z webových stránek. Zdroj: vlastní

Po nainstalování je nutné nastavit parametry virtuálního počítače. V našem případě tyto:

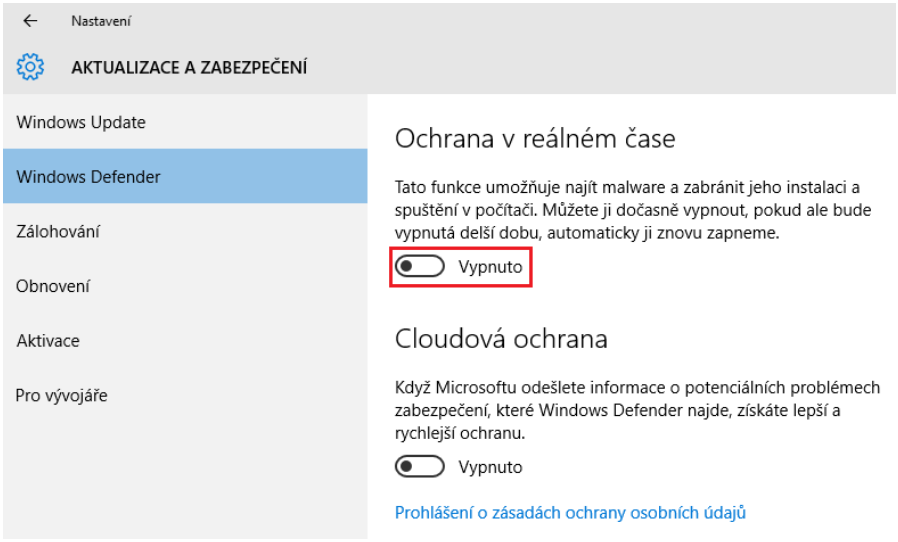
- Operační systém: Windows 10 (64-bit)
- Paměť RAM: 8192 MB
- Paměť ROM: 50 GB
- Počet procesorů: 4



Obrázek 4: Parametry virtuálního počítače. Zdroj: vlastní

Nyní je vše připraveno a virtuální počítač můžeme spustit zelenou šipkou směřující doprava s nápisem „Spustit“.

Dále je potřeba zakázat antivirový program a ochranu, aby nám Windows neblokoval stažený malware a neházel ho do karantény. To uděláme v Nastavení → Aktualizace a zabezpečení → Windows Defender → Ochrana v reálném čase.



Obrázek 5: Vypnutí skenování souborů. Zdroj: vlastní

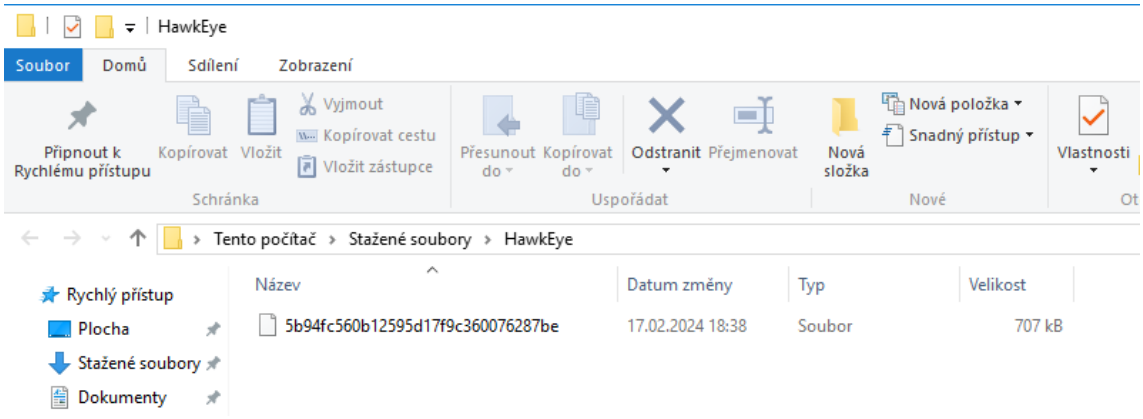
4.2 Statická analýza

4.2.1 HashMyFiles a VirusTotal

- Název testovaného malware: *5b94fc560b12595d17f9c60076287be.exe*
- Typ testovaného malware: *Trojan, Keylogger*

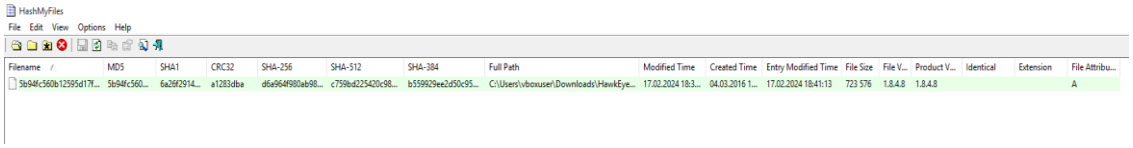
Nástroj HashMyFiles slouží k vygenerování MD5 a SHA1 hashů souborů, což jsou data přeměněná na kód složený z číslic a písmen. Po stažení ZIP souboru a jeho následném rozbalení je vše připraveno a program lze spustit.

Komprimovanou složku HawkEye.zip si rozbalíme. Hned po rozbalení si lze všimnout, že název souboru se jeví jako podezřelý.



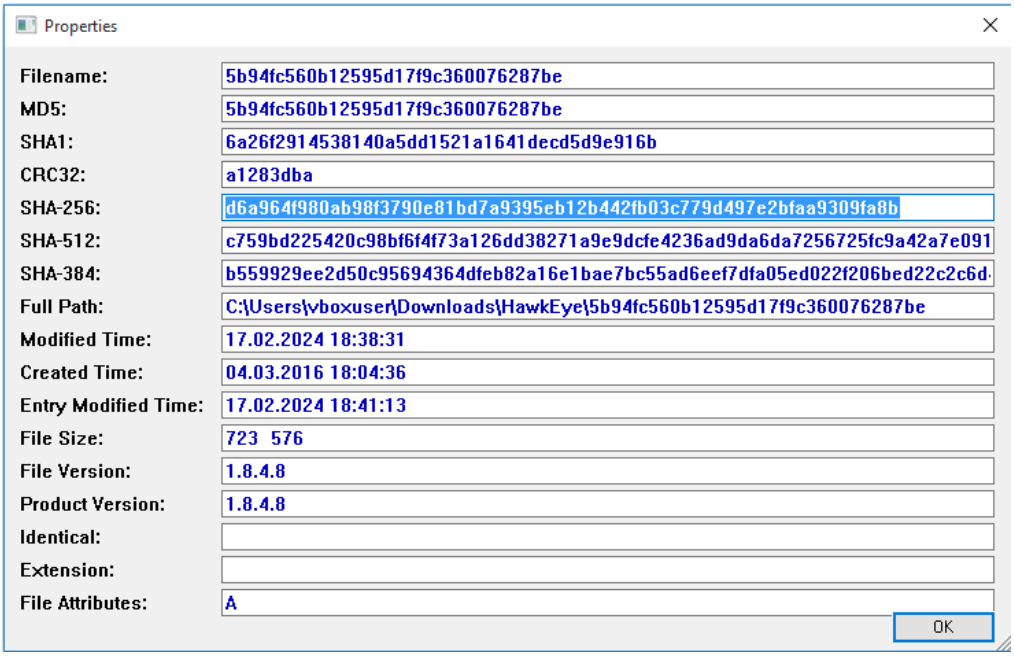
Obrázek 6: Podezřelý název souboru. Zdroj: vlastní

Spustíme tedy program HashMyFiles a přidáme soubor pomocí záložky File → Add Files nebo klávesy F2. Zobrazí se nám řádek, kde se nachází námi přidaný soubor a ve sloupcích se nachází dané informace o souboru.



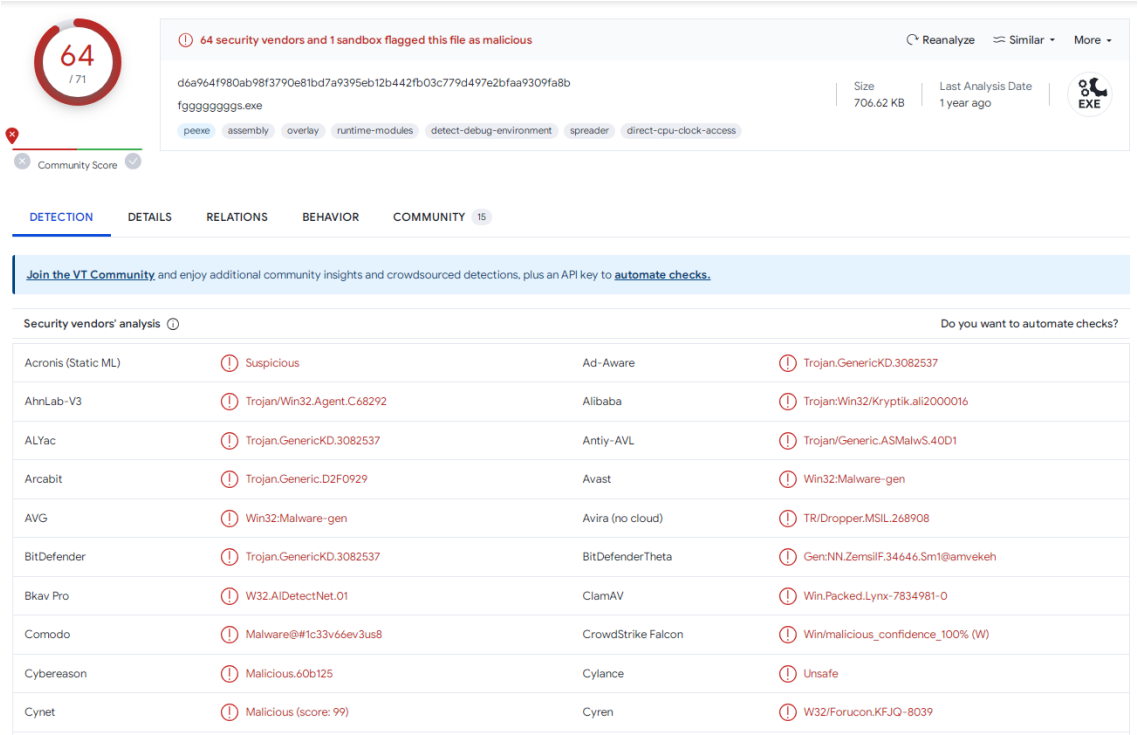
Obrázek 7: Výpis hashů a informací souboru. Zdroj: vlastní

Mnohem lepší přehled získáme dvojitým kliknutím na řádek se souborem a zobrazí se nám informace v tabulce.



Obrázek 8: Výpis hashů a informací souboru v tabulce. Zdroj: vlastní

Nyní si zkopírujeme např. SHA-256 hash a vložíme ho do webové stránky virustotal.com, kde se provede online analýza antivirovým nástrojem. V našem případě bylo skóre 64/71, což znamená, že 64 antivirových programů označilo soubor za škodlivý.

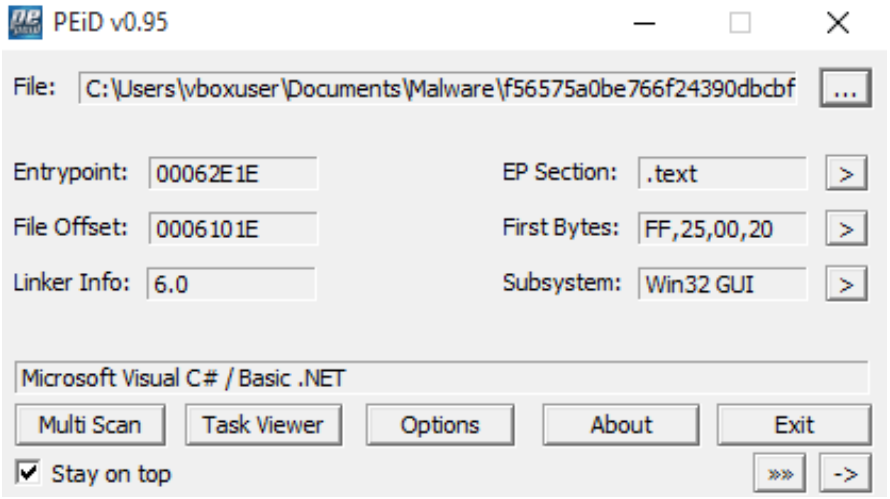


Obrázek 9: Výsledek analýzy VirusTotal. Zdroj: vlastní

4.2.2 PEiD a UPX

- Název testovaného malware: *f56575a0be766f24390dbcbf3b655b27.exe*
- Typ testovaného malware: *Trojan, Dropper, Miner*

Další dva vzorky malware budeme analyzovat v nástroji PEiD, což je program používaný v oblasti reverzního inženýrství. Byl vytvořen pro identifikování packerů a kompilátorů, které byly použity při překrytí spustitelných (PE) souborů.
Po stažení bezplatně dostupného programu ho můžeme otevřít a začít používat. Pomocí třech teček vpravo nahoře vybereme soubor, který chceme otevřít, v našem případě *f56575a0be766f24390dbcbf3b655b27.exe.*



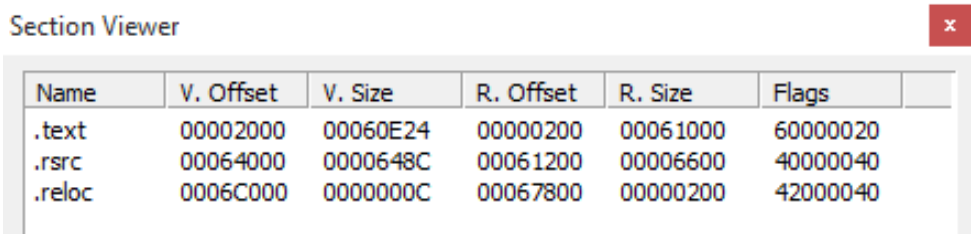
Obrázek 10: První testovaný malware v PEiD. Zdroj: vlastní

Níže si objasníme, co jednotlivé řádky znamenají:

- **Entrypoint**: paměťová adresa, kde se spustí program, po jeho načtení
- **File Offset**: posun v souborovém systému, kde se nachází entrypoint
- **Linker Info**: verze linkeru, který se používal k sestavení programu
- **EP section**: sekce v paměti, kde se nachází entrypoint
- **First Bytes**: první čtyři bajty souboru
- **Subsystem**: typ subsystému, pro který je program určen[4]

[4] Určité pasáže byly vygenerovány pomocí umělé inteligence.

