Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Analýza malware

2017

Tomáš Orlík

Vedoucí práce: Mgr. Petr Kajča, Ph.D.

Studijní obor: Informatika, prezenční forma

Bibliografické údaje

Autor:	Tomáš Orlík
Název práce:	Analýza malware
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2017
Studijní obor:	Informatika, prezenční forma
Vedoucí práce:	Mgr. Petr Kajča, Ph.D.
Počet stran:	40
Přílohy:	1 DVD
Jazyk práce:	český

Bibliograhic info

Author:	Tomáš Orlík
Title:	Malware analysis
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2017
Study field:	Computer Science, full-time form
Supervisor:	Mgr. Petr Kajča, Ph.D.
Page count:	40
Supplements:	1 DVD
Thesis language:	Czech

Anotace

Práce popisuje analýzu malware od jeho obdržení až po důsledky. Zabývá se detailním popisem chování konkrétního vzorku malware, pro který byly využity nástroje spojené s analýzou.

Synopsis

The thesis describes malware analysis from its receiving to the consequences. It is dealing with the detailed behaviour description of particular malware sample and tools for analysis, which were used.

Klíčová slova: malware; analýza; virus;

Keywords: malware; analysis; virus;

Děkuji svému vedoucímu Mgr. Petru Krajčovi, Ph.D. za obecné rady a vedení mé práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam tabulek

Seznam zdrojových kódů

1 Úvod

Počítačové programy jsou vytvářeny, aby člověku pomohly a usnadnily práci. Podobně je tomu i u malware, který je vytvořen k tomu, aby pomohl dosáhnout cíle svému tvůrci. Rozdíl se skrývá v jejich povaze. Běžný software ovládá uživatel a slouží k jeho prospěchu legitimní cestou, avšak malware je autonomní a přináší prospěch pouze svému tvůrci, přičemž páchá zlo a kriminální činy. První část práce je věnována malware z obecného pohledu, dále se dozvíme, jak takový malware získat a následně analyzovat jeho chování.

Téma autor zvolil pro své osobní zaměření na operační systémy či sítě a s nimi související bezpečnost. Bezpečnost je populární téma, avšak bezpečnost v oblasti informačních technologií je pro běžného člověka téma nedostižné. Zároveň taky opomíjené, protože jedničky a nuly nejdou na první pohled vidět a zlo fungující v této podobě lze z počátku lehce přehlédnout. Pravdou však je, že následky těchto hrozeb mohou být mnohem větší, jelikož žijeme v globalizovaném digitálním světě.

Na internetu dnes existují služby k online analýze malware. Jednou z nich je například malwr.com. Cílem práce je analýza chování vybraného viru, zejména zjištění jeho chování, akcí prováděných na počítači a způsob komunikace. Rozdíl mezi online analýzou a analýzou v práci bude v popisu postupů, technik a nástrojů, lépe rozebraná komunikace s okolím a to jaká data a kam odesílá.

2 Malware

2.1 Definice

Malware je termín, kterým se označuje jakýkoliv software, který je nainstalovaný na cílovou stanici a vykonává nechtěné úlohy, často ve prospěch třetí strany. Malwarem můžou být označeny programy od jednoduchých obtěžujících oken s reklamou, způsobujících invazi na počítač, až po zničující útoky, kterými jsou například krádež hesla a dat nebo infikování ostatních zařízení na síti. Kromě již uvedeného, některý typ malware je vytvořen pro to, aby odesílal informace o procházení webu reklamním společnostem nebo jiné třetí straně, bez vědomí uživatele.[1]

2.2 Pojem slova malware

Malware označuje veškerý škodlivý kód, který je použit pro to, aby poškodil oběť, ať už sběrem citlivých informací, odcizením, snížením výkonu či pouhým zobrazením nechtěné reklamy. Pojem malware vychází z anglických slov malicous a software, což v překladu do českého jazyka znamená zlomyslný, podlý nebo také zákeřný software. Pod pojem malware lze zařadit nejen počítačové viry, které jsou všeobecně známy a proti kterým se uživatelé chrání, ale také červy, rootkity, backdoor, ransomware a další. Kdo malware tvoří? Malware nevzniká v antivirových společnostech pro jejich větší zisk, ale často může být produktem studentů programátorů, kteří si chtějí otestovat svůj kód nebo jen pro zábavu, přičemž největším podílem přispívají tzv. black-hat neboli crackeři. Zde je potřeba si dát pozor a nezaměňovat s hackery, kteří jsou tzv. white-hat a zabývají se především penetračními testy.

2.3 Historie

Historie malware se píše již od vzniku čtvrté generace počítačů (začátek 80. let 20. stoeltí) a to zejména na univerzitách. Zde bylo vždy zapotřebí sdílet data mezi jednotlivými počítači a proto dochází k rozmachu malware. První malware se šířily pomocí disket.[2] Je zapotřebí si uvědomit rychlost takového šíření mezi jednotlivými stanicemi oproti následující fázi nastupující v 90. letech 20. století, v níž zažívá svůj největší boom internet a od této doby se viry šíří napříč celým světem. Koncem 90. let začínají útoky v podobě emailových červů cílit i na domácí uživatele a postupem k 21. století se v útocích stále pokračuje, jelikož jejich autoři jsou motivováni zejména finančním ziskem. Na uživatele čekají nečekaná a nekontrolovatelná pop up okna a další nástrahy v podobě Javascriptu. Rok 2002 se objevuje botnet Agobot.[3] Další hrozbou je rootkit, který se v roce 2005 objevuje u společnosti Sony. Jedná se o klamnou ochranu proti kopírování CD. Tato chyba se týkala zhruba 22 miliónů nosičů. Škodlivý kód byl v tomto případě součástí digital rights management (DRM) a vmístil se do operačního systému. Následně se Sony omlouvá a vydává uninstaller.[4] Na začátku útoku

s masivním dopadem je však backdoor, ukradené přihlašovací údaje k FTP účtu. To však vede k ještě rychlejšímu šíření malware, protože je možné využít kompromitujících webových stránek k šíření a není potřeba složitě doručovat malware do cíleného počítače. Nezávislý bezpečnostní IT institut AV-TEST zaznamenává množství unikátních malware kódů od roku 1990. Jednotlivé vzorky malware jsou od sebe odlišeny pomocí kryptografické hashovací funkce MD5. Vzestupnou tendenci lze vidět v následující tabulce.

Tabulka 1: Množství malware postupem času

Rok	Unikátní počet kódů
1990	9044
1994	28613
1999	98428
2005	333425
2006	972606
2007	5490960

S nárustem malware souvisí i zisky antivirových a bezpečnostních společností, které se v období 2000 - 2010 zvýšily desetinásobně a to na hodnotu 16,5 mld. amerických dolarů.[2][5][6]

2.4 Typy malware

2.4.1 Virus

2.4.1.1 Popis

Virus je nejstarší a nejznámější podoba malware. Sám o sobě je virus de facto neškodný, potřebuje k sobě hostitele ve formě spustitelného souboru nebo makra v Microsoft Office. V současné době jsou počítačové viry spíše na ústupu. Je proto potřeba se chránit i před nevirovými hrozbami jako například červy z internetu.

2.4.1.2 Historie

První škodlivý program, který lze přirovnat k viru, se objevil v roce 1982. Elk Cloner program napsal patnáctiletý Rich Skrenta. Jednalo se o samoreproduktivní virus boot sektoru, který nakazil počítače Apple II skrze disketu. Jeho funkce spočívala v tom, že při každém padesátém spuštění počítače se objevila básnička o infiltraci počítače. Program nebyl nazván virem, protože se jeho šíření zastavilo v okruhu Skrentových kamarádů.

První vir pocházel od Freda Cohena z roku 1983. V rámci bezpečnostního semináře na Lehigh University v Pensylvánii demonstroval Cohen svůj kód a během pěti minut od načtení převzal kontrolu nad systémem. Len Adleman přirovnal samoreproduktivní program k viru a tím dal základ novému termínu počítačový virus.[7]

2.4.1.3 Chování

Viry jsou v různých formách, ale všechny mají pravděpodobně dvě společné fáze. Fázi infekce a fázi útoku. Fáze infekce je iniciována spuštěním samotného viru nebo spouští, kterou může být den nebo čas, jiná souběžná akce prováděná na počítači, počítadlo v rámci viru. Autoři viru však chtějí, aby se vir šířil tak rychle, aby nebyl zachycen. V rámci útočné fáze virus páchá nechtěné akce, kterými může být náhodná změna dat na disku, snižování výkonu počítače, simulace psaní nebo méně závažné akce jako přehrávání hudby, animace na obrazovce a jiné.[8]

2.4.2 Červ - worm

Počítačový červ je na rozdíl od počítačového viru zcela autonomní. Jedná se o vlastní program, který se šíří nejčastěji pomocí počítačové sítě na jednotlivé počítače, kde se replikuje a postupuje dál. Do jednotlivých počítačů se červi dostanou využitím bezpečnostních chyb. V napadeném počítači se instaluje škodlivý software, který se spouští s každým spuštěním systému.