V našem případě tedy vidíme, že entrypoint se nachází v sekci .text, a právě zde se nachází strojový kód. Standartní EP sekce lze detailněji prohlédnout kliknutím na symbol „>“ vpravo vedle EP Section.

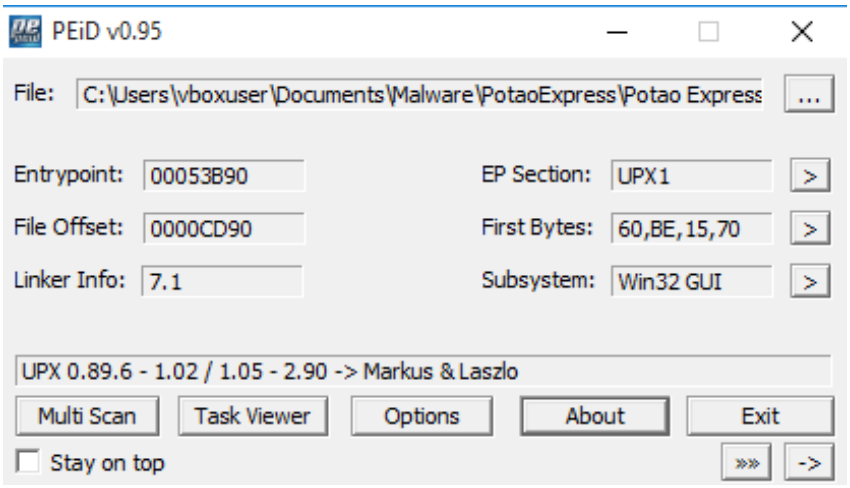


Obrázek 11: Zobrazení EP sekcí v PEiD. Zdroj: vlastní

Jedná se o program s graficky uživatelským rozhraním, určený pro operační systém Windows.

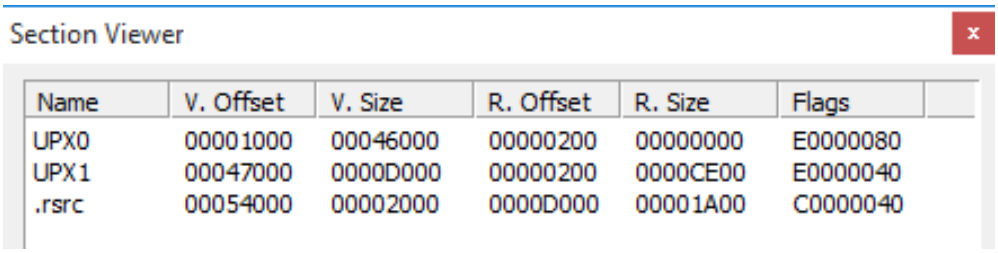
- Název testovaného malware: *Potao_1st_Version_0C…D29.exe*
- Typ testovaného malware: *Trojan*

Na druhém vzorku si můžeme všimnout, že binární soubor byl zabalen pomocí UPX, což je nástroj používaný při obfuskaci.



Obrázek 12: Druhý testovaný malware v PEiD. Zdroj: vlastní

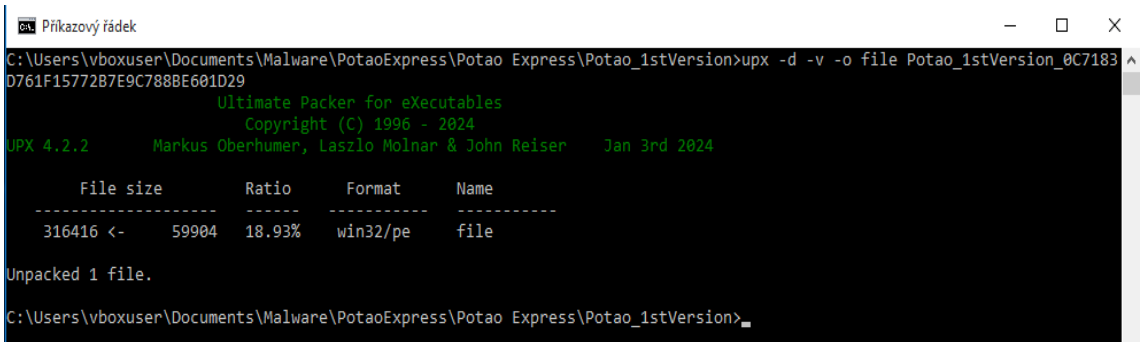
Když si opět rozklikneme EP sekce, zjistíme, že chybí standartní sekce souboru, jako jsou text, data nebo rdata. To naznačuje, že binární soubor je buď zašifrovaný, nebo jinak obfuskovaný.



Name	V. Offset	V. Size	R. Offset	R. Size	Flags
UPX0	00001000	00046000	00000200	00000000	E0000080
UPX1	00047000	0000D000	00000200	0000CE00	E0000040
.rsrc	00054000	00002000	0000D000	00001A00	C0000040

Obrázek 13: Zobrazení EP sekcí v PEiD. Zdroj: vlastní

Měli bychom tedy dále rozbalit nebo dešifrovat data, která jsou uložena na různých místech souboru. K tomu lze využít program UPX, který provede operaci zpětného zabalení. UPX je nástroj příkazového řádku, je tedy nutné ho spustit právě přes něj. Přepneme se do adresáře, ve kterém se nachází soubor, který chceme zpětně zabalit. A spustíme příkazem (viz obr. 14).



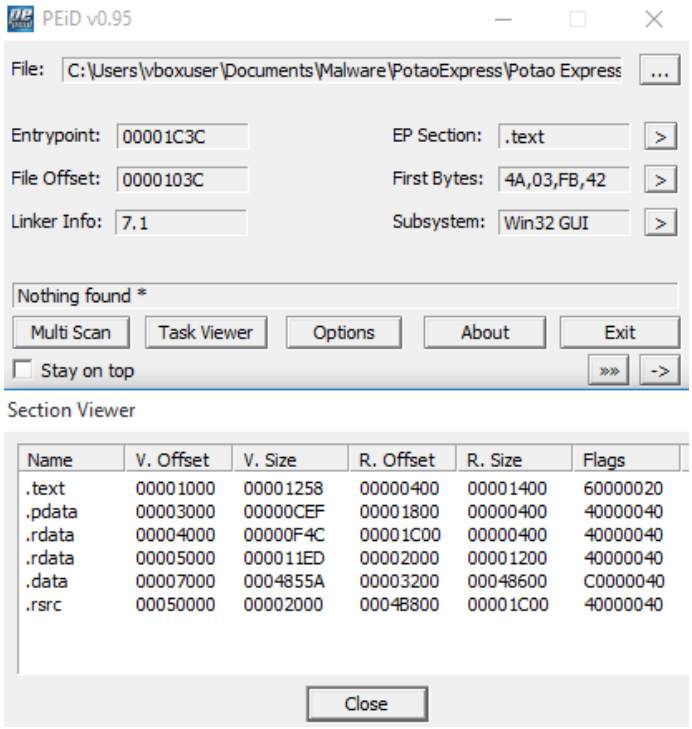
Obrázek 14: Zpětné zabalení souboru pomocí UPX. Zdroj: vlastní

Níže si objasníme syntaxi příkazu:

- **upx**: spouštíme pomocí UPX
- **-d**: přepínač, který říká, že chceme provést dekompresi souboru
- **-v**: přepínač, sloužící k podrobnějšímu výstupu o své činnosti
- **-o** file: přepínač, určující název výstupního souboru po dekompresi. V našem případě „file“
- **název souboru**

Po provedení tohoto příkazu se zajistilo, že původně komprimovaný soubor byl dekomprimován a výsledkem je nový soubor, který obsahuje dekomprimovaný kód, včetně zpětně obnovených sekcí.

V programu PEiD otevřeme zpětně zabalený soubor a vidíme, že i přes nenalezení konkrétního kompilátoru je nyní program schopný rozpoznat různé očekáváné části binárního souboru.



Obrázek 15: Soubor po dekompresi v PEiD. Zdroj: vlastní

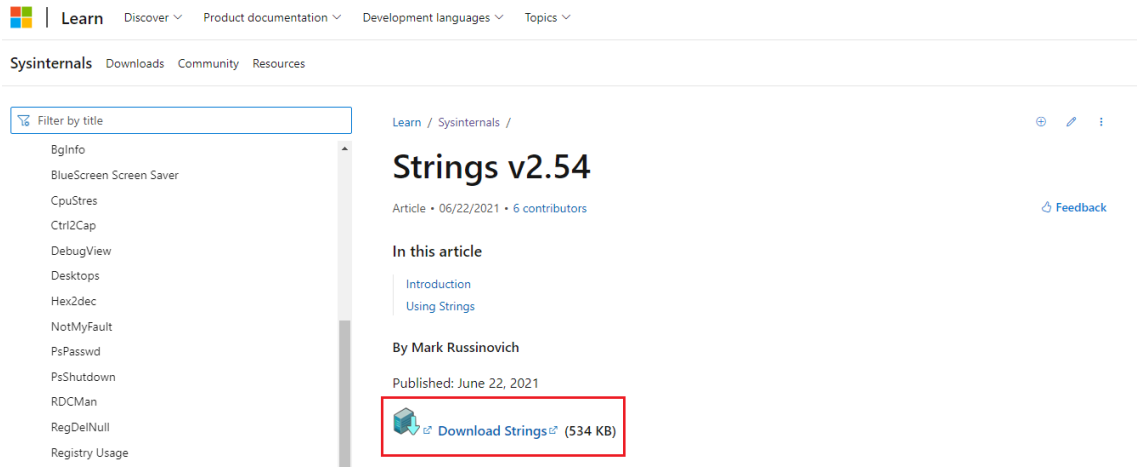
Výsledkem je, že dekomprese pomocí nástroje UPX úspěšně obnovila původní strukturu souboru, včetně standartních sekcí. Tento krok může být užitečný při další analýze malware, protože standartní sekce mohou obsahovat důležité kódy a data, které jsou součástí chování a funkcionality malware.

4.2.3 Hledání řetězců

- Název testovaného malware: *malware_sample*
- Typ testovaného malware: *Trojan, Dropper, Miner*

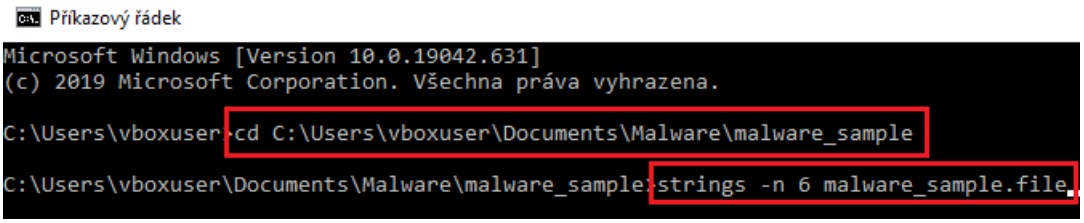
V této kapitole práce se budeme zabývat hledáním řetězců v souborech. Tato technika nám může odhalit spoustu užitečných identifikátorů v pozdější dynamické analýze, jako např. funkcionalitu malware, detekci CnC serverů nebo najití zašifrovaných dat.

Nejprve si stáhneme program String z oficiálních Microsoft webových stránek, v našem případě verzi 2.54.



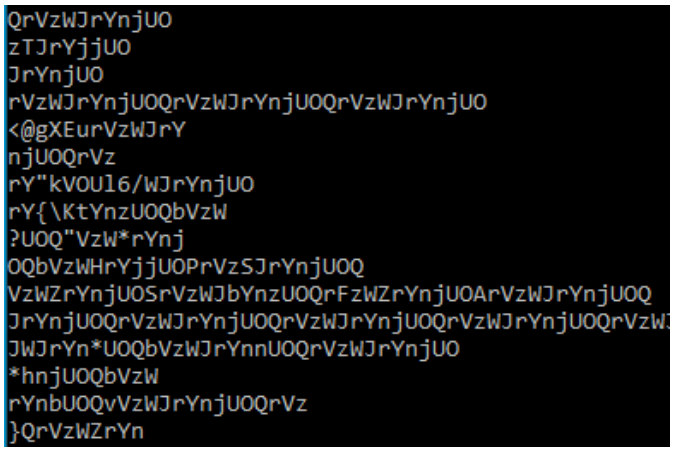
Obrázek 16: Stažení programu Strings. Zdroj: vlastní

Samotný program je dobré extrahovat do složky *C:\Windows\System32,* která je nastavená jako „PATH“ proměnná prostředí a není třeba nastavovat novou cestu. Spustíme si příkazový řádek, přepneme se do složky, kde se nachází náš vzorek a spustíme program strings. Přepínač *„-n 6“* znamená, že chceme zobrazit pouze řetězce delší jak 6 znaků.



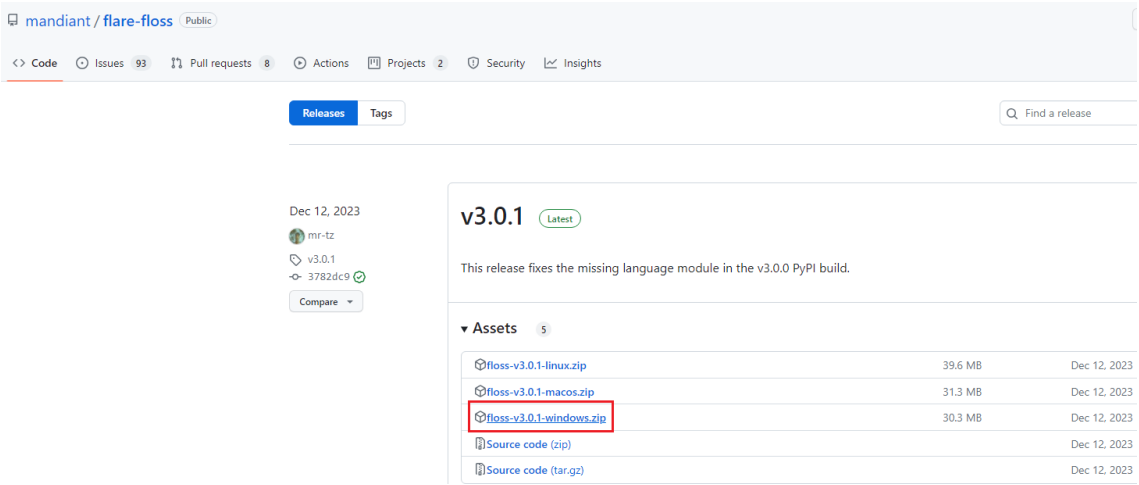
Obrázek 17: Spuštění programu Strings. Zdroj: vlastní

Po delším zkoumání zjistíme, že se jedná pouze o náhodné znaky a nelze z toho nic zjistit. Jedná se tedy o obfuskovaný malware.



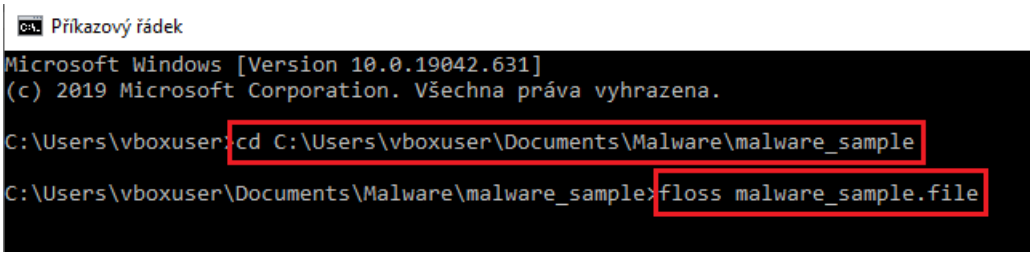
Obrázek 18: Zašifrované znaky. Zdroj: vlastní

Existuje nástroj, který dokáže zpětně deobfuskovat kód a extrahovat důležité řetězce, jmenuje se Flare Obfuscated String Solver. Je k dispozici pro operační systémy Linux, MacOS a Windows na webové stránce github.com.



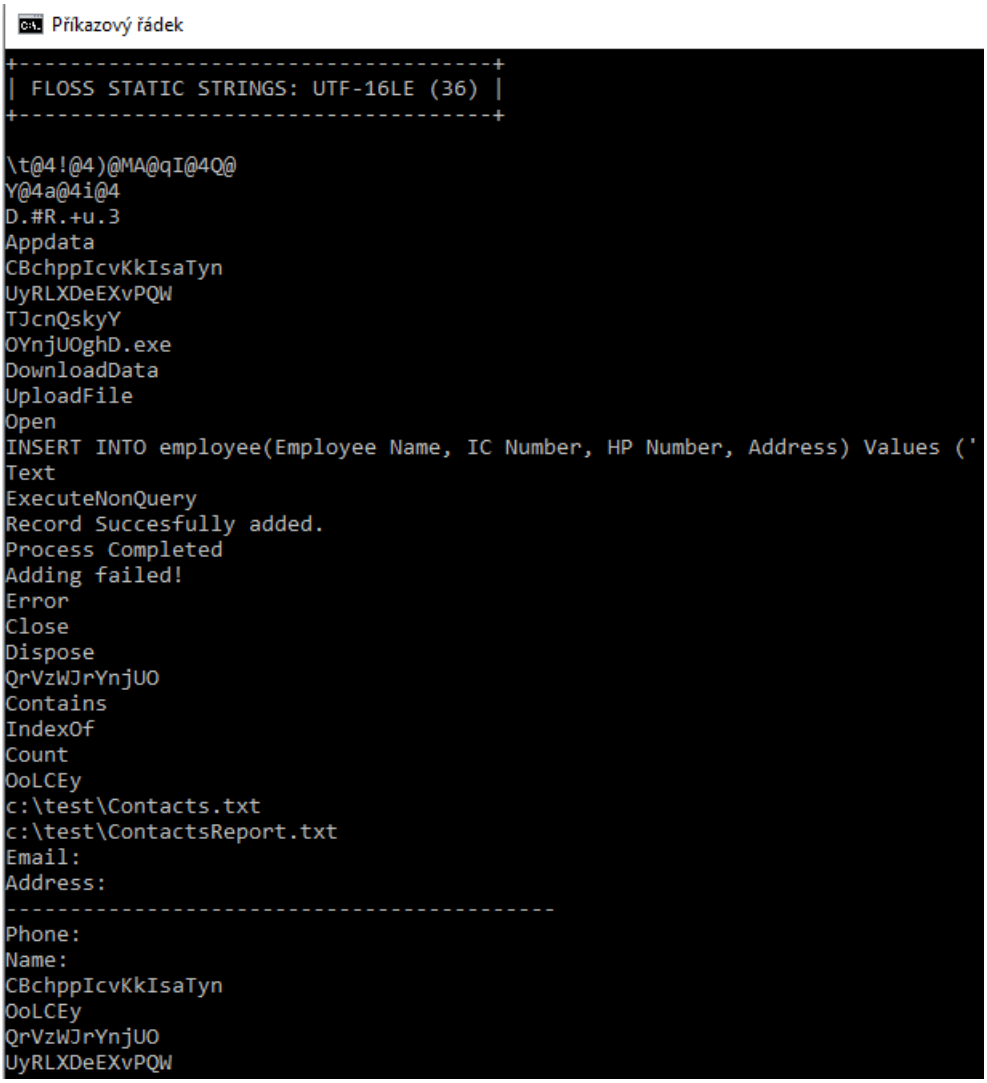
Obrázek 19: Stažení programu Flare Obfuscated String Solver. Zdroj: vlastní

Program opět extrahujeme do složky *C:\Windows\System32*, jako u programu Strings. Znovu spustíme příkazový řádek, přepneme do složky, kde se nachází testovaný vzorek a spustíme program Flare.



Obrázek 20: Spuštění programu Flare Obfuscated String Solver. Zdroj: vlastní

Čas deobfuskace záleží na velikosti vloženého souboru. Po chvíli můžeme vidět, že program úspěšně deobfuskoval nějaké řetězce.

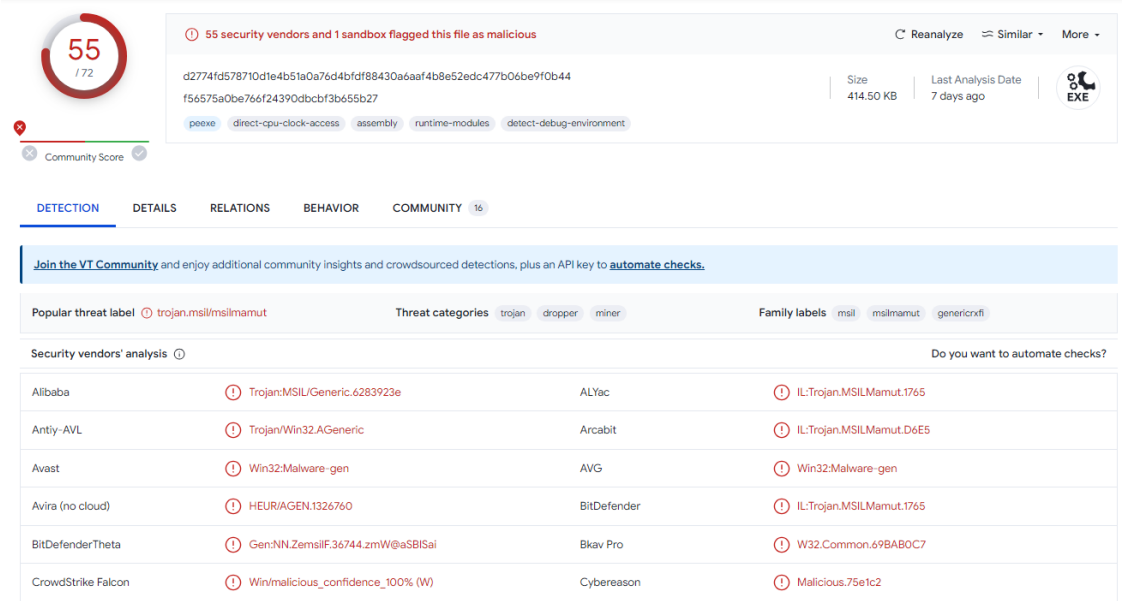


Obrázek 21: Deobfuskované řetězce. Zdroj: vlastní

Nyní si rozebereme možný význam jednotlivých řetězců:

- **Appdata** – označuje cestu k adresáři na systému Windows, malware nejspíše manipuluje sobory a nastavení uloženými v tomto adresáři
- **OYnjUOghD.exe** – spustitelný soubor, který může být součástí malwaru
- **DownloadData, UploadFile, Open** – malware provádí stahování dat, nahrávání souborů a otevíraní souborů
- **INSERT INTO employee** – SQL příkaz pro vkládání dat zaměstnanců do databáze
- **Txt, Process Completed, …** – různé zprávy, stavové hlášky nebo akce v kódu malwaru
- **Contains, IndexOf, Count** – názvy funkcí nebo metod používaných v kódu malwaru
- **c:\test\Contacts.txt, c:\test\ContactsReport.txt** – cesty k souborům, kam se nejspíše ukládají informace
- **Email, Address, Phone, Name** – možná data, které malware sbírá[5]

Jakmile vložíme SHA-256 hash souboru do VirusTotal, zjistíme, že se opravdu jednalo o škodlivý program.



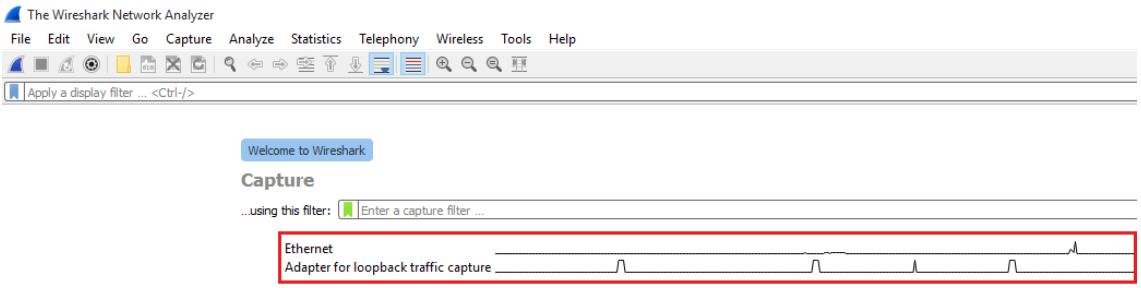
Obrázek 22: Report z webové stránky VirusTotal. Zdroj: vlastní

[5] Určité pasáže byly vygenerovány pomocí umělé inteligence.

4.3 Dynamická analýza

4.3.1 Wireshark

Jako první nástroj, ve kterém budeme analyzovat malware pomocí dynamické analýzy je bezplatný open-source nástroj Wireshark. Používá se pro analýzu síťového provozu. Po stažení z oficiálních webových stránek a následné instalaci je nástroj připraven k použití. Pokud chceme zachytit tok sítě na našem rozhraní, je třeba ho vybrat na úvodní straně při spuštění programu.



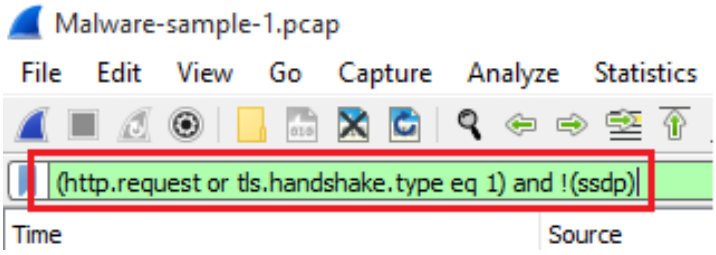
Obrázek 23: Výběr rozhraní v programu Wireshark. Zdroj: vlastní

V našem případě rozhraní vybírat nebudeme, protože máme přímo k dispozici soubor se síťovým provozem na rozhraní, kde se objevil malware.