2.4.3 Spyware

Spyware je program, který shromažďuje statistické data o napadeném uživatelově počítači a odesílá je přes internet. Tato data se obvykle získávají ze souborů cookies nebo z historie prohlížeče. Nicméně je však možné, že spyware nainstaluje další nevyžádaný software, bude zobrazovat reklamy nebo přesměruje webovou aktivitu na nechtěný obsah. Spyware se na rozdíl od virů a počítačových červů neumí sám replikovat a šířit. Z toho vyplývá, že je distribuován samotnými útočníky. Tento druh malware může najít své uplatnění v komerční sféře. Žádaná data se mohou schovávat v cookies nebo v offline HTML obsahu v Temporary Internet Files adresáři. Je tedy složité jej nalézt a odstranit.

2.4.4 Trojský kůň

Trojský kůň je program, který se na první pohled tváří být legitimním a v pořádku. Avšak jeho kód obsahuje skrytý malware. Typickým příkladem často bývá program na odstraňování malware. Jiný typ koně se může skrývat v programu, který se stahuje z neověřených zdrojů. Kód takového programu může být modifikován a právě zde se může skrývat trojský kůň. Tento rádoby užitečný nástroj je pak uživatelem spuštěn. Microsoft Windows tomuto typu malware nevědomky napomáhají, protože v základním nastavení jsou skryty přípony souborů. Neznalý uživatel si takto velmi jednoduše nainstaluje například erotický spořič obrazovky.

2.4.5 Keylogger

Keylogger zaznamenává uživatelovy stisknuté klávesy a sbírá tak citlivé informace jako jsou uživatelská jména, hesla, čísla platebních karet aj. Keylogger může být použit i legitimním způsobem a to například instalací IT oddělením

jako monitorovací nástroj. Mnohem častěji se software do počítače dostane ilegálním způsobem a je součástí spyware. Samotný keylogger se může nacházet na různých místech - v operačním systému, v API vrstě, v paměti nebo v jádře systému. Posbíraná data odesílá jako běžná data, proto je obtížné tuto komunikaci rozeznat.[9]

2.4.6 Backdoor

Tzv. zadní vrátka. Backdoor se instaluje pomocí trojského koně nebo červa. Útočník takto napadený počítač vzdáleně ovládá a používá pro rozesílání emailů nebo pro tvorbu botnet. Botnet je síť zombie, počítačů napadených backdoor, řízených centrálně z jednoho místa, využívána pro Distributed Denial of Service - DDoS útoky. Backdoor využitý pro DDoS útoky se stávají největší reálnou hrozbou budoucnosti internetu věcí. Internet of Things - IoT je síť jednoduchých zařízení propojených internetem, komunikujících se svým řídícím zařízením. Tato zařízení lze obvykle koupit za pár desítek dolarů, proto nelze očekávat nějaké zabezpečovací mechanismy. Hesla v těchto jednoduchých zařízeních nelze změnit, protože jsou součástí firmware. Koneckonců ani výkon nelze očekávat, ale ve spojení mnoha zařízení z různých míst, lze takto velmi kvalitně a jednoduše vytvořit DDoS útok. V říjnu roku 2016 byl takto odhalen útok z kamerových zabezpečovacích systémů pod názvem Mirai a ve špičce útoku dosahoval datový tok až 620 Gbps.[10] Na začátku března roku 2017 to byl backdoor kamerových systémů Dahua. Tyto kamerové systémy vyrábí světový gigant Hikvision.[11]

2.4.7 Rootkit

2.4.7.1 Popis

Pojem rootkit vznikl spojením slov root jako privilegovaný uživatel Unixu a kit - sada. Jedná se tedy o balíček programů maskujících jiný malware. Jakmile jednou získá útočník přístup k administrátorskému účtu, má vyhráno. Zaměřuje se na známé zranitelné složky systému nebo na hesla pomocí sociálního inženýrství tzv. phishingu. Klíčem je tedy získat účet root neboli účet ze skupiny Administrators. Rootkit poté může modifikovat existující software, dělat zásahy do kernelu, kde se může následně ukrýt. Pro jeho detekci je nutný sekundární a důvěryhodný operační systém, kde je zapotřebí statické behaviorální detekce malware a analýzy memory dump.[12] Odstranění může být nereálné nebo je nutná reinstalace operačního systému. V případě napadení firmware je zapotřebí výměna komponenty nebo specializovaného nástroje.

2.4.7.2 Techniky skrývání rootkitu

Rootkit se může ukrývat na různých úrovních abstrakce.

- Uživatelský prostor. Rootkit zde figuruje jako uživatelská úloha (steganografie), obvykle napadne existující proces a přepíše paměť aplikace vlastním

obsahem.

- Jaderný prostor. V jaderném prostoru může modifikovat jádro operačního systému, ovladače, System Service Descriptor/Dispatcher Table, Interrupt Descriptor Table, I/O Request Packet Function. Kvůli implementaci do operačního systému je značně ztížena detekce.

- Hypervizor. Rootkit se může ukrývat v hypervizoru, který modifikuje nebo nahradí, a tím získá přístup k virtuálním počítačům. Princip ukrývání na úrovni hypervizoru není v současné době znám, ale byl již koncepčně dokázán.

- Firmware. Nejhůře odhalitelný typ rootkitu. Skrývá se v samotném hardwaru, je jeho součástí. Může být například součástí BIOSu, firmware routeru a jiných. Mnohdy nelze odstranit pouhou reinstalací systému, je zapotřebí hardware vyměnit.

2.4.7.3 Detekce

Zatímco se rootkity stávají sofistikovanějšími a různorodějšími, je zapotřebí různých taktik a nástrojů na jejich odhalení.

- Důvěryhodný sekundární operační systém. Jakmile je napaden operační systém, je zapotřebí jiného operačního systému. Tím může být systém načtený z CD. Výhodou je, že CD nelze přepsat.

- Behaviorální analýza. Jakmile se rootkit dostane do počítače, nastane snížení výkonu, které je někdy detekovatelné. V případě této analýzy je zapotřebí mít identický hardware, software a měřit čas API volání.

- Kontrola integrity. Klíčovou analýzou může být kontrola záznamů v registru Windows a porovnání s čistým, nenapadeným systémem.

- Porovnání rozdílů. Porovnání dat na disku a dat načtených do operační paměti, jestli jsou skutečně stejné nebo zda došlo během nahrání k jejich modifikaci.

- Výpis paměti. Detekci rootkitů lze také provést analýzou při výpisu paměti, protože rootkit nemá šanci detekovat a zablokovat analýzu. K získání správného výpisu může být potřeba samostatný hardware.[13][14]

2.4.8 SQL injection attack

2.4.8.1 Popis

Spočívá v napadení databáze vsunutím (odsud injection) zákeřného kódu, který modifikuje databázi. SQL injection zároveň využívá programátorovy chyby při zadávání přihlašovacích údajů. Chyba útočníkovi povolí prolomit ověřovací mecha-

nismus a poté neoprávněně zasahovat do databáze, ať už prohlížet důvěrné záznamy, modifikovat data v databázi, mazat je, vypnout systém řízení báze dat a tím učinit službu nedostupnou nebo dokonce přidat útočníkovi oprávnění administrátora.[15] SQL injection je běžným jevem u PHP a ASP aplikací kvůli stále velkému rozšíření starých rozhraní.

2.4.8.2 Technika SQL injection

Při autentizaci uživatele potřebujeme ověřit heslo k uživatelskému jménu. Představme si například jednoduchou aplikaci, která nás vybízí k zadání uživatelského jména a hesla.

Zdrojový kód 1: Technika SQL injection attack 1

```
1 uName = getRequestString("username");
2 uPass = getRequestString("userpassword");
3
4 sql = 'SELECT * FROM Users
5    WHERE Name ="' + uName + '" AND Pass ="' + uPass + '"'
```

SQL dotaz z předchozího kódu při běžném přihlášení uživatele pomocí uživatelského jména a hesla.

Zdrojový kód 2: Technika SQL injection attack 2

```
1 SELECT * FROM Users
2    WHERE Name ="Jan Novak" AND Pass ="mojeTajneHeslo";
```

Nyní využijeme vlastnosti, že výraz

```
or ""=""
```

je vždy pravdou. Vložíme jej upravený

```
" or ""="
```

do pole pro heslo. Do uživatelského jména lze napsat cokoliv, protože heslo bude vždy vyhodnoceno jako pravdivé.

Zdrojový kód 3: Technika SQL injection attack 3

```
1 SELECT * FROM Users
2    WHERE Name ="xxx" AND Pass ="" or ""="";
```

Tímto způsobem lze obejít ověřovací mechanismus. Jako výsledek dostaneme všechny všechny řádky z tabulky Users, mezi kterými jsou uživatelská jména a hesla. Hesla bývají zašifrovaná pomocí hashovací funkce, například MD5. Principem reverzního prolamování hashe lze zjistit původní heslo. [16]

Dvěma nejčastějšími typy SQL injection jsou tzv. error-based a union-based techniky. Technika typu error-based využívá chybových zpráv databázového serveru, díky kterým získá informace o struktuře databáze. V některých případech stačí pouze error-based SQL injection, aby útočník vyčetl všechny záznamy z databáze.

Druhým přístupem je union-based SQL injection technika. Tento typ spočívá ve spojení dvou a více SELECT dotazů do jednoho výsledku pomocí operátoru UNION. Mějme například internetový obchod http://www.eshop.com, kde jsou jednotlivé položky dostupné pod svým ID, které se skrývá v adrese http://www.eshop.com/products/?&itemID=2861. Přidáním

```
UNION SELECT ItemName, ItemPrice, ItemDescription
FROM Items WHERE ItemPrice < 100
```

k itemID dostaneme nejen informace o produktu jehož identifikační číslo je 2861, ale také veškeré záznamy, které jsou levnější než 100.[17] V následujícím kódu je možné vidět, jak bude vypadat SQL dotaz aplikace na databázový server, jestliže obdrží http://www.eshop.com/products/?&itemID='2861' UNION SELECT ItemName, ItemPrice, ItemDescription FROM Items WHERE ItemPrice < 100 GET požadavek.

Zdrojový kód 4: Union-based SQL injection attack

```
1 SELECT ItemName, ItemPrice, ItemDescription
2    FROM Items
3    WHERE itemID = '2861'
4 UNION
5 SELECT ItemName, ItemPrice, ItemDescription
6    FROM Items
7    WHERE ItemPrice < 100;
```

Útočníci mohou zvolit také jinou variantu spojení dvou SQL příkazů. Při nesprávném filtrování neplatných znaků, lze použít pro spojení dvou příkazů středník. V následujícím kódu si ukážeme, jak by mohl vypadat takový SQL dotaz.

Zdrojový kód 5: SQL injection attack

```
1 SELECT ItemName, ItemPrice, ItemDescription
2    FROM Items
3    WHERE itemID = 2861; DROP TABLE Users;
```

Předchozí kód nám opět zobrazí informace o produktu, ale zároveň smaže tabulku s uživateli.[18]

2.4.9 Ransomware

2.4.9.1 Popis

Ransomware je druh malware, software, který zablokuje přístup k datům počítače a žádá za přístup zaplacení výkupného. V angličtině ransom, odtud ransomware. Typickými zástupci jsou CryptoLocker, Locky, CryptoWall. Tyto vzorky šifrují data, která pak bez znalosti privátního klíče nejde přečíst. Winlocker nebo policejní virus jsou zástupci ransomware, kteří nešifrují data, ale zablokují k nim přístup.[19]

2.4.9.2 Historie

Počátky ransomware se datují od roku 1989. První vzorky se šířily pomocí disket, které byly rozposlány poštou. Jednalo se o pozemní poštu na rozdíl od elektronické. Na disketě byl soubor `autoexec.bat`, který zašifroval data pomocí symetrické šifry. V roce 2012 se objevuje tzv. "policejní virus", který nešifruje data, ale zablokuje přístup do adresáře Windows. Touto dobou si útočníci najímají lepší překladatele, aby byl útok úspěšnější a zpráva byla věrohodnější. Tento typ viru byl zároveň cílen na konkrétní státy. Ve Spojených státech se jednalo o zprávy od FBI, u nás to byla Policie České Republiky. Typickým představitelem byl Win32 /Filecoder.Q, který se u nás šířil převážně pomocí serveru uloz.to. Postupně se zlepšovala grafika viru, začalo se využívat phishingu a sociálního inženýrství. Útočníci začínali nabízet dešifrování jednoho souboru zdarma, aby ukázali uživateli, že je skutečně funkční a zároveň potřebovali přesvědčit uživatele k platbě převážně ve virtuální měně. Částky bývají dvě, jedna nižší při rychlé platbě, druhá vyšší při pozdější platbě. Dnes je moderní Ransomware as a Service. Službu lze najít na Tor síti, kde se uživatel zaregistruje a nechá si vytvořit malware na míru, uživatel si zvolí výslednou částku v BTC. Programátoři mají v tomto případě 30 % zisku. Toto řešení má stinnou stránku pro zadavatele, jelikož vir má společný základ a antivirové společnosti si hlídají dané části. Je potřeba proto vytvořit určitý dropper, který zůstane chvíli nedetekovaný.[20]

2.4.9.3 Chování

Ransomware šifruje postupně data na disku a typicky se zaměřuje na soubory s příponou .DOC, .XLS, .JPG, .ZIP, .PDF, ale dnes již lze sledovat, že některé verze jsou schopné zašifrovat CAD soubory, webové stránky, SQL databáze a soubory vytvořené účetními programy. Tyto soubory jsou po zašifrování přejmenovány a je jim přidělena koncovka .CRYPT nebo .XXX, kde X je písmeno abecedy. Záleží na konkrétním vzorku malware. Na konci šifrování je většinou změněna tapeta, která oznamuje, že je počítač napaden a je zobrazena zpráva, která vybízí postiženého uživatele k zaplacení dekryptovacího software v Bitcoin měně. Uživatel má možnost se podívat na ID peněženky do blockchain, zda již za danou hrozbu někdo zaplatil, kolik a jestli mají útočníci již dostatek financí, aby

zveřejnili privátní klíč veřejně na internetu. Poškozený uživatel může díky pseudonymní měně sledovat své peníze a vidět, za co byly utraceny. Mnohdy jsou ale převedeny na anonymní měnu, díky které jsou transakce nedohledatelné a neidentifikovatelné. Jenom za poslední dva roky oběti ransomware zaplatily za svá data výkupné přes 25 milionů dolarů.[21]

3 Sběr malware pro účel práce

3.1 Z emailové schránky se špatným spam filtrem

Je mnoho možností, jak shromáždit škodlivý kód pro potřebnou analýzu. Jedna z nejjednodušších možností je založit si emailový účet u nejnavštěvovanějšího českého vyhledávače a počkat, až se začne plnit spamem. Uživatel bude dostávat nevyžádanou poštu nabízející intimní služby, ohromné výdělky nebo jen koření života v podobě viagry. Odkazy v těle těchto zpráv v sobě nesou unikátní uživatelovo ID, proto při kliknutí na odkaz se začne emailová schránka plnit nevyžádanou poštou o to rychleji, protože uživatel tím potvrdí funkčnost a dostupnost schránky. Tento typ malware není však vhodný pro analýzu. Emaily nemívají přílohu i přesto, že u daného emailového poskytovatele si lze nechat doručit spustitelný soubor s příponou .EXE.

3.2 Z napadeného systému

Další možností je obejít pár nezabezpečených počítačů a nebo poprosit přátelé o zaslání zachyceného malware. Možnost se jeví jako zajímavá z pohledu socializace a praktické ukázky, že se konkrétní malware dostal až k hostiteli a začal působit. U malware tohoto typu je zapotřebí si uvědomit, jak se ke svému hostiteli dostal. Většinou tomu předcházel nevyžádaný email, podvodný software zaručující zrychlení systému - backdoor nebo pomocí zavirovaného flash disku. V jednom z případů analýzy malware bylo využito průniku pomocí backdoor a nebylo možné zanalyzovat kód komplexně, protože se v počítači nacházel již jen jakýsi zlomek. Malware po sobě zametl stopy a zároveň odinstaloval backdoor. Jednalo se o malware, který napojil daný PC do sítě botnet a útočil na redakční systémy s WordPress. Malware se na systému spouštěl s přihlášením každého uživatele, kde si vytvořil klon procesu, který byl však jinak pojmenovaný než výchozí. Vzhledem k použitým názvům byl viditelný a nepřehlédnutelný i ve správci úloh.

3.3 Veřejné virové databáze

Na internetu existuje mnoho veřejných internetových databází s malware různého druhu. Některé databáze k tomu připojují i možnost rychlé automatické analýzy uploadovaného kódu. Ne však každá taková databáze nabízí možnost stáhnout si kód k sobě. Je to dáno politikou dané organizace, která se snaží chránit uživatele digitálních technologií.