- Název testovaného malware: *86607.dat*
- Typ testovaného malware: *Trojan, QakBot*

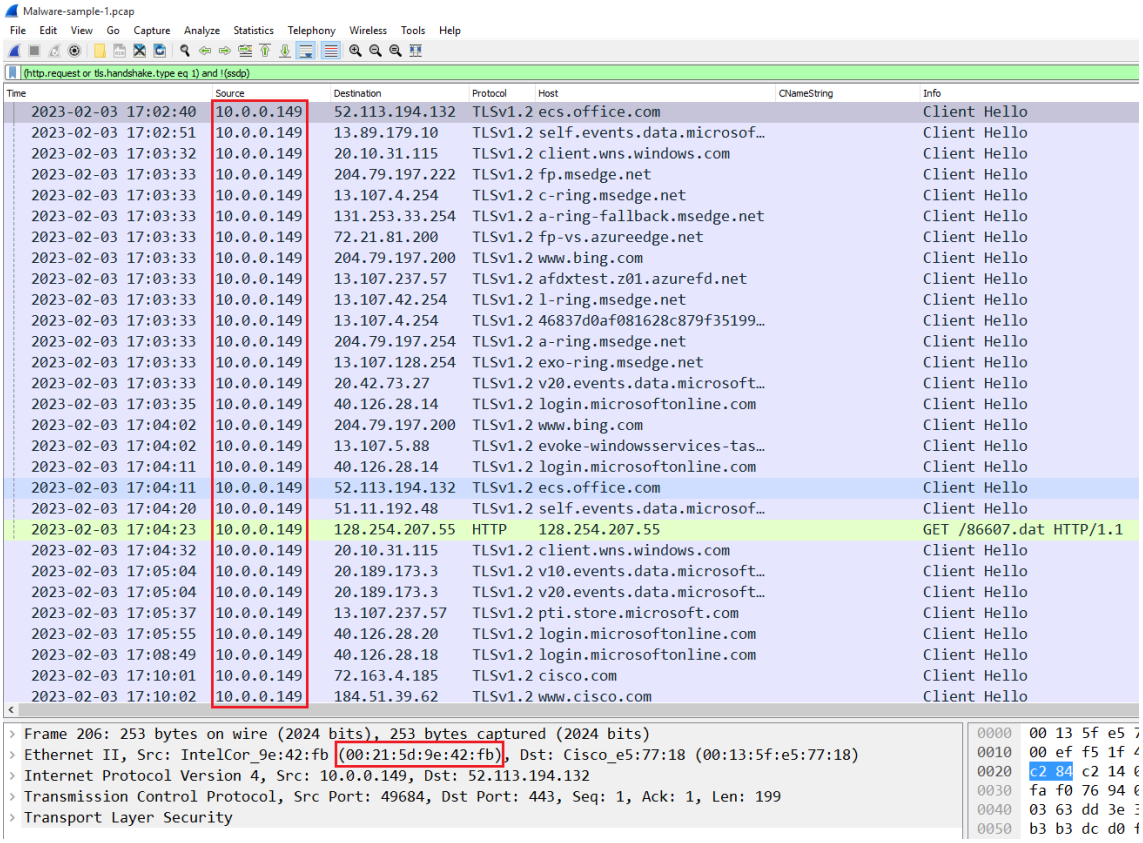
Otevřeme si náš soubor pomocí záložky vlevo nahoře File → Open nebo kombinací klávesových zkratek CTRL+O. Do filtru nahoře zadáme *(http.request or tls.handshake.type eq 1) and !(ssdp))*. To nám zajistí, že se nám zobrazí pouze HTTP požadavky a TLS handshaky s typem 1, označované jako Client Hello. To je první fáze

TLS spojení, kde klient zašle zprávu serveru pro zahájení komunikace. Dále filtr vylučuje pakety, které obsahují SSDP, tedy protokol pro objevování zařízení v síti.



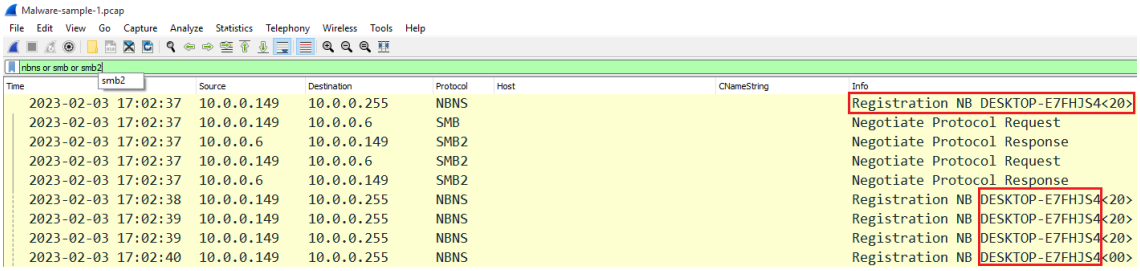
Obrázek 24: Počáteční filtr v programu Wireshark. Zdroj: vlastní

Už po vyfiltrování jsme schopni určit IP a MAC adresu zdrojového zařízení.



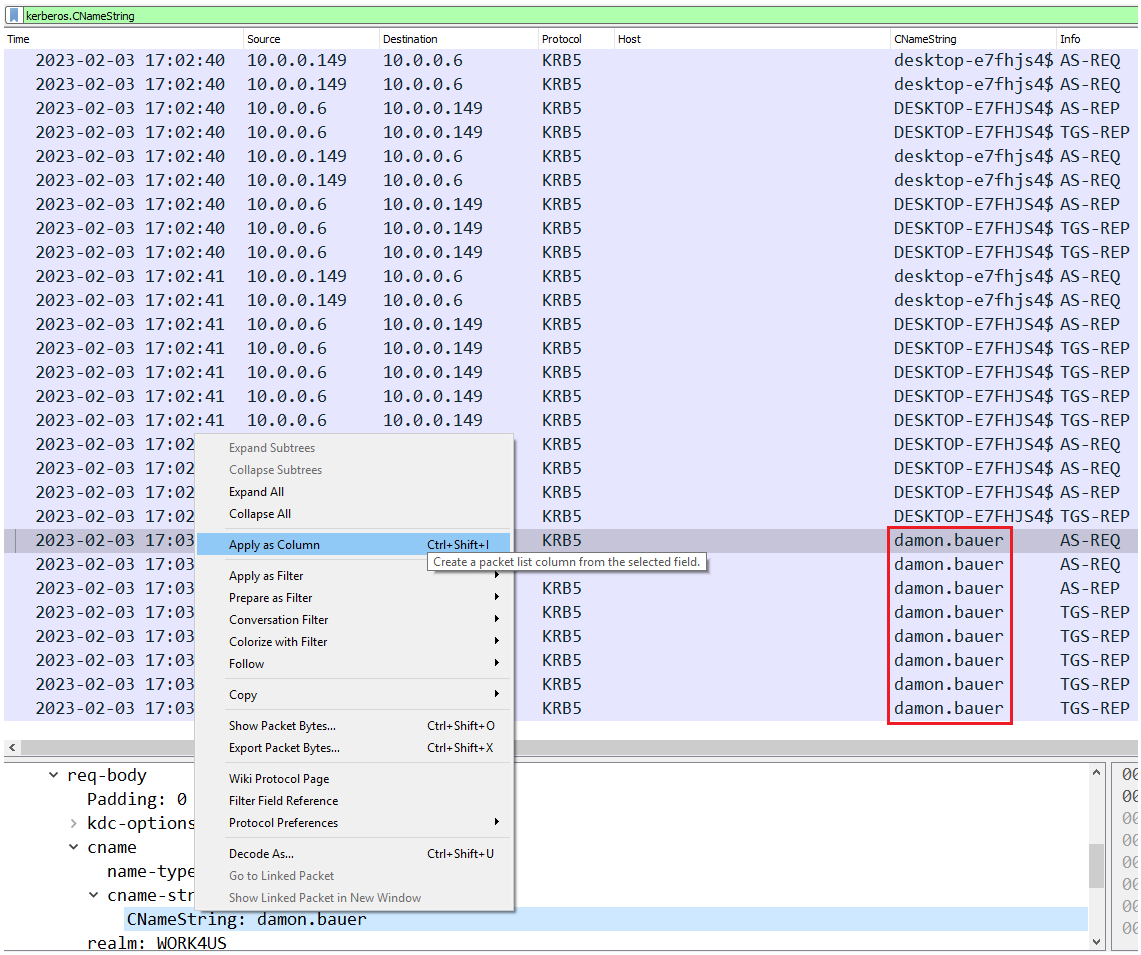
Obrázek 25: Zdrojové IP a MAC adresy v programu Wireshark. Zdroj: vlastní

Nyní se pokusíme zjistit název zdrojového počítače. Do filtru zadáme příkaz *nbns or smb or smb2,* který nám zobrazí pouze pakety s těmito protokoly. V prvních pár paketech ve sloupci jsme schopni zjistit název počítače.



Obrázek 26: Zjištění názvu PC v programu Wireshark. Zdroj: vlastní

Dále můžeme nalézt název uživatelského účtu. Použijeme opět nový filtr příkazem *kerberos.CNameString,* který hledá atribut CNameString v Kerberos protokolu, jenž se používá pro ověřování a autentizaci uživatel. Tudíž můžeme očekávat, že název uživatelského účtu by se mohl nacházet právě zde. V posledních paketech se po rozkliknutí protokolu Kerberos ve spodní části a následné aplikace atributu CNameString jako sloupec zobrazí název uživatelského účtu.



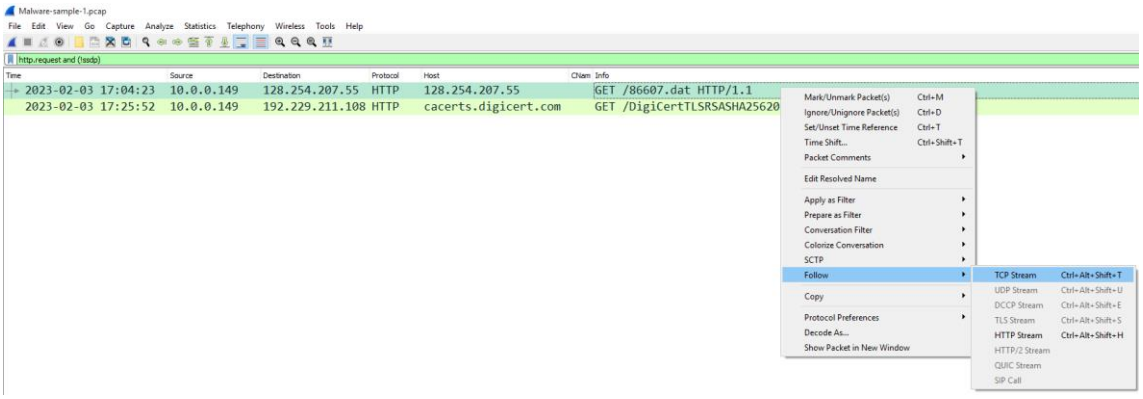
Obrázek 27: Zjištění názvu uživatelského účtu v programu Wireshark. Zdroj: vlastní

V tomto okamžiku máme k dispozici nějaké základní informace ohledně zdrojového zařízení, a to:

- Zdrojová IP adresa zařízení: *10.0.0.149*
- Zdrojová MAC adresa zařízení: *00:21:5d:9e:42:fb*
- Název zařízení: *DESKTOP-E7FHJS4*
- Název uživatelského účtu: *damon.bauer*

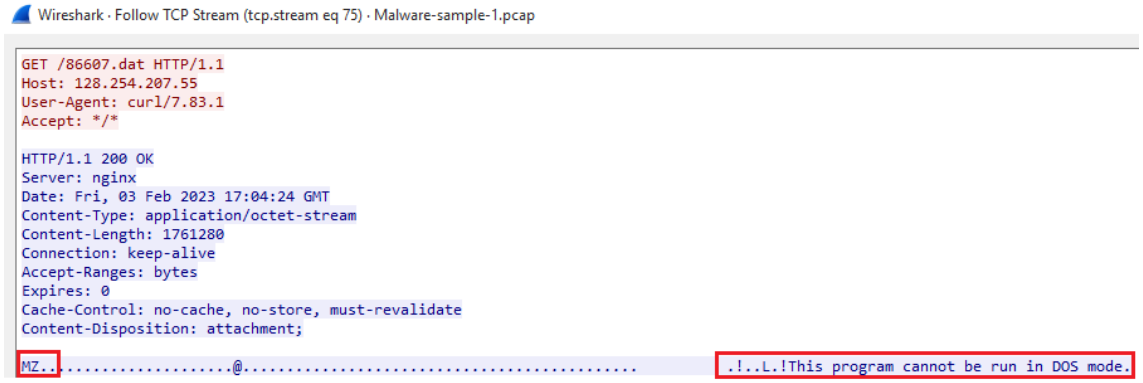
Dále budeme hledat, zda se mezi pakety nachází nějaká nežádoucí komunikace, popřípadě škodlivý soubor. Jako první si znovu vyfiltrujeme pouze HTTP požadavky.

Druhý požadavek je na běžnou adresu, zatímco první je požadavek na IP adresu pro 86607.dat. Zobrazíme si TCP proud prvního požadavku, ten zahrnuje všechny pakety přenášené mezi zdrojem a cílem této TCP komunikace.



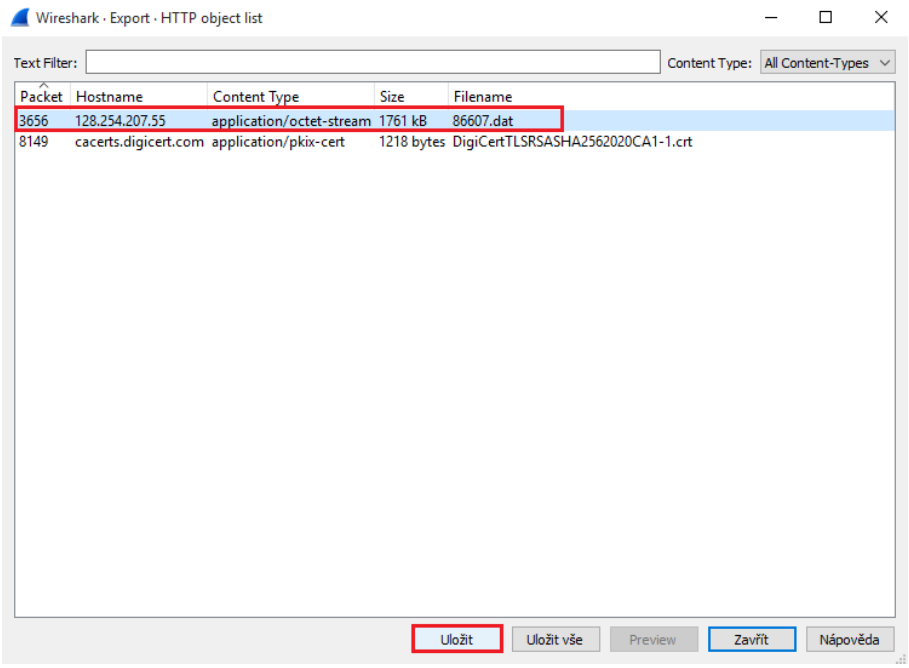
Obrázek 28: Zobrazení TCP proudu v programu Wireshark. Zdroj: vlastní

Z výpisu lze usoudit, že se jedná o část hlavičky souboru, která obsahuje znaky „MZ“, které označují starší formát exe souborů na platformě Windows. Také je možné vidět oznámení, že program nelze spustit v DOS módu.



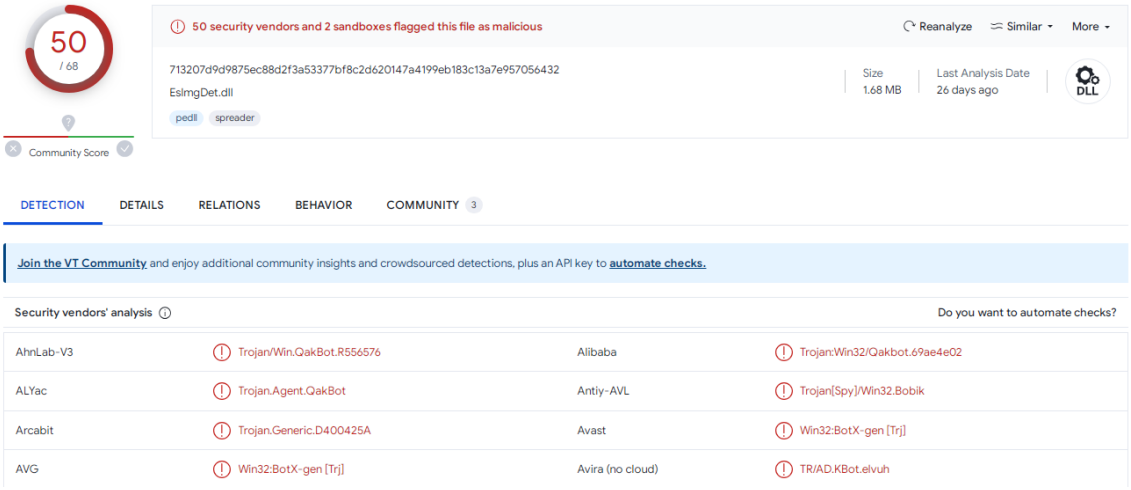
Obrázek 29: Informace TCP proudu v programu Wireshark. Zdroj: vlastní

Soubor je možné vyexportovat a stáhnout do našeho zařízení pomocí záložky File → Export Objects → HTTP → výběr souboru → Uložit.



Obrázek 30: Uložení souboru v programu Wireshark. Zdroj: vlastní

Soubor si následně vložíme do programu HashMyFiles, vygenerujeme hash, který následně vložíme do VirusTotal.



Obrázek 31: Report z webové stránky VirusTotal. Zdroj: vlastní

Na výsledném skóre lze vidět, že soubor byl opravdu škodlivý. Konkrétně se jednalo o QakBot Trojan.

V tomto programu si demonstrujeme ještě jednu analýzu na jiném vzorku.

Stejně, jako u prvního vzorku, aplikujeme počáteční filtr *http.request or tls.handshake.type eq 1) and !(ssdp).* Vidíme požadavek s cílovou adresou 80.77.25.175 a poznámkou *GET /main.php.* To může signalizovat, že malware navázal spojení s určitým webovým serverem, konkrétně tedy souborem *main.php*, který je nejspíše na serveru dosptupný.

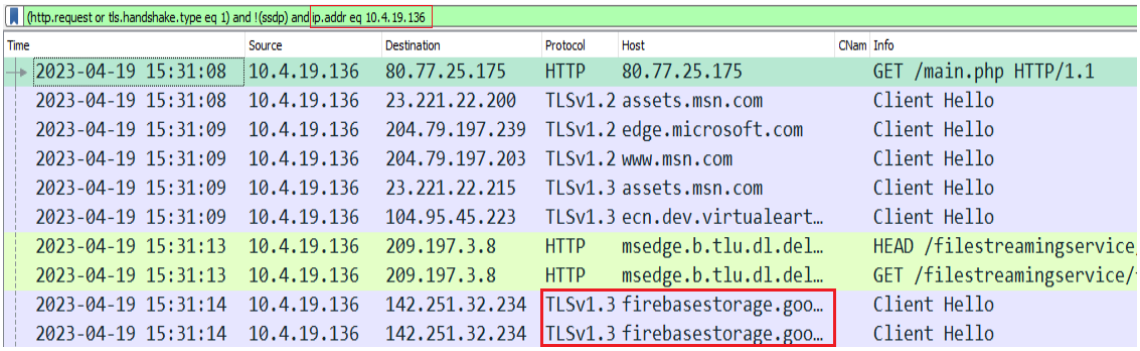
2023-04-19 15:28:40	10.4.19.136	20.242.39.171	TLSv1.2	fe3cr.delivery.mp.m…	Client Hello
2023-04-19 15:28:41	10.4.19.136	104.208.16.89	TLSv1.2	v10.events.data.mic…	Client Hello
2023-04-19 15:28:43	10.4.19.136	104.208.16.89	TLSv1.2	v10.events.data.mic…	Client Hello
2023-04-19 15:30:41	10.4.19.136	40.126.28.23	TLSv1.2	login.microsoftonli…	Client Hello
2023-04-19 15:30:43	10.4.19.136	142.251.46.131	TLSv1.3	update.googleapis.c…	Client Hello
2023-04-19 15:30:43	10.4.19.136	204.79.197.200	TLSv1.2	edgeservices.bing.c…	Client Hello
2023-04-19 15:31:07	10.4.19.136	204.79.197.203	TLSv1.2	www.msn.com	Client Hello
2023-04-19 15:31:08	10.4.19.136	80.77.25.175	HTTP	80.77.25.175	GET /main.php HTTP/1.1
2023-04-19 15:31:08	10.4.19.136	23.221.22.200	TLSv1.2	assets.msn.com	Client Hello
2023-04-19 15:31:09	10.4.19.136	204.79.197.239	TLSv1.2	edge.microsoft.com	Client Hello
2023-04-19 15:31:09	10.4.19.136	204.79.197.203	TLSv1.2	www.msn.com	Client Hello
2023-04-19 15:31:09	10.4.19.136	23.221.22.215	TLSv1.3	assets.msn.com	Client Hello
2023-04-19 15:31:09	10.4.19.136	104.95.45.223	TLSv1.3	ecn.dev.virtualeart…	Client Hello

Obrázek 32: Podezřelý HTTP GET požadavek. Zdroj: vlastní

Opět, stejně jako u prvního vzorku, prozkoumáme TCP proud a vidíme, že hlavička odpovědi HTTP zobrazuje kód 302, který přesměrovává provoz na podezřelou URL adresu.

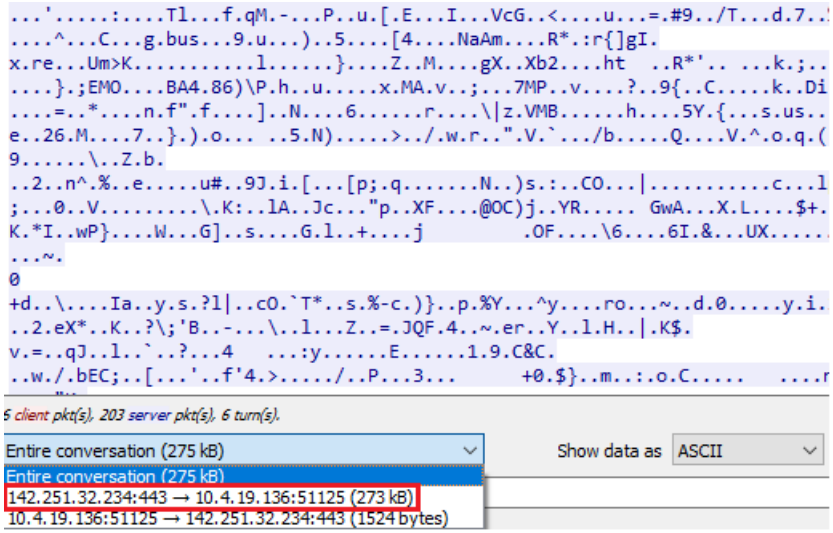
Obrázek 33: Další informace TCP proudu. Zdroj: vlastní

TCP proud zavřeme a do filtru přidáme příkaz IP adresy klienta, to odhalí provoz HTTPS na podezřelé adrese krátce po provozu na původní URL adrese se souborem *main.php*.



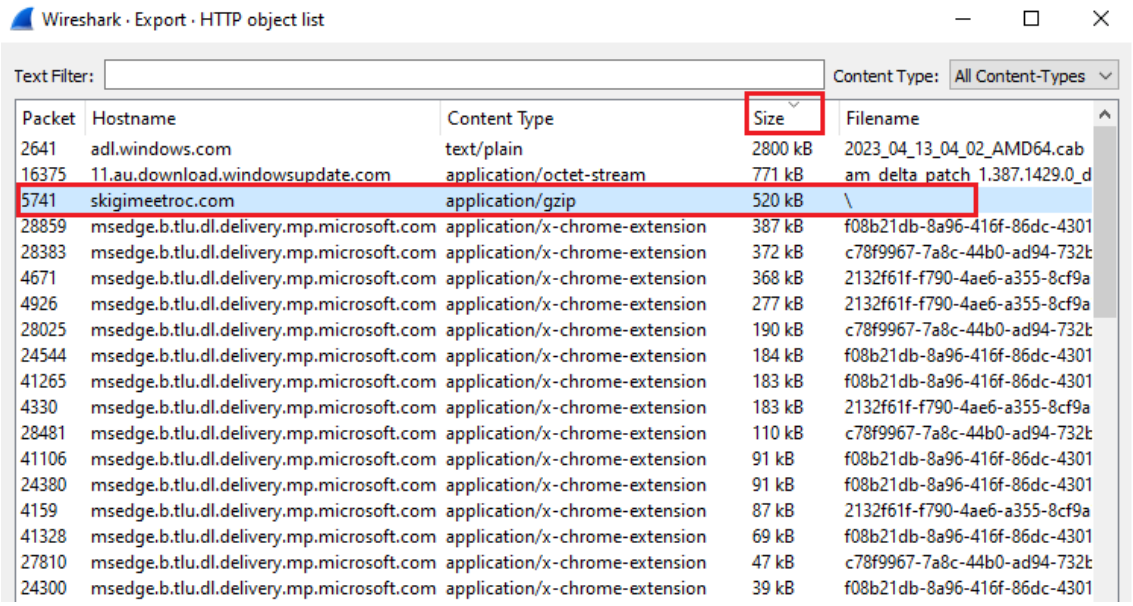
Obrázek 34: Vyfiltrování paketů podle IP adresy klienta. Zdroj: vlastní

Znovu prozkoumáme TCP proud této komunikace, který nám odhaluje 273 KB odeslaných dat ze serveru do hostitele. To s největší pravděpodobností značí, že do hostitele mohl být odeslán nějaký soubor.



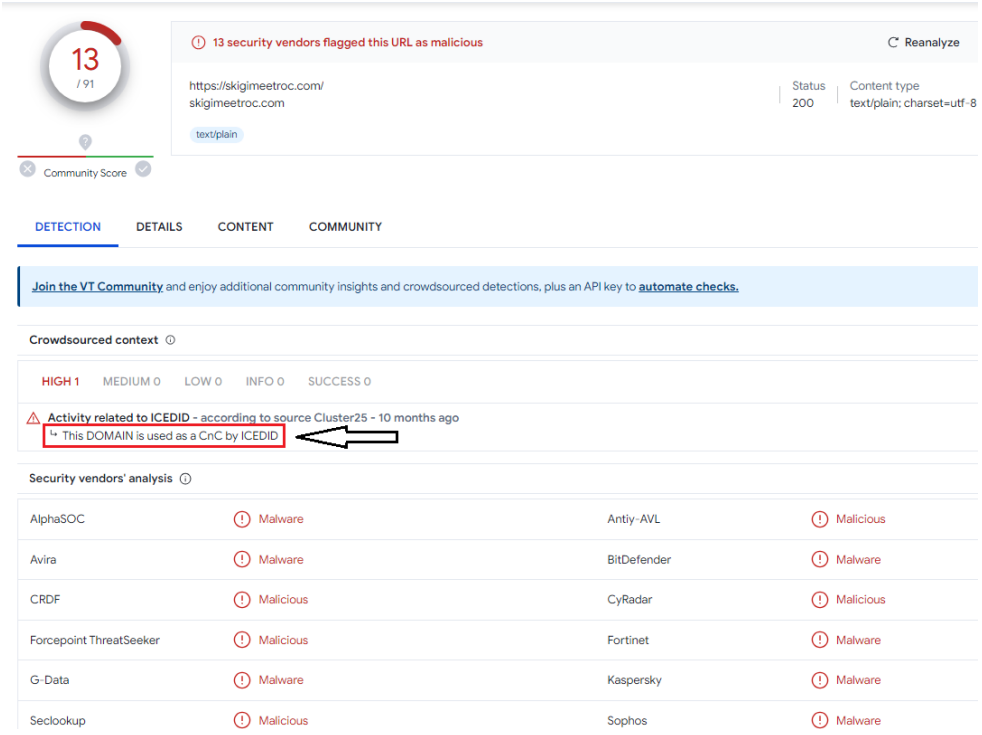
Obrázek 35: Zobrazení dat odeslaných ze serveru. Zdroj: vlastní

Stejně jako u prvního vzorku se podíváme na stažené soubory pomocí File → Export Objects → HTTP. Soubory seřadíme podle velikosti od největšího, pomocí dvojitého kliknutí na Nadpis sloupce „Size“. Jako podezřelý se jeví objekt na paketu 5741 díky své URL adrese. Všechny ostatní jsou totiž z Microsoft URL adres.