3.4 Honeypot

Z hlediska bezpečnosti počítačových sítí a systémů je dobré se vždy zaobírat bezpečností a její prevencí. K tomu slouží takzvaný Honeypot. Jedná se o informační systém, který včas zachytí potenciální hrozbu a upozorní na ni. Následná

analýza chování konkrétního malware pomůže lépe zabezpečit systém a připravit se na útok. Z hlediska konfigurace se jedná o jednoduchý server, který má navíc nástroje pro sledování přístupu, logování přístupu a jiné sniffing nástroje. U komplexního řešení se uvažuje celá síť, která je v demilitarizované zóně s firewallem, který zajišťuje skenování komunikace, DNS serverem, mail serverem, FTP serverem, web serverem, případně dalšími klientskými či serverovými stanicemi.[22]

3.4.1 HoneySpam

Honeypot k zachytávání spamu se nazývá HoneySpam. V lepším případě se HoneySpam skládá z webového serveru, kde se vystaví emailové adresy, open proxy serveru, za který se útočník schová, open mail relay, což je SMTP server, který nevyžaduje autentizaci uživatele a dovolí odeslat neověřené emaily kamkoliv a cílového mail serveru.

3.4.2 HoneySpam použitý pro práci

V našem případě jsme vytvořili kombinaci webového serveru s cílovým mail serverem, protože útočník nemá problém si zajistit vlastní smtp relay a na našem mail serveru nebylo požadováno ověření zpráv. Zároveň jsme měli nastavený doménový koš, jenž zjednodušil vybírání zpráv a zachycení jejich většího množství. Rovněž zde není žádný spam filtr a ani jiný filtr, který by zamezil příjem spustitelných souborů. Pro provoz mailového serveru je zapotřebí vlastnit nějakou doménu nebo být jejím správcem, mít veřejnou IP adresu a poskytovatelem internetu mít povolený port 25 v obou směrech. Tento problém se zdál být největší, protože běžní poskytovatelé internetu nechtějí mít na své síti open mail relay. Další stěžejní věcí byla existence domény. Jelikož se tato doména nechvalně zapsala do konfigurace jiných serverů, bylo vhodné si vytvořit doménu novou a nebrat již existující nebo vytvářet subdoménu. Pro náš honeypot byla pořízena doména italské generické domény .cloud, která tou dobou měla necelý rok. Zprvu se zdálo, že by se mohla stát ideálním terčem pro útočníky díky své době na trhu. Abychom podpořili nárust spamu a příchozí pošty, začali jsme používat adresy ze své domény k registraci na nejrůznějších stránkách, které měly původ zejména v Indii, Rusku a Spojených státech. Tyto státy patří mezi TOP státy světa, které rozesílají největší množství spamu.[23] Tento spam se však někdy nepodařilo zaregistrovat, tvůrci webových projektů, aplikací i nevěrohodných stránek nepřijímají při registraci emailové adresy s generickou doménou .cloud a oznámí tuto emailovou adresu za nevalidní.

3.4.3 Side HoneySpam

Při registraci domény je zapotřebí vypnit WHOIS. WHOIS je query response protokol definovaný ve standardu RFC 3912. Používá se pro záznam domény, adresního bloku, jejich zodpovědných osob aj. U osoby je mimo jiné uvedena i emailová adresa použita při registraci domény. Do jednoho dne od registrace

domény pro HoneySpam začaly pravidelně chodit spamy na jinak neaktivní emailový účet použitý při registraci. Spamy většinou nabízely tvorbu responzivních webů, SEO, tvorbu firemního loga, identity a jiné služby spojené s start-up.

4 Analýza

4.1 Prostředí pro analýzu

Pro analýzu bylo dobré vytvořit vhodné testovací prostředí. Toto prostředí zjednoduší samotnou analýzu při vhodně zvolených nástrojích. Tyto nástroje pak dovolí opakovat provedené kroky, vracet se zpět v čase do jednotlivých stavů, zkoumat přístup k souborovému systému, síti, registrům aj.

4.1.1 VirtualBox

Pro testování malware je zapotřebí mít hostitelský systém, ve kterém je možné vir spustit a dále analyzovat. Vhodným řešením je použít virtualizační program - hypervizor, ve kterém je možné si nainstalovat a nabootovat hostitelský systém. Pro práci byl použit VirtualBox od Oracle ve verzi 5.1. VirtualBox je multiplatformní hypervizor, který je distribuován pod licencí GNU GPL, i proto byl zvolen právě tento nástroj.[24] Hypervizor dokáže běžet ve dvou módech. V softwarovém módu, kdy je jako program součástí operačního systému a ostatních procesů. Tento mód je využit u levnějších notebooků, jejichž procesory nepodporují virtualizaci nebo v hardwarovém módu s potřebným hardwarem. U Intel se jedná o procesory s Intel Virtualization Technology - VT-x a u firmy AMD jde o procesory s technologií AMD Virtualization - AMD-V.

Bezespornou výhodou použití virtualizovaného systému je možnost si systém konfigurovat elektronicky a dynamicky přidělovat prostředky. Je zde možnost nastavit počet jader CPU, množství RAM, velikost grafické paměti, počet monitorů, parametry a umístění virtuálního disku nebo nastavení parametrů sítě. Hypervizor podporuje až 4 síťové karty, kdy je možnost si u každé síťové karty vybrat mód provozu neboli k čemu bude připojena. V nabídce je NAT, vnitřní síť, síť pouze s hostem, síťový most na hardwarovou kartu nebo NAT síť. Tyto možnosti najdou své využití při tvorbě rozsáhlejší sítě. Sdílení dat mezi hostitelským a hostovaným systémem je zde umožněno v několika možných variantách a to pomocí sdílených složek, sdílené sítě, sdílené schránky a režimu "Táhni a pusť". U všech možností sdílení dat lze nastavit povolený směr sdílení. V rámci práce bylo využito klonování systémů, aby byl ušetřen čas instalací nového systému pro jiný typ viru. A také možnost snapshotů nachází své využití před stažením maligního kódu, aby byla možná okamžitá obnova napadeného systému například ransomwarem.

Výhodná je virtualizace, kdy je potřeba nechat dva systémy běžet simultánně. Je však potřeba si uvědomit, že nedochází k úspoře HW prostředků v pravém slova smyslu, ale je zapotřebí virtualizovat na dostatečně výkonném stroji, který svůj výkon nevužívá po celou dobu běhu a v době nízkého vytížení může zpracovávat instrukce virtuálního systému.

4.1.2 Windows 7

Hostovaný operační systém byl Windows 7 Enterprise se Service Pack 1, verze 6.1.7601. Edice Enterprise byla zvolena díky většímu množství funkcí, oproti verzi Standard, a nižším požadavkům na hardware.

4.1.3 Microsoft Office 2010

Kancelářský balík firmy MS ve verzi Professional byl použit pro tuto práci kvůli vybraného vzorku na testování, který byl v dokumentu Word. Součástí kancelářského balíku je také vývojové prostředí Visual Basic. Programovací jazyk Visual Basic for Applications je od MS a je používán k definování uživatelských funkcí v kancelářském balíku MS Office. Díky tomu je možné automatizovat opakované úlohy, přistupovat k Windows API pomocí dynamicky linkovaných knihoven a tím spojit data z MS Access s Excelem a výsledek odeslat Outlookem.

4.1.4 Wireshark

Wireshark je světově nejrozšířenější a multiplatformní analyzátor síťového provozu.[25] K práci byl používán ve verzi 2.2.3 64-bit. Wireshark zachytává síťové pakety a zobrazuje je chronologicky za sebou. Při startu programu je vybrána síťová karta, která bude odposlouchávána. V samotné aplikaci se jednotlivé pakety zobrazují pod sebou, kde se v každém řádku nachází čas, zdrojová adresa, cílová adresa, protokol, délka a základní informace. Ve spodní části lze nalézt po výběru paketu detailnější informace a obsah paketu. Vzhledem k množství paketů, které na sítích putuje, aplikace Wireshark nabízí filtrování záznamů.

4.1.5 Sandboxie

Sandbox je program, který izoluje spouštěný program od systému, dat, ostatních aplikací, internetu, de facto všeho a pracuje jen s prostředky, které mu uživatel přidělí. Pro práci byl použit Sandboxie ve verzi 5.18, který byl vyvinut firmou Sophos. Původním autorem je Ronen Tzur. Izolované prostředí Sandboxie je pro analýzu malware velkým pomocníkem. Dokáže odstínit uživatelovy data od systému a zároveň zaručit funkčnost aplikace. Po spuštění aplikace je možnost vytvořit si vlastní sandbox, kterému je potřeba nastavit vlastnosti, mezi které řadíme: do kterých složek může přistupovat, které programy smí spouštět, jestli může komunikovat po síti, přistupovat k registrům a s jakým uživatelským oprávněním zkoumané aplikace poběží. Samotný Sandboxie potřebuje ke svému běhu oprávnění administrátora. Aplikace spuštěné v Sandboxie jsou pro zřetelnost označeny žlutým pruhem kolem celého okna, pokud uživatel najede na zápatí okna.[26]

4.1.5.1 Přístup k datům

Jestliže potřebuje aplikace spuštěná v Sandboxie přistoupit k datům a má k tomuto kroku oprávnění, vytvoří se kopie těchto dat v předem definované izolované složce, ve které probíhají změny. Programy tudíž nepracují s adresáři po celém disku, ale pouze v jedné složce. Zde je také vidět, které soubory byly modifikovány v průběhu programu. Tyto soubory mnohdy obsahují velmi důležitá data pro samotnou analýzu. Po ukončení programů v sandboxu, je uživatel dotázán, zda chce modifikovaná data odstranit, přesunout na své místo nebo přesunout do vlastní složky.

4.1.5.2 Nedostatečné oprávnění

Mnohé programy potřebují ke svému běhu systémové procesy a podpůrné programy, o jejichž spuštění uživatel netuší. Proto je při tvorbě sandboxu v Sandboxie vhodné nastavit, aby byl o těchto skutečnostech uživatel informován v podobě dialogového okna. V tomto okně je možnost jednoduchým odkliknutím přidat požadované oprávnění.

4.1.5.3 Prostředí

Ve chvíli, kdy má uživatel nastavený svůj sandbox, zobrazí se mu v hlavním ovládacím panelu. Spuštění programu je provedeno zobrazením místní nabídky, ve které si lze vybrat program z nabídky Start, jakýkoliv program podle adresy, prohlížeč internetu, emailového klienta nebo průzkumníka Windows. Spuštěný program se zobrazí v hlavním kontrolním panelu pod názvem uživatelova sandboxu. Mimojiné je zde také titulek ze záhlaví okna a PID, díky kterému je proces lépe dohledatelný. V rámci běhu programu lze procházet soubory, obnovovat je do jejich původního umístění nebo je ukončovat.

4.1.6 Process Monitor

Process Monitor je pokročilá monitorovací utilita pro operační systém Windows, která v reálném čase zobrazuje aktivitu souborového systému, práci s registry, aktivitu procesů a jednotlivých vláken. Process Monitor v sobě kombinuje dvě starší utility skupiny Sysinternals Filemon a Regmon a mnoho dalších vylepšení včetně filtrování, komplexní výpis vlastností události jako jsou session ID, parent ID, uživatelské jméno, informace o procesu a výpis zásobníku vlákna aj. Process Monitor je klíčová utilita při řešení problémů se systémem a při analýze malware.[27] Pro účely práce byla použita verze 3.31.

4.1.7 WinDbg

Windows Debugger je součástí Windows Kits Debugging Tools for Windows. Pro studium vzorku byla použita verze 10.0.15063.137 X86. Software si je možné bezplatně stáhnout na stránkách Microsoftu. V programu je možné ladit program

či proces spuštěný v operačním systému. Proces je možné spustit přímo z ladícího prostředí nebo možné připojit již běžící proces. Jakmile je proces spuštěn, je možné jej pozastavit a analyzovat. V aplikaci je několik oken, díky kterým lze sledovat paměť procesu, disassemblované jednotlivé kroky, vykonávané příkazy, systémová volání, vlákna procesu, registry s jejich hodnotami a uživatelem definované proměnné.

4.1.8 Hexplorer

Program Hexplorer ve verzi 2.6 naprogramoval Polák Marcin Dudek. Hexa prohlížeč byl použit k otevírání binárních souborů, jejich zpětnému překladu do strojového kódu a následné analýze. Prohlížeč zobrazuje ve výchozím nastavení dokument ve třech sloupcích. V levém sloupci je verze v šestnáctkové soustavě, v prostředním sloupci v ASCII a v pravém sloupci lze zobrazit nástroje, mezi které patří zobrazení vzorů, zpětný překladač do strojového kódu, šifrování, Fourierova transformace, kontrolní suma a výskyt znaků.

4.2 Testovaný malware a jeho chování

4.2.1 Popis vzorku

Pro účely testování byl použit malware z veřejné malware databáze malware-domainlist.com. Název souboru byl contract_info.doc. U malware je časté, že se vyskytuje v různých podobných verzích a s jiným jménem souboru. V tomto případě to jsou například: contract_jtennant.doc, contract_alexander.forst.doc, contract_office.doc, contract_dan.doc, contract_meyer.doc, contract_bing.doc, contract_michael.doc, contract_rferris.doc aj.[28] Proto je důležité soubory porovnávat na základě kontrolní sumy.

- MD5: 22bc5e8549c99160fc784b5852f73208
- SHA1: 377a80e3f4b630744a7117c43aef37a71df34f56

Kontrolní suma byla vygenerována pomocí Microsoft File Checksum Integrity Verifier. Virům se typicky přiděluje tzv. obálka, aby se jednodušeji rozlišilo, jak se virus chová a jakým typem souboru se šíří. V případě testovaného viru se jedná o W97M.Dropper jeho vzor byl poprvé objeven 16. ledna 2016,[29] samotný vir vznikl 18. ledna 2017.

Soubor je typu MS Word dokumentu, který nenese v příponě označení makra. Soubor má 196 kB, 2 stránky, 0 slov, 5 znaků, 1 řádek, 1 odstavec. Jeho autorem je Gabriel, byla použita šablona Normal.dot a kontrola gramatiky je zapnutá pro ruský jazyk. Při otevření souboru se uživateli zobrazí výzva k povolení makra. Pokud by tak neučinil, škodlivý kód se neprovede. Obsahem dokumentu, který je chráněn proti úpravám heslem, je obrázek s logem MS Office a návodem. Dále je zde napsáno v anglickém jazyce, že je soubor chráněn. A aby nebyl neznalý uživatel ochuzen o spuštění škodlivého kódu, je zde jednoduchý návod

ve třech krocích, který nabádá ke stažení dokumentu, který může být prohlížen online, povolení úprav a povolení maker. Popis se zablokováním se hájí tím, že chráněné dokumenty nelze prohlížet online. Na druhé stránce dokumentu se již nic nenachází. Podstatná část se nachází ve zdrojovém kódu ve Visual Basic.

4.2.2 Rozbor kódu

Kód je napsán ve Visual Basic for Application a nese známky obfuskace. V rámci deklarace systémových knihoven je v komentáři píseň Look At Me Now. Názvy procedur, funkcí a proměnných nedávají smysl a je velmi náročné se v dokumentu orientovat. Názvy proměnných jsou však reálná slova z oblasti biologie.

Zdrojový kód 6: Vybraná část původního kódu makra

```
sparingly = puddler(VarPtr(numbly), VarPtr(rectus) + 8,
discreet)
manichord = -1
aplasia = 56 + 91 - 54 - 93
circumscribed = 0
chimney = 9495
exemplification = 115 + 64 + 24 + 3893
deontology = 124 - 60
parcere = agape(ByVal manichord, aplasia,
ByVal circumscribed, chimney, ByVal exemplification,
ByVal deontology)
dame = Round(473.1269 + 345.1064)
finch = finch Or 242

puddler aplasia, numbly, 116 + 124 + 5354
intercede = 44
mythic = 15880
empress = 397897
foramen = SLN(empress, mythic, intercede)

storing = aplasia
End Function
Sub auriga()
Dim catalase As Byte
Dim extraordinariness As Byte
midil = ThisDocument.ComputeStatistics(wdStatisticPages)
```

Ve zdrojovém kódu 1 lze vidět například funkci SLN - pokles hodnoty aktiva za období nebo funkci ComputeStatistics, která vrací počet stránek. Tyto funkce jsou v kódu navíc a nehrají žádnou roli. Funkce SLN je dokonce v kódu několikrát. Kód se nachází v jednom modulu a jednom Microsoft Word Object. Po odstranění obfuskace je zde 6 funkcí, které obstarávají tvorbu a spuštění malware. Jedna z funkcí, která byla zpočátku skrytá, zajišťuje, aby se ihned po spuštění dokumentu spustila hlavní funkce s vykonáváním nechtěných instrukcí.