Obrázek 36: Podezřelá URL adresa v exportu. Zdroj: vlastní

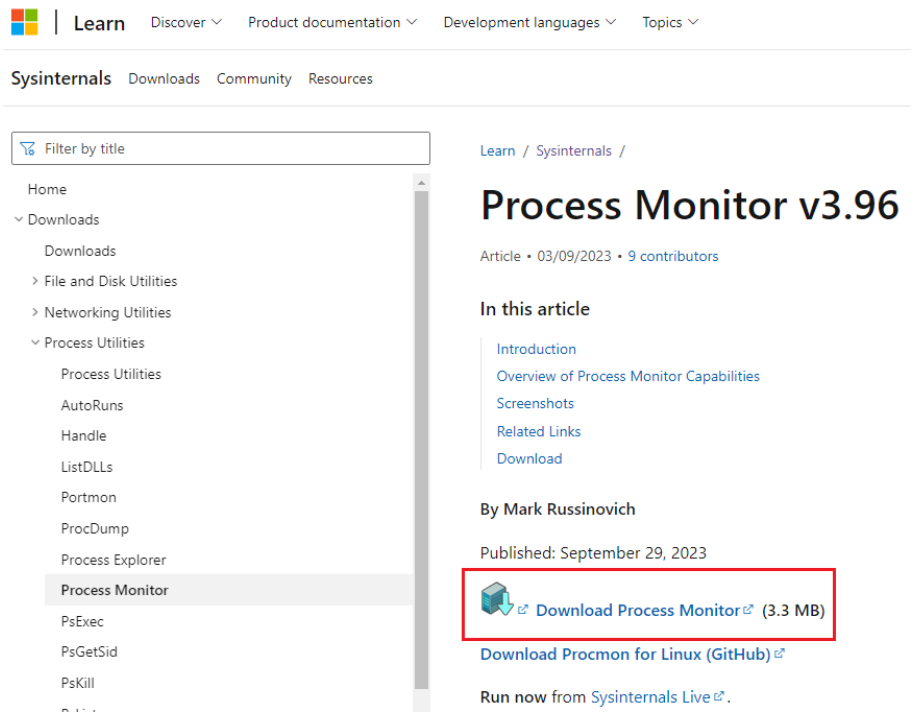
Po následném vložení této URL adresy do webové stránky VirusTotal je patrné, že tato webová stránka je opravdu infikovaná. Dokonce je zde poznámka, že se jedná o řídící a kontrolní server (CnC) pro malware ICEID, což je bankovní trojan využívaný pro vzdálený přístup.



Obrázek 37: Report podezřelé URL adresy ve VirusTotal. Zdroj: vlastní

4.3.2 Process Monitor

Druhým nástrojem dynamické analýzy malwaru v této práci je Process Monitor, jinak také nazývaný Procmon. Slouží k poskytnutí detailních informací o aktivitách procesů, které v dynamické analýze hrají důležitou roli. Program opět stáhneme z oficiálních stránek Microsoft.

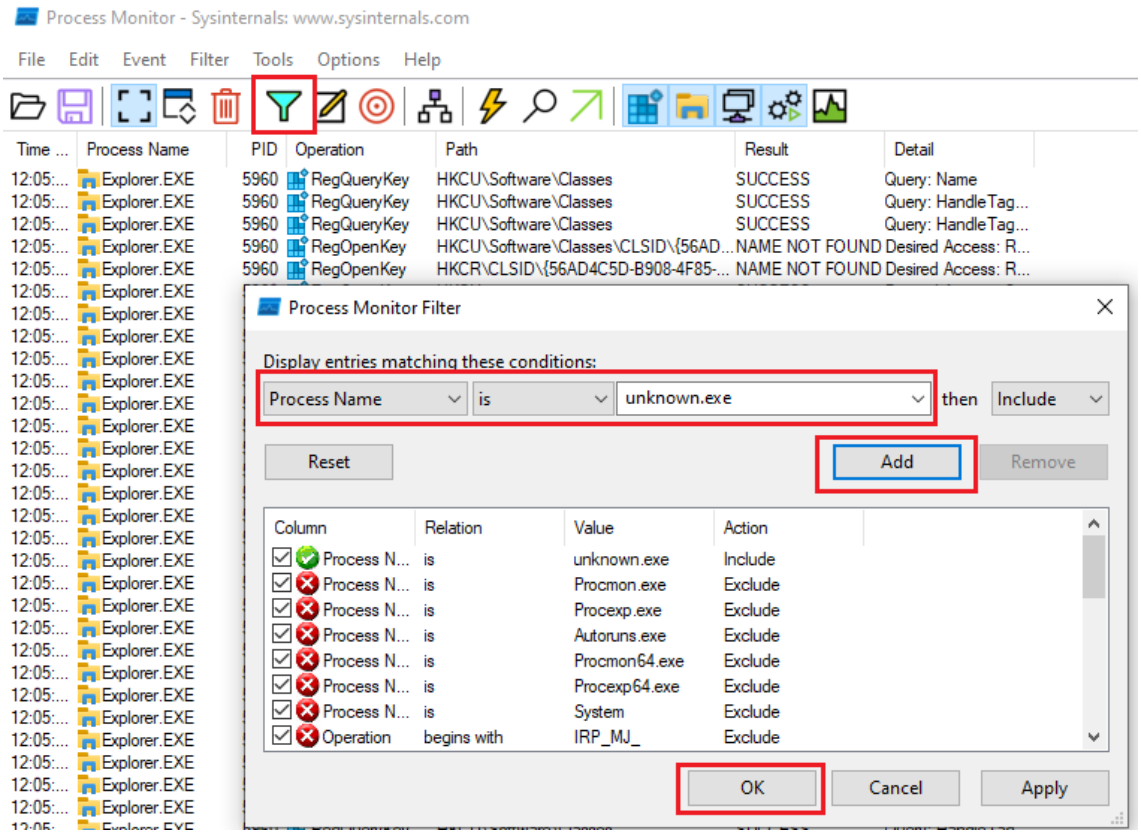


Obrázek 38: Stažení programu Process Monitor. Zdroj: vlastní

Po stažení extrahujeme a otevřeme soubor *Procmon.exe*.

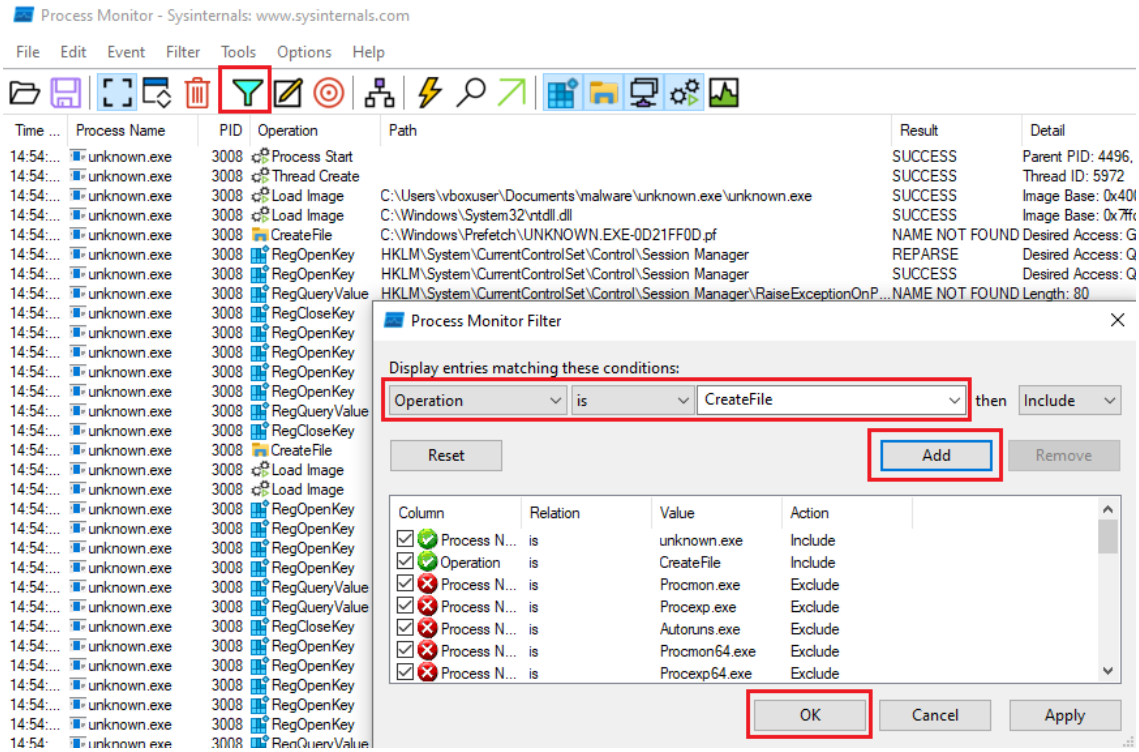
- Název testovaného malware: *unknown.exe*
- Typ testovaného malware: *Trojan, Spyware*

Otevřeme si náš malware s názvem *unknown.exe.* Po spuštění se nic viditelného nestalo, podíváme se tedy do programu Process monitor. Jediná nám známá informace o tomto malwaru je jeho název. Otevřeme tedy Filtr a zde zadáme, že chceme zobrazit pouze procesy, které se jmenují uknown.exe.



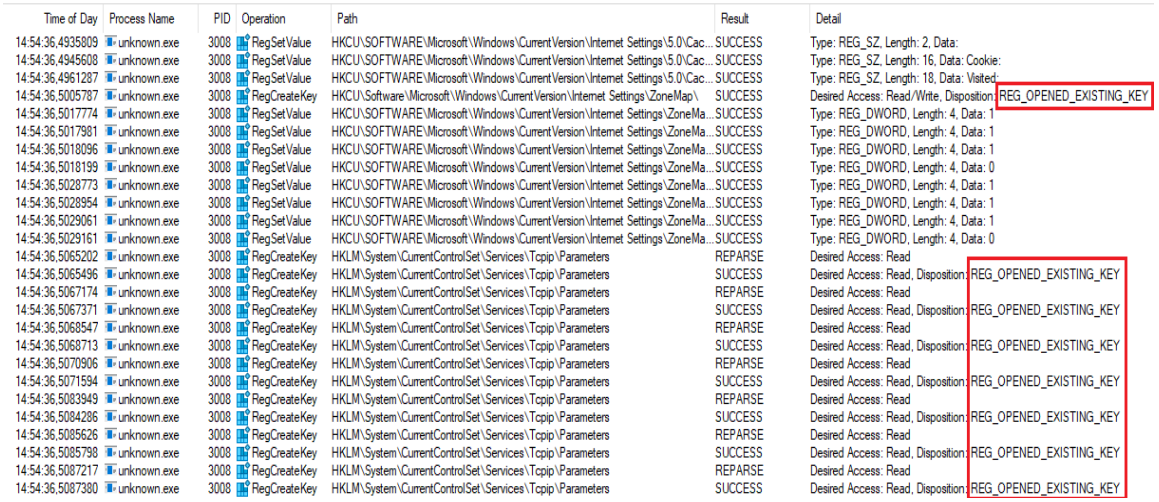
Obrázek 39: Vyfiltrování procesů podle názvu. Zdroj: vlastní

Avšak program vyfiltroval přes 2000 procesů souvisejících s tímto názvem. Nachází se zde aktivita registrů, souborového systému, síťová aktivita a aktivita procesů a vláken. Procházet je všechny po jednom by zabralo mnoho času, a proto budeme ještě více detailněji filtrovat. Pomocí nového filtru se podíváme, zda je tu nějaká operace vytváření nových souborů.



Obrázek 40: Vyfiltrování procesů podle operace. Zdroj: vlastní

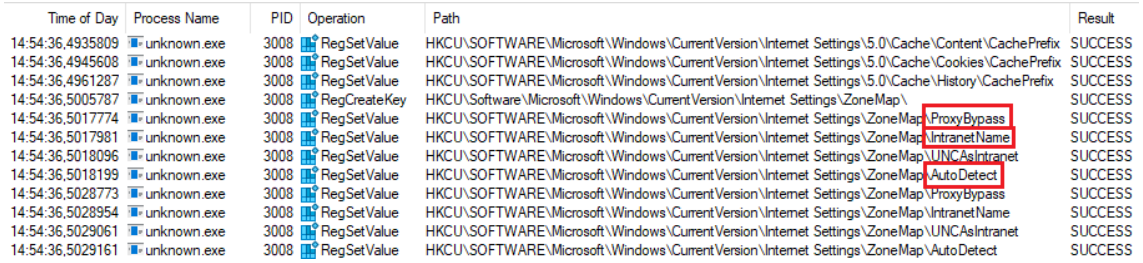
Vidíme, že malware vytvořil spoustu souborů s příponou .dll do složky C:\Windows\Windows32. To může znamenat, že tyto soubory se mohou jevit jako legitimní knihovny, které se následně použijí k zapojení do spouštění aplikací nebo procesů. To může umožnit, že se malware bude spouštět automaticky při každém spouštění systému nebo jiných aplikací. Dále tyto DLL soubory mohou být spojeny se změnami v registrech systému, díky tomu pak malware dokáže přežívat restarty systému a udržovat svou aktivitu skrytou. Opět vyfiltrujeme operace pro vytváření klíčů a nastavení hodnoty v registru. Je patrné, že tento malware opravdu zasahuje i do registrů operačního systému. Dokonce je natolik „chytrý“, že nevytváří nové klíče, ale přepisuje hodnoty již existujících klíčů. To lze vidět ve sloupci Detail.