Hlavní funkce obstarává komunikaci s ostatními funkcemi a zajišťuje postupnou tvorbu malware. Škodlivý kód se totiž nenachází přímo v jazyce Visual Basic for Application (VBA), ale je na první pohled skrytý. Nachází se ve formuláři, který se jmenuje bop a obsahuje 15 záložek. Funkce prochází veškeré karty a na jedenácté kartě si načte název této karty, ve kterém je 7460 znaků dlouhý řetězec se zakódovanými instrukcemi. Tento řetězec neobsahuje pro člověka čitelné informace. Část řetězce se zakódovanými instrukcemi:

```
e;4Xl#"Z<U5BYD;\*;;"dAi)X2;"5glhLUia75)
```

Zde je důležité zmínit, že řetězec obsahuje znaky jako lomítko, středník a jiné symboly. V momentě importu řetězce do jiného vývojového prostředí je ohlášena chyba právě kvůli nevalidního ukončení řetězce a nelze proto pokračovat v provádění dalších funkcí. Analýza tedy musela být nutně prováděna v původním prostředí.

Aby bylo možné s řetězcem dále pracovat na úrovni operačního systému, jsou jeho znaky převedeny pomocí funkce StrConv z kódování Unicode do výchozího kódování operačního systému. Dále se pak k hodnotám přičítají podle podmínky, jestli jsou sudé či liché čísla 14 nebo 15, mocniny 2, jsou aplikovány operátory celočíselné dělení a AND. V této části jsou binární data s instrukcemi hotova.

V další části kódu se zapíše do paměti ukazatel na vytvořený řetězec pomocí funkce NtWriteVirtualMemory ze systémové knihovny ntdll.dll, alokuje se paměť funkcí NtAllocateVirtualMemory opět z ntdll.dll a zapíše se do alokované paměti funkcí NtWriteVirtualMemory řetězec s instrukcemi. Odkaz na alokovanou paměť je předán hlavní funkci. V hlavní funkci se již jen vytvoří nové vlákno pomocí funkce SHCreateThread z knihovny shlwapi.dll a vláknu se předá odkaz na paměť, ve které jsou další instrukce k vykonávání škodlivého kódu.

Zdrojový kód 7: Ukázka kódu po odstranění obfuskace

```
1 Function funcWillWriteVirtualMemory(NumOfEl, AccessWid, _
2 WriteBuffer)
3 #If Win64 Then
4     Dim Addr As LongPtr
5     Dim NumOfElem As LongPtr
6     Dim ElemWritten As LongPtr
7     Dim AccessWidth As LongPtr
```

```
8       Dim WriteBuf As LongPtr
9   #Else
10      Dim NumOfElem As Long
11      Dim Addr As Long
12      Dim AccessWidth As Long
13      Dim ElemWritten As Long
14      Dim WriteBuf As Long
15      Dim ptrBSTR As Long
16  #End If
17
18      NumOfElem = NumOfEl
19      WriteBuf = WriteBuffer
20      AccessWidth = AccessWid
21
22      Addr = -1
23
24      WriteVirtualMemory ByVal Addr, NumOfElem, _
25      AccessWidth, WriteBuf, ElemWritten
26
27  End Function
```

Zdrojový kód 2 je čitelnější, názvy proměnných reflektují svoji funkci a je zbaven přebytečných výpočtů. V kódu je patrné, že programátor počítá se spuštěním také v Office 64bitové verze i přes její nepřízeň ze strany IT profesionálů a vývojářů.[30]

4.2.3 Komunikace po síti

Po vytvoření vlákna se spustí proces svchost.exe, který obstarává běh malware. Co se týče komunikace po síti, proces nejprve pošle požadavek na DNS server. Je to požadavek na A záznam na adresu api.ipify.org. Vzápětí dojde odpověď, že api.ipify.org je type CNAME pro api.ipify.org.herokudns.com a IP adresa serveru 54.235.212.238. Poté proběhne navázání spojení díky three-way handshake a je odeslán HTTP GET požadavek. Proces zjišťuje, jaká je jeho veřejná IP adresa. Odpověď přichází protokolem HTTP v plain textu. Jakmile proces zná svou IP adresu a pomocí systémových knihoven a registrů zjistí další informace o hostitelském systému, spojí se se serverem 109.120.170.116 a pomocí metody POST odešle data do formuláře na adrese http://ningherthadpa.ru/ls5/forum.php.

Podoba dat, která jsou odeslána do formuláře:

```
Form item: "GUID" = "12702342871922442248"
```

```
Form item: "BUILD" = "1801b"
Form item: "INFO" = "PC-TEST @ PC-Test\Uživatel"
Form item: "IP" = "158.194.129.201"
Form item: "TYPE" = "1"
Form item: "WIN" = "6.1(x64)"
```

Je zřejmé, že si útočník dělá statistiku používaného operačního systému, jeho verze, uživatelů a GUID, což je Globally Unique Identifier nebo také UUID - Universally Unique Identifier. Tento identifikátor je standardem a slouží k jednotnému označování zdrojů. Identifikátor má délku 128 bitů, což zaručuje jedinečnost napříč vesmírem a časem.[31] Jako odpověď na odeslanou identitu přichází požadavek o navázání spojení a pakety s částmi stránky. Pokud uživatel simuluje odesílání požadavku s daty do formuláře, dostane se na stránku БизнесРОСТ, která však zobrazí pouze Ошибка 404. Chybová zpráva číslo 404 znamená - chyba způsobená klientem, daná stránka neexistuje. Celá komunikace se stále opakuje a to beze změn. V rámci příprav práce a samotnou analýzou se adresa pro odeslání formuláře změnila na brost.kz. Klientská stanice o tom byla informována HTTP chybovou zprávou číslo 301 - Moved Permanently, což znamená, že stránka byla navždy přesunuta jinam. Jedná se o URL přesměrování. Pomocí příkazu DIG na systému Linux lze zjistit, že doména ningherthadpa.ru má svůj jmenný server na dollardns.net, tudíž mohlo jít skutečně jen o dočasnou doménu po dobu životního cyklu viru. V rámci běhu procesu lze také na síti odchytit pakety, které využívají NetBIOS Name Service a broadcastem se dotazují na doménu lotihecter.com. Odpověď na tento požadavek proces nedostal.

4.2.4 Chování procesu

Vytvoření procesu iniciovala aplikace WINWORD.EXE. Visual Basic, přestože se zobrazuje jako samostatné okno, není samostatným procesem, proces s malware spouští Word. V následujících odstavcích budou popsány registry a knihovny, které jsou načteny během běhu malware a s nimiž malware komunikoval. Analýza knihoven a registrů proběhla pomocí Process Monitor.

Nejprve dojde k načtení hodnot ze systémových registrů. Načtou se tiskárny, aktuální verze operačního systému z registru CurrentVersion, načtení hodnoty, kde se nachází instalace systému, dochází k načtení knihovny WsmSVC.dll z WinRM Client Shell API, WSMAuto.dll - The Automation layer that provides scripting support - knihovna zajišťující podporu HTTP a HTTPS transportu, součást Windows Remote Management Architecture, a systémové knihovny ntdll.dll.[32]

Opět probíhá načtení hodnot z registrů - Image File Execution Option, Session Manager. Dále pak knihovny z WOW64, které poskytují rozhraní mezi 32bitovou verzí ntdll.dll a jádrem operačního systému a zachycují systémová volání, systémové knihovny kernel32.dll a kernelbase.dll, psapi.dll - Process Status API, což je knihovna, která pomáhá získat informace o procesech a ovladačích zařízení,[32]

v registru jsou následně zjištěny cesty k uživatelovým adresářům jeho profilu. Jedná se o složky Stažené soubory, Dokumenty, Oblíbené položky a Plocha.

Po zjištění cesty k adresářům se zjišťuje místní jazykové nastavení v registrech Multilingual User Interface (MUI) a v registrech Locale, probíhá načtení hodnot z registru při nouzovém režimu, načítá se knihovna msvcrt.dll - běhová knihovna jazyka C, sechost.dll - Event Tracing Functions, knihovna rpcrt4.dll - Developing RPC Windows Applications, knihovna sspicli.dll - obsahuje ověřovací funkce, knihovna cryptbase.dll - knihovna používaná při prolamování oprávnění, knihovna advapi32.dll - zajišťuje přihlášení uživatele do systému, knihovna wininet.dll - WinInet API, knihovna shlwapi.dll poskytující WinInet API, knihovna gdi32.dll - Graphic Device Interface, knihovna user32.dll pro zasílání zpráv, knihovna lpk.dll - Language Pack, knihovna usp10.dll - Uniscribe Unicode script processor, knihovna urlmon.dll - sloužící jako Multipurpose Internet Mail Extensions MIME handler a Component Object Model (COM) aplikace pro WinInet knihovnu.[32]

Dále pak knihovna ole32.dll - jedná se o knihovnu pro podporu Object Linking and Embedding (OLE), tato technologie umožňuje sdílet data mezi aplikacemi podporované MS Windows, knihovna oleaut32.dll - OLE Automation, knihovna crypt32.dll implementující certifikáty a šifrování zpráv, knihovna msasn1.dll zajišťující běhové prostředí pro ASN.1 API, knihovna iertutil.dll užívaná pro navázání na Internet Explorer k exportu tabulky, knihovna imm32.dll - pro Windows Input Method Manager (IMM) API klienta, knihovna msctf.dll - MS Text Service Module.[32]

Dále se pracuje s registry Compatibility, LoadAppInit_DLLs, GRE_ Initialize, OLE, Services v nichž je crypt32, Internet Settings, zde proběhne přečtení všech jmen 24 podklíčů. V další části proces opět načítá knihovny, protože bude komunikovat po síti.

Je načtena knihovna iphlpapi.dll zodpovědná za přenos dat přes TCP/IP, knihovna nsi.dll - NetWare System Interface, knihovna winnsi.dll, knihovna dhcpcsvc6.dll - DHCPv6 Client, ws_32.dll - Windows Socket 2.0, knihovna k vytvoření spojení ke specifickému soketu.[32]

Dále je načteno z registru jméno počítače. Jedná se o registr ComputerName. Jsou také načteny informace týkající se instalace systému. Pak se pracuje s knihovnou comctl32.dll - Common Control library popisující, jak identifikovat verzi, kterou aplikace využívá a vysvětluje zacílení aplikace. Následujícími načtenými registry jsou Internet Settings a Internet Explorer a jeho FeatureControl klíče, Cache a její klíče. Následuje knihovna shell32.dll, která obsahuje funkce pro Windows Shell, knihovna profapi.dll, která značí User Profile Basic API, další operací je načtení SQMClient a WinSock2.[32]

Windows Sockets 2 dovoluje vytvořit pokročilé aplikace, které mohou komunikovat po síti, přenášet data nezávisle na tom, jaký byl použit síťový protokol. Aby mohl být Winsock použit, jsou mu nastaveny registry. Jelikož jsou v kódu použity pro komunikaci doménová jména, je potřeba načíst knihovnu dnsapi.dll, což je DNS Client API, knihovna zodpovědná za komunikaci s DNS servery, jenž

vytváří DNS query.[32]

V tomto úseku probíhá vytvoření nového vlákna a načtení knihovny rasapi32.dll - Remote Access Service API. Jsou zde funkce k vytvoření vytáčeného spojení, správě telefonního adresáře aj. Dále pak knihovna rasman.dll zodpovědná za bezpečné ověření vzdáleného uživatele, knihovna rtutils.dll - Remote RAS tracing, knihovna sensapi.dll pro zjištění stavu sítě a napájení, pomocná knihovna WinSock nlaapi.dll - Network Location Awareness, knihovna rpcss.dll poskytující infrastrukturu pro COM a zároveň součást Remote Procedure Call (RPC), knihovna mswsock.dll - MS Windows Socket pomocná knihovna WinSock, knihovna winrnr.dll - Microsoft LDAP RnR Provider, knihovna wshtcpip.dll, která pro WinSock zajišťuje socket-level komunikaci přes TCP/IP, knihovna fwpulnt.dll - Windows Filtering Platform (WFP), knihovna netprofm.dll, která sbírá informace o síti, k níž je počítač připojen a o případných změnách informuje aplikace, knihovna s kryptografickými funkcemi cryptsp.dll, knihovna rsaenh.dll - Enhanced Cryptographic Provider,[32] rpcrtremote.dll, jedna z knihoven, která může být použita k obejití Windows User Account Control (UAC) u Windows 7.[33]

V dalším kroku je navázáno spojení z napadeného počítače na vzdálený server, odeslány data a přijato potvrzení .

```
PC-Text:49697 -> 109.120.170.116:http
```

Nyní je načtena knihovna ntmarta.dll - NT Multiple Access RouTing Authority, která zajišťuje bezpečnost Win32 API, oprávnění různých objektů jako soubory, klíče a služby.[32] Následuje komunikace ze strany serveru. Je navázáno spojení a přichází několik paketů s daty, ve kterých je chybová stránka. Následuje odpojení od serveru.V tuto chvíli má malware veškeré potřebné informace, načtené knihovny a jednou odeslal získané informace o napadeném PC. U tohoto kroku malware svou činnost nekončí. Je ukončeno vlákno, ve kterém jsou poprvé odeslána data. Následně je vytvořeno vlákno další a data opět odeslána na cílový server.

Po ukončení aplikace MS Office Word zůstává proces svchost.exe i nadále spuštěný na pozadí a vykonává svou práci. Proces je možné ukončit ve správci úloh. V případě analýzy přes Sandbox je pohodlnější zvolit jediný proces svchost.exe a nesnažit se ukončovat systémové procesy.

4.2.5 Zpětný překlad ze strojového kódu

V makru dokumentu se hned v první části získal řetězec z formuláře, který byl součástí makra a poté proběhlo dešifrování z nic neříkajících znaků na přesné instrukce. Pro tyto instrukce byla alokována paměť, která byla předána novému procesu. Jedna z možností, jak zachytit instrukce v řetězci je vytvořit si vlastní knihovnu, simulovat činnost ntdll.dll a zároveň získat data. V knihovně se nachází funkce NtWriteVirtualMemory, která byla volána a pomocí které byl řetězec zapsán do paměti. V rámci vytváření knihovny pro systém Windows bylo

čeleno komplikacím s registrací knihovny do registrů, aby ji bylo možné použít ve vývojovém prostředí Visual Basic. Další možností je vytvořit si vlastní funkci SHCreateThread a zkoumat data přes debugger v rámci vlákna. Poslední a použitou možností je, upravit si kód tak, aby se řetězec uložil do binárního souboru.

Soubor je potřeba otevřít v nástroji pro zpětný překlad ze strojového kódu. Výsledek překladu souboru je možné nalézt v příloze A na optickém disku ve složce malware v souboru OUTPUT.TXT.

4.2.6 Popis spuštěného procesu

Spuštěný proces svchost.exe nemá velké nároky na výkon, zabírá 2876 kB v operační paměti. V rámci jednoho odeslání dat potřebuje proces 20 kB dat na samotné odeslání dat a režie s tím spojené.

Server 109.120.170.116 běží v ruském cloudu společnosti infobox, která poskytuje své služby od roku 2002.[34] Zároveň tato společnost má pronajatý blok adres 109.120.170.0 - 109.120.170.255.[35]

4.2.7 Důvod vytvoření

Stránky brost.kz nabízejí v rámci svého eshopu prostředky a taky služby pro podnikání. Jedná se o razítka, pokladní systémy, registrace firmy, právnické služby a rovněž služby programátora. Je možné, že je mapován trh a následně je vytvářena nabídka služeb. Pravděpodobnějším důvodem bude průzkum obětí a následné vytváření malware na míru. Díky WHOIS lze nalézt registrátora domén, který má smyšlené jméno, americký původ a odkazy na další jeho domény vedoucí do Ruska.[36]

4.3 Postup při analýze

Před samotnou analýzou je zapotřebí mít nainstalován veškerý vypsaný software, aby bylo možné dále pokračovat podle postupu nebo mít staženo předem připravené prostředí ve formě exportovaného virtuálního počítače, který byl optimalizován pro svůj běh v hypervizoru VirtualBox a je součástí přílohy B.

4.3.1 Síťové pakety

K zachytávání poslaných paketů jsme spustili program WireShark. Po spuštění WireSharku, jsme byli vyzváni, abychom vybrali síťový adaptér. Následně probíhá zachytávání paketů a jejich výpis na obrazovku. Záleží na velikosti dané sítě, ale ve větších sítích jsme okamžitě nastavili filtrování paketů, aby se nezobrazovaly pakety, které se analýzy netýkají.

```
ip.src_host || ip.dst_host
```

4.3.2 Systémová volání

Podobně jako pro síťové pakety, jsme spustili Process Monitor na zachytávání systémových volání. Ihned po spuštění jsme byli vyzváni, abychom nastavili filtrování. Jsou zde dvě hlavní možnosti, buď zahrnout nebo vyloučit záznamy a následně jsme zvolili další parametry, podle kterých probíhá filtrace. V tomto programu je zapotřebí na začátku pouze spustit zachytávání. Poté, co je dokončeno makro ve Visual Basicu, vzniká nový proces, podle jehož Process Identifier (PID) jsme filtrovali záznamy.