Time of Day	Process Name	PID	Operation	Path	Result	Detail
14:54:36,4935809	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cac...	SUCCESS	Type: REG_SZ, Length: 2, Data:
14:54:36,4945608	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cac...	SUCCESS	Type: REG_SZ, Length: 16, Data: Cookie:
14:54:36,4961287	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cac...	SUCCESS	Type: REG_SZ, Length: 18, Data: Visited:
14:54:36,5005787	unknown.exe	3008	RegCreateKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\	SUCCESS	Desired Access: Read/Write, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5017774	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
14:54:36,5017981	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
14:54:36,5018096	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
14:54:36,5018199	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14:54:36,5028773	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
14:54:36,5028954	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
14:54:36,5029061	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
14:54:36,5029161	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMa...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14:54:36,5065202	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5065496	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5067174	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5067371	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5068547	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5068713	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5070906	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5071594	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5083949	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5084286	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5085626	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5085798	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY
14:54:36,5087217	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	REPARSE	Desired Access: Read
14:54:36,5087380	unknown.exe	3008	RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	SUCCESS	Desired Access: Read, Disposition: REG_OPENED_EXISTING_KEY

Obrázek 41: Viditelné přepsání klíčů v registru. Zdroj: vlastní

Malware také manipuluje s proxy nastavením, konfigurací zóny intranetu a s nastavením automatického detekování zóny v prohlížeči.[6] To jsme schopni zjistit z cesty k registru.



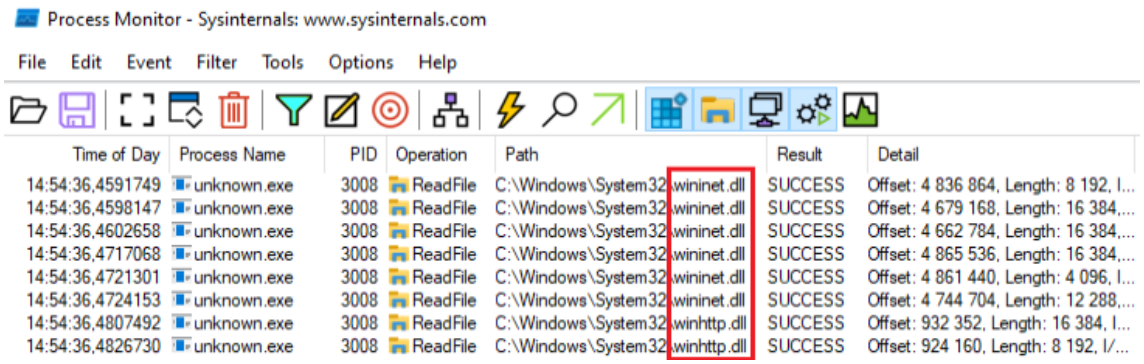
Time of Day	Process Name	PID	Operation	Path	Result
14:54:36,4935809	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Content\CachePrefix	SUCCESS
14:54:36,4945608	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Cookies\CachePrefix	SUCCESS
14:54:36,4961287	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History\CachePrefix	SUCCESS
14:54:36,5005787	unknown.exe	3008	RegCreateKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\	SUCCESS
14:54:36,5017774	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS
14:54:36,5017981	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS
14:54:36,5018096	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS
14:54:36,5018199	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS
14:54:36,5028773	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS
14:54:36,5028954	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS
14:54:36,5029061	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS
14:54:36,5029161	unknown.exe	3008	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS

Obrázek 42: Odhalení chování malwaru pomocí cesty k registru. Zdroj: vlastní

Můžeme tedy očekávat např. změnu zabezpečení v internetovém prohlížeči, nebo jeho samotného chování.

Jako poslední filtr si zadáme např. operaci ReadFile, abychom zjistili, zda si malware čte informace z nějakého souboru.

[6] Určité pasáže byly vygenerovány pomocí umělé inteligence.

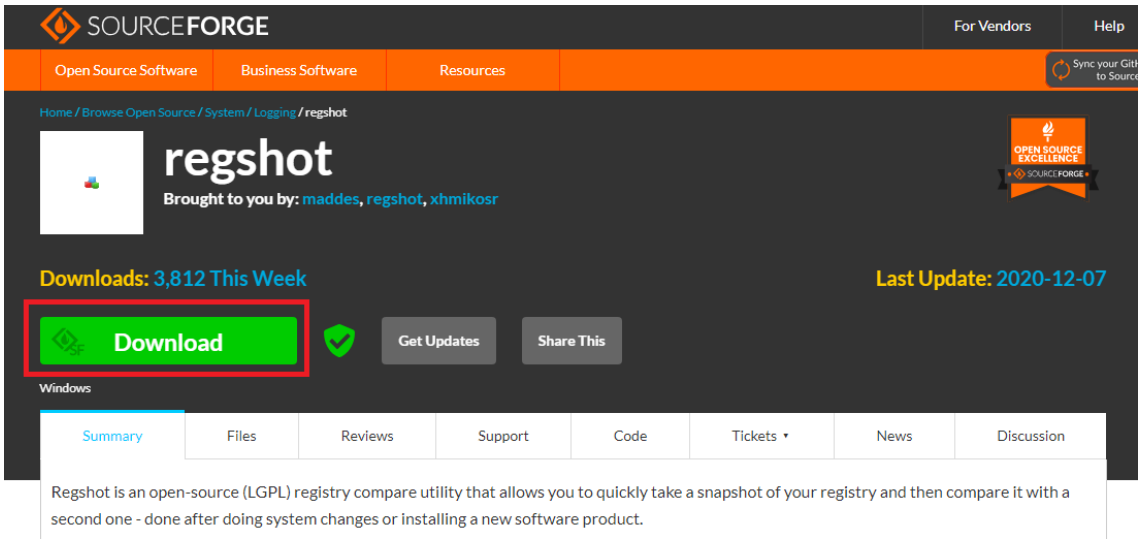


Obrázek 43: DLL knihovny v cestě k souboru. Zdroj: vlastní

A vidíme, že čte informace z knihoven *wininet.dll* a *winhttp.dll,* které poskytují prostředky pro síťovou komunikaci. To může dělat např. z důvodu získání informací o možnostech komunikace přes HTTP v rámci operačního systému. Pomocí toho může malware přizpůsobit síťovou aktivitu a komunikaci s řídícím serverem tak, aby se vyhnul detekci a analýze bezpečnostními nástroji.

4.3.3 Regshot

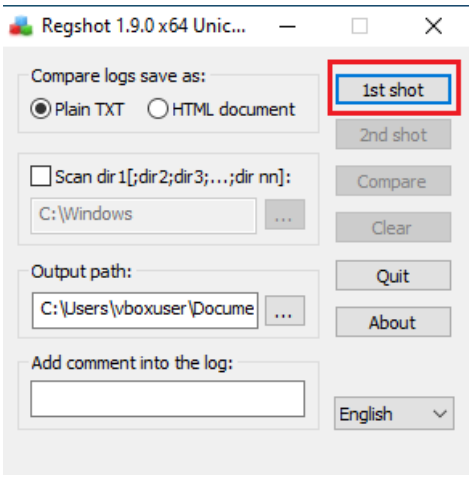
V posledním nástroji této práce se detailněji zaměříme na změny registrů u stejného vzorku malwaru, jako jsme zkoumali v předchozí kapitole. Program Regshot slouží k porovnání stavů registru systému před a po provedení určitých operací, v našem případě po spuštění malwaru. Program si stáhneme a extrahujeme.



Obrázek 44: Stažení programu Regshot. Zdroj: vlastní

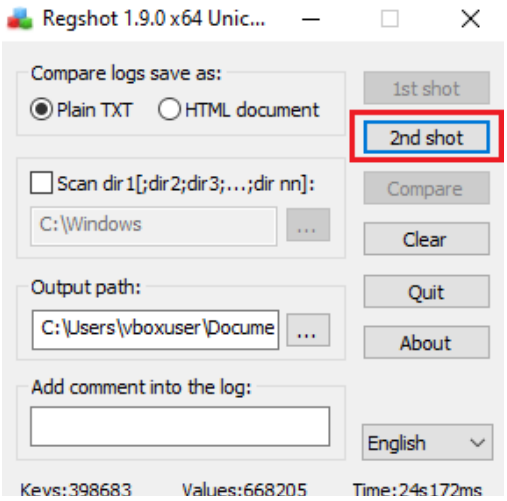
- Název testovaného malware: *unknown.exe*
- Typ testovaného malware: *Trojan, Spyware*

Otevřeme si program *Regshot-x64-Unicode.exe.* Po spuštění vybereme, v jakém formátu chceme mít výpis, složku, kterou má program skenovat a složku, kam má program uložit soubor s výpisem. Poté klikneme na tlačítko „1st shot" a vytvoří se první snímek registru se všemi zaznamenanými aktuálními hodnotami a klíči v registru.



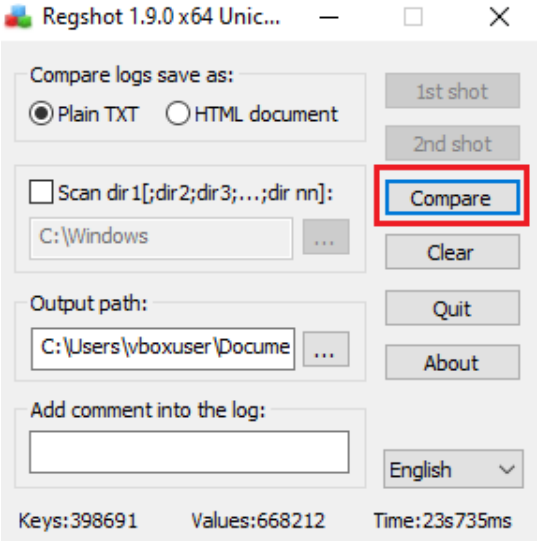
Obrázek 45: Zachycení prvního snímku v programu Regshot. Zdroj: vlastní

Jako dalším krokem je spuštění malwaru *unknown.exe*, který máme k dispozici z minulé kapitoly. Po spuštění se vrátíme do programu Regshot a zahájíme druhý snímek registru pomocí tlačítka „2nd shot“.



Obrázek 46: Zachycení druhého snímku v programu Regshot.
Zdroj: vlastní

Počkáme na vytvoření druhého snímku a poté klikneme na tlačítko „Compare“. Ještě před vytvořením textového souboru s porovnáním si dole můžeme všimnout, že přibylo 8 nových klíčů, za které bude nejspíše odpovědný náš malware.



Obrázek 47: Porovnání dvou snímků v programu Regshot.
Zdroj: vlastní

Po kliknutí na tlačítko „Compare“ se nám otevře textový soubor se změnami v registru před a po otevření malwaru. Na začátku dokumentu se nachází základní

informace jako číslo verze programu, komentáře, datum a čas, název PC a název uživatelského účtu, na kterém byly snímky pořízeny. Pod tímto oddílem se už nachází změny mezi pořízenými snímky, konkrétně přidané klíče. Některé důležité z nich si rozebereme níže.

```
Regshot 1.9.0 x64 Unicode
Comments:
Datetime: 2024/3/4 13:16:03   ,   2024/3/4 13:18:32
Computer: WIN10NEW , WIN10NEW
Username: vboxuser , vboxuser

----------------------------------
Keys added: 8
----------------------------------
HKLM\SOFTWARE\Microsoft\Security Center\Svc\Vol
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\PackageVolumes\1\MutablePackagesOnline
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Orchestrator\InstallAtShutdown
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8688
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8748
HKU\S-1-5-21-3972257930-3082822804-3131636512-1000\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\JumplistData
HKU\S-1-5-21-3972257930-3082822804-3131636512-1000\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows
\CurrentVersion\AppModel\SystemAppData\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\PersistedStorageItemTable
\CurrentWorkingDirectory
HKU\S-1-5-21-3972257930-3082822804-3131636512-1000_Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion
\AppModel\SystemAppData\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\PersistedStorageItemTable
\CurrentWorkingDirectory
```

Obrázek 48: Výpis přidaných klíčů v programu Regshot. Zdroj: vlastní

- **HKLM\SOFTWARE\Microsoft\Security Center\Svc\Vol**
  - Tento klíč může souviset s nastavením Centra zabezpečení systému Windows. Nejspíše se jedná o údaj úrovně ochrany systému nebo stavu antivirového systému.
- **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Orchestrator\InstallAtShutdown**
  - Pravděpodobně se týká plánování aktualizací Windows. Obsahuje informace o tom, zda jsou nainstalovány aktualizace při vypnutí systému.
- **HKLM\SOFTWARE\Microsoft\Windows\WindowsErrorReporting\TermReason\8688** a **\8748**
  - Tyto klíče jsou spojeny s Windows Error Reportingem a mohou obsahovat kódy nebo důvody, proč se daný proces ukončil.
- **HKU\S-1-5-21-3972257930-3082822804-3131636512-1000\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\JumplistData**

- Tento klíč je spojen s historií vyhledávání nebo seznamem nedávných položek v Průzkumníku Windows.[7]