4.3.3 Sandbox

Na závěr jsme spustili sandbox. Díky sandboxu jsme mohli zajistit, aby proces nepřistupoval ke zdrojům, na které nemá nárok. Což jsme zajistili nastavením nového sandboxu. V první chvíli jsme nastavili velikost paměti na disku, kterou může používat a adresáře, které jsme monitorovali. Povolené aplikace jsme jednoduše nastavili až za běhu programu, protože při snaze spustit nepovolený program, se nám zobrazila zpráva a měli jsme možnost dvojklikem běh tohoto programu povolit. V danou chvíli zbývá spustit MS Office Word a otevřít testovaný soubor. Otevření Wordu jsme provedli otevřením kontextové nabídky v sandboxu a následně výběrem aplikace nebo zvolením "Run any program"a zadáním cesty souboru.

4.3.4 Visual Basic

Makra jsou ve výchozím nastavení každé aplikace MS Office z bezpečnostních důvodů zakázána. V případě analýzy jsme museli povolit jejich běh. Následně jsme přepnuli do Visual Basic přes kartu Vývojář. Obecně existuje klávesová kombinace Alt+F11, která nám taktéž otevřela Visual Basic.

V levém horním okraji je zobrazena struktura projektu. Nás zajímá "ThisDocument"v Microsoft Word Objects a modul "bryopsida". Dále je zde formulář "bop", ze kterého se budou získávat instrukce pro vlákno.

V originální verzi vzorku byly okamžitě po povolení makra spuštěny instrukce k vytvoření viru, ale v upravené verzi bylo potřeba spustit proceduru "funcStart". Makro jsme spustili, postupně krokovali a sledovali jednotlivé proměnné. Zaměřili jsme se na jednotlivé systémové instrukce a hodnoty s jakými byly volány. Zjistili jsme, že se v kódu získával název formuláře, ze kterého byly postupně vytvořeny instrukce, jenž byly jako parametr předány při alokaci paměti. Tato paměť byla opět jako parametr předána při vytvoření nového vlákna.

Získání instrukcí předaných při alokaci paměti jsme provedli uložením binárního pole do souboru a jeho následným zpětným překladem do strojového kódu programem Hexplorer. Přeložený soubor je součástí přílohy A na DVD.

4.3.5 Kód makra

Kód makra pro tvorbu vlákna viru nesl známky obfuskace, proto jsme jednotlivé proměnné přejmenovali tak, aby název odpovídal účelu použití. Totéž jsme provedli s funkcemi a procedurami. V kódu se vyskytovaly funkce ComputeStatistics a SLN. SLN se nacházela hned několikrát. Postupným krokováním programu a přejmenováním proměnných jsme dospěli k úsudku, že tyto funkce jsou v kódu nadbytečné a jsou pouze jednou z obfuskačních technik. Další použitou technikou obfuskace bylo vložení písně do komentáře a zrušení formátování kódu. Komentáře jsme proto smazali a formátování upravili tak, aby byl kód přehlednější. Vlastní komentáře jsme v textu použili před přejmenováním proměnných. Volání matematických a logických funkcí použitých jako argument při volání druhé funkce, která je aplikovala, jsme přepsali do procedury, která je před modifikací volala. Dalšími obfuskačními technikami byly vkládání mrtvého kódu, kódování dat a jejich složení za běhu programu. Mrtvý kód jsme smazali a dekódování dat s následným složením jsme ponechali původní s tím rozdílem, že některé číselné vstupní argumenty byly zadány ve formě sčítanců nebo menšence a menšitele, ty jsme přepsali jako součet nebo rozdíl.[38]

4.3.6 Systémový proces svchost.exe

Vir běžel jako systémový proces svchost.exe. Jeho PID jsme sledovali v sandboxu nebo pomocí správce úloh, protože procesy spuštěné v sandboxu běžely pod jiným uživatelským účtem. Domníváme se, že pokud by běžný uživatel spustil takto upravený dokument, nejspíš si nevšimne jednoho spuštěného procesu svchost.exe navíc. Proces se tvářil jako systémový, ale na rozdíl od těch skutečně systémových byl spuštěn pod uživatelským účtem daného člověka, tudíž jsme jej snadno identifikovali.

4.3.7 WinDbg

Chování procesu jsme zkoumali také pomocí aplikace WinDbg. V nabídce programu jsme vybrali "Attach To a Process"nebo stiskli klávesovou zkratku F6. Následně vybrali proces svchost.exe se správným identifikátorem. Poté jsme zkoumali registry, paměť, vlákna, volání zásobníku aj. Výsledky jsme nakonec nepoužili pro samotnou analýzu.

4.3.8 Samotná analýza výsledných dat

Následovalo analyzování všech zachycených dat včetně hlášek, které jsme obdrželi při běhu sandboxu. Prošli jsme všechna systémová volání, zkontrolovali knihovny, které byly načteny, zjistili s čím souvisí a proč byly volány, totéž platí pro systémové registry. V rámci síťové komunikace jsme prošli všechny záznamy a jejich podrobnosti, které se zobrazují v dolní části programu Wireshark. V době zachytávání paketů jsme neměli spuštěné na pozadí žádné další programy komunikující po síti, abychom dostali jen pakety související s virem.

Závěr

Na začátku práce jsme se seznámili s pojmem slova malware a jeho vybranými typy, jako jsou virus, počítačový červ, spyware, trojský kůň, ransomware aj.

V následující kapitole jsme řešili, jaké jsou možnosti sběru vzorků malware. Podrobněji jsme rozebrali honeypot a jeho možnosti, protože realizace honeypotu byla jedním z praktických bodů práce.

Před samotným rozborem jsme popsali nástroje pro analýzu vybraného malware a vytvořili testovací prostředí. Každý nástroj jsme popsali, vysvětlili jeho hlavní funkce a vždy uvedli přesnou verzi, kterou jsme použili.

Následně jsme podrobně charakterizovali zkoumaný malware, kterým byl virus W97M.Dropper. Virus pocházel z veřejné internetové databáze a jednalo se o makro virus. V této kapitole jsme uvedli postup samotné analýzy. Způsob zachycení paketů a systémových volání, na co se jsme zaměřili v procházení kódu makra a popisu chování viru, který se skrýval za systémové procesy.

V rámci analýzy jsme měli vytvořit vlastní knihovnu, díky které bychom získali data z paměti potřebná pro zpětný překlad do strojového kódu a další analýzu chování. V této části nám chyběly potřebné informace pro implementaci a proto zůstává předmětem dalšího výzkumu.

Výsledkem práce je zjištění, že virus za pomoci systémových knihoven sbíral data o uživateli a počítači ze záznamů z registrů. Zjišťoval identifikátor a jméno počítače, verzi operačního systému, uživatelské jméno a z webového API api.ipify.org obdržel veřejnou IP adresu. Tyto informace pak odeslal pomocí HTTP POST metody na ruský server. Domníváme se, že cíl, jenž jsme si v zadání práce stanovili jsme splnili.

Conclusions

At the beginning, we learned about the word malware and its selected ones types such as virus, computer worm, spyware, trojan, ransomware etc.

In the next chapter, we have discussed how to collect malware samples. We have described the honeypot and its versions, because the realization of honeypot was one of the practical points of the thesis. Before analyzing, we have described tools for analyzing selected malware and created the testing environment. We described each tool, explained his main features and always mentioned the exact version we used.

Then we described the malware sample, which was the W97M.Dropper virus. The virus came from a public Internet database and it was a macro virus. In that chapter we presented the analysis itself. How to capture a packet and system calling, what we have focused during debugging the macro code and described the behavior of the virus that hid behind the system processes.

In the analysis we had to create our own library, which would get memory data needed for disassembly and further behavioral analysis. In this part we did not have the necessary information for implementation and therefore remains the subject further research.

The result of this work is finding out that the virus in aid of system libraries collected data about the user and the host from the record from the registry. Identified a host name, operating system version, username, and from the web API api.ipify.org has received its public IP address. This information was sent using HTTP POST methods on a Russian server. We believe that the goal we have determined in the assignment of the work we met up.

A Obsah přiloženého DVD

Struktura přiloženého DVD se skládá ze 2 složek, ve kterých je uložena práce a zkoumaný vzorek:

doc/
: Ve složce se nachází soubor s vygenerovanou prací ve formátu PDF. Složka dále obsahuje potřebné soubory k vytvoření daného souboru.

malware/
: Obsahuje zkoumaný vzorek malware ve verzi, v jaké byl obdržen k analýze. Z bezpečnostních důvodů je zkomprimovaný. Dále se ve složce nachází soubor OUTPUT.TXT v němž je zpětný překlad binárního souboru ze