Následující dvě sekce výpisu se týkají smazání a přidání hodnot klíčů.

```
Values deleted: 2
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\BITS\PerfMMFileName: "Global\MMF_BITS03a86584-b24d-4f59-a37d-41489e453885"
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\VolatileNotifications\41C64E6DA387D055:  01 00 04 80 44 00 00 00 50 00 00
00 00 00 00 00 14 00 00 00 02 00 30 00 02 00 00 00 00 00 14 00 03 00 00 00 01 01 00 00 00 00 00 05 12 00 00 00 00 00 14 00
00 00 01 00 01 01 00 00 00 00 00 05 12 00 00 00 01 01 00 00 00 00 00 05 12 00 00 00 01 01 00 00 00 00 00 05 12 00 00 00 20
00 00 00

----------------------------------
Values added: 9
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8688\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8688\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8688\CreationTime:  5B 87 7E 3A 36 6E DA 01
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8748\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8748\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8748\CreationTime:  F7 A9 71 3A 36 6E DA 01
HKLM\SYSTEM\ControlSet001\Control\WMI\Security\c688cf83-9945-5ff6-0e1e-1ff1f8a2ec9a:  01 10 00 01 14 4B B0 05 50 00 00 00 00
00 00 05 5C C0 00 00 00 00 00 01 14 40 00 00 00 00 00 03 34 40 00 00 00 00 00 00 02 20 00 02 20 00 00 00 01 10 00 00 00 00
00 01 14 40 00 01 18 80 00 00 0D D0 00 00 02 20 00 00 01 10 02 20 00 00 00 00 00 00 00 00 00 01 13 30 00 00 04 40 00 00 00
00 00 02 20 00 00 00 00 00
HKLM\SYSTEM\CurrentControlSet\Control\WMI\Security\c688cf83-9945-5ff6-0e1e-1ff1f8a2ec9a:  01 10 00 01 14 4B B0 05 50 00 00
00 00 00 00 05 5C C0 00 00 00 00 00 01 14 40 00 00 00 00 00 03 34 40 00 00 00 00 00 00 02 20 00 02 20 00 00 00 01 10 00 00
00 00 00 01 14 40 00 01 18 80 00 00 0D D0 00 00 02 20 00 00 01 10 02 20 00 00 00 00 00 00 00 00 00 01 13 30 00 00 04 40 00
00 00 00 00 02 20 00 00 00 00 00
HKU\S-1-5-21-3972257930-3082822804-3131636512-1000\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\JumplistData
\windows.immersivecontrolpanel_cw5n1h2txyewy!microsoft.windows.immersivecontrolpanel:  33 6E 9E 3A 36 6E DA 01
```

Obrázek 49: Výpis smazaných a upravených hodnot klíčů v programu Regshot. Zdroj: vlastní

První smazaná hodnota má za následek úpravu funkčnosti BITS, což je služba, která poskytuje přenos dat v pozadí mezi klientem a serverem. To může ovlivnit procesy jako aktualizace systému, stahování nebo synchronizace dat. Druhá smazaná hodnota klíče souvisí s oznámeními ve Windows. Můžeme očekávat buď porušení funkčnosti oznámení nebo pokus o skrytí činnosti malware.

Co se týče přidání nových hodnot, tak prvních šest souvisí opět s Windows Error Reportingem, kde malware pravděpodobně manipuloval s informacemi o chybách a ukončení, to může mít za následek zkreslené nebo falešné zprávy o chybách. Další dva jsou spojeny se zabezpečením WMI (Windows Management Instrumentation). Lze opět očekávat změny v systémových nastavení a povolení, což může ovlivnit správu a monitorování systému. Celkový počet změn v registrech bylo 87.

[7] Určité pasáže byly vygenerovány pomocí umělé inteligence.

5 Shrnutí výsledků

V rámci praktické části práce byly otestovány vzorky malware v jednotlivých programech či nástrojích pro statickou a dynamickou analýzu.

6 Závěry a doporučení

V rámci bakalářské práce byl představen základní pohled na historii kybernetické bezpečnosti v České republice včetně legislativního rámce v podobě zákona o kybernetické bezpečnosti. Dále byla představena nová směrnice NIS2, která je nyní velmi aktuálním tématem v oblasti kybernetické bezpečnosti. Také byly popsány modely informační bezpečnosti, konkrétně CIA triáda a Parkerian Hexad. Co se týče analýzy rizik, tak v této práci byly představeny pojmy jako je zranitelnost, riziko, hrozba aj. Následně výběr a popis jednotlivých typů malware, mezi které patří např. červ, ransomware nebo trojský kůň. Na základě vybraného tématu byly techniky analýzy popsány nejen teoreticky, ale i prakticky. Pro statickou analýzu byly využity programy a nástroje jako je HashMyFiles, VirusTotal, PEID, UPX a Strings. Pro dynamickou pak Wireshark, Process Monitor a Regshot.

Závěrem je třeba zmínit, že během analýz došlo ke stažení a spuštění škodného malwaru a celé virtuální prostředí bylo infikované ransomwarem a nemožné nadále používat. Proto je doporučeno testovat a analyzovat právě ve virtuálním prostředí, které umožňuje minimalizovat rizika pro reálné prostředí a zároveň poskytuje bezpečný prostor pro experimentaci.

7 Seznam použité literatury

[1] HRŮZA, Petr. *Kybernetická bezpečnost*. Brno: Univerzita obrany, 2012. ISBN 978-80-7231-914-5.

[2] *Věcný záměr zákona o kybernetické bezpečnosti*. 2012, číslo 382. Dostupné také z: https://www.govcert.cz/download/legislativa/container-nodeid-926/vecny-zamer-final-vlada.pdf

[3] *Memorandum o Computer Security Incident Response Team České republiky*. Praha: Ministerstvo vnitra České republiky. 2010. s. 4. Čj. MV-106696 OKB-2010.

[4] O sdružení. CZ.NIC [online]. [cit. 2023-08-11]. Dostupné z: https://www.nic.cz/page/351/

[5] *Strategie pro oblast kybernetické bezpečnosti České republiky na období 2012–2015.* In: 781. 2012. Dostupné také z: https://nukib.cz/download/publikace/strategie_akcni_plany/strategie_kb_2012-2015.pdf

[6] FEIX, Miroslav a Salibor PROCHÁZKA. *Recent Objectives of Cyber Defence in the Department of Defence.* Vojenské rozhledy [online]. 2017, 26(3), 31-50 [cit. 2023-08-11]. ISSN 12103292. Dostupné z: https://www.vojenskerozhledy.cz/kategorie-clanku/vystavba-ozbrojenych-sil/aktualni-ukoly-kyberneticke

[7] Zákon č. 181/2014 Sb.: Zákon o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti) [online]. 2014 [cit. 2023-08-11]. Dostupné z: https://www.zakonyprolidi.cz/cs/2014-181

[8] Legislativa KB: Zákon č. 181/2014 Sb., o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti). *NÚKIB* [online]. [cit. 2023-08-12]. Dostupné z: https://www.nukib.cz/cs/kyberneticka-bezpecnost/regulace-a-kontrola/legislativa/

[9] E. GLADDEN, Matthew. *The handbook of information security for advanced neuroprosthetics.* 2. vydání. 2017. ISBN 978-1-944373-09-2.

[10] CARLSON, Kristofer. *The CIA Triad: The Key to Understanding Information Technology*. 2021.

[11] PENDER-BEY, Georgie. *The Parkerian hexad: The CIA Triad Model Expanded.*

[12] REID, Randall C. a Arthur H. GILBERT. Using the Parkerian Hexad to introduce security in an information literacy class. In: *2010 Information*

Security Curriculum Development Conference [online]. New York, NY, USA: ACM, 2010 [cit. 2023-08-16]. ISBN 9781450302029. Dostupné z: https://www.researchgate.net/profile/Q-Kharma/publication/334184776_Secure_Medical_Internet_of_Things_Framework_based_on_Parkerian_Hexad_Model/links/5df37894a6fdcc28371d8e39/Secure-Medical-Internet-of-Things-Framework-based-on-Parkerian-Hexad-Model.pdf

[13] JIRÁSEK, Petr, Luděk NOVÁK a Josef POŽÁR. *Výkladový slovník kybernetické bezpečnosti: Cyber security glossary*. Třetí aktualizované vydání. Praha: Policejní akademie ČR v Praze, 2015. ISBN 978-80-7251-436-6.

[14] SIKORSKI, Michael a Andrew HONIG. *Practical malware analysis: the hands-on guide to dissecting malicious software*. San Francisco: No Starch Press, c2012. ISBN 978-1-59327-290-6.

[15] GANDOTRA, Ekta, Divya BANSAL a Sanjeev SOFAT. Malware Analysis and Classification: A Survey. *Journal of Information Security* [online]. 2014, **05**(02), 56-64 [cit. 2023-08-19]. ISSN 2153-1234. Dostupné z: doi:10.4236/jis.2014.52006

[16] WONG, Reginald. *Mastering Reverse Engineering: Re-engineer your ethical hacking skills.* 2018. ISBN 978-1788838849.

[17] Co je to Disassembler? *IT-SLOVNÍK.CZ* [online]. [cit. 2023-08-21]. Dostupné z: https://it-slovnik.cz/pojem/disassembler

[18] *What are the advantages and disadvantages of static and dynamic disassembly?* [online]. [cit. 2023-08-25]. Dostupné z: https://www.linkedin.com/advice/0/what-advantages-disadvantages-static-dynamic-disassembly

[19] Tahir, R. "A Study on Malware and Malware Detection Techniques." *International Journal of Education and Management Engineering (IJEME),* Vol. 8, No. 2, 2018, pp. 20-30. DOI: 10.5815/ijeme.2018.02.03.

[20] Co je adware a jak ho spolehlivě odstranit? *ESET* [online]. [cit. 2023-08-28]. Dostupné z: https://www.eset.com/cz/adware/

[21] Co je spyware a jak ho spolehlivě odstranit? *ESET* [online]. [cit. 2023-08-28]. Dostupné z: https://www.eset.com/cz/spyware/

[22] Co je botnet? *ESET* [online]. [cit. 2023-08-29]. Dostupné z: https://www.eset.com/cz/botnet/

[23] Co je ransomware a jak se proti němu bránit? *ESET* [online]. [cit. 2023-08-29]. Dostupné z: https://www.eset.com/cz/ransomware/

[24] Viruses, Worms, and Other Malware. In: *Essential Computer Security* [online]. Elsevier, 2006, s. 41-52 [cit. 2023-08-30]. ISBN 9781597491143. Dostupné z: doi:10.1016/B978-159749114-3/50010-3

[25] Co je počítačový červ? *AVAST* [online]. [cit. 2023-08-30]. Dostupné z: https://www.avast.com/cs-cz/c-computer-worm

[26] Trojský kůň: Jak odstranit tento vir nejen z mobilu? *ESET* [online]. [cit. 2023-08-30]. Dostupné z: https://www.eset.com/cz/trojsky-kun/

[27] Trojský kůň. *AVAST* [online]. [cit. 2023-08-30]. Dostupné z: https://www.avast.com/cs-cz/c-computer-worm

[28] Co je malware? Jak se zbavit malwaru? *ESET* [online]. [cit. 2023-08-30]. Dostupné z: https://www.eset.com/cz/malware/

[29] Co je to Rootkit a jak ho odstranit. Avast [online]. [cit. 2023-08-31]. Dostupné z: https://www.avast.com/cs-cz/c-rootkit

[30] 82/2018 Sb. Vyhláška o kybernetické bezpečnosti. In: *Sbírka zákonů*. 2018. ISSN 1211-1244.

[31] NÁRODNÍ ÚŘAD PRO KYBERNETICKOU A INFORMAČNÍ BEZPEČNOST. Zpráva o činnosti 2022. Praha, 2023, 30 s. Dostupné také z: https://www.nukib.cz/download/publikace/zpravy_o_stavu/Zprava_o_cinnosti_NUKIB-2022.pdf

[32] O NÚKIB. *Národní úřad pro kybernetickou a informační bezpečnost* [online]. [cit. 2023-09-30]. Dostupné z: https://nukib.cz/cs/o-nukib/

[33] The Confidentiality, Integrity, Availability (CIA) triad. In: r*esearchgate.net* [online]. [cit. 2023-09-28]. Dostupné z: https://www.researchgate.net/publication/346192126/figure/fig1/AS:961506053197825@1606252315731/The-Confidentiality-Integrity-Availability-CIA-triad_W640.jpg

[34] Parkerian-hexad-model. In: *cleverandsmart.cz* [online]. [cit. 2023-10-20]. Dostupné z: https://www.cleverandsmart.cz/wp-content/uploads/Parkerian-hexad-model-600x604.jpg

[35] Začátkem srpna dojde u NBÚ k zásadní změně. *NBÚ* [online]. [cit. 2023-10-20]. Dostupné z: https://www.nbu.cz/cs/aktualne/1192-zacatkem-srpna-dojde-u-nbu-k-zasadni-zmene/

[36] VÍT, Svatopluk. Směrnice NIS2 přinese změny IT procesů, přijde přibližně za rok. *Root.cz* [online]. 2023 [cit. 2023-10-23]. Dostupné z:

https://www.root.cz/clanky/smernice-nis2-prinese-zmeny-it-procesu-prijde-priblizne-za-rok/

[37] NÚKIB. *Nová směrnice EU o bezpečnosti sítí a informací* [online]. [cit. 2023-10-23]. Dostupné z: https://osveta.nukib.cz/course/view.php?id=145

[38] SAFETICA. *NIS2: Rozsah, účel a jaké změny očekávat* [online]. 2023 [cit. 2023-10-27]. Dostupné z: https://www.safetica.com/cs/blog/nis2-rozsah-ucel-a-jake-zmeny-ocekavat

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání bakalářské práce

Autor:	**Radek Netolický**
Studium:	I2100252
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	**Moderní techniky analýzy malware**
Název bakalářské práce AJ:	Malware analysis

Cíl, metody, literatura, předpoklady:

Cílem bakalářské práce je představení problematiky analýzy malware pro zajištění parametrů důvernosti, dostupnosti a integrity dat, tedy základním stavebním kamenům zajištění bezpečnosti. V teoretické části autor představí a podrobně popíše principy bezpečnosti s důrazem na hrozby typu malware a jejich dopad na zajištění bezpečnosti. V praktické části pak autor vytvoří praktická řešení dílčích úloh statické a dynamické analýzy malware dle předem definovaných usecase.

Wireshark for Security Professionals. Hoboken: John Wiley, 2017. ISBN 978-1-118-91821-0.

Zadávající pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Ing. Tomáš Svoboda, Ph.D.
Oponent:	Ing. Lubomír Almer, Ph.D.
Datum zadání závěrečné práce:	15.10.2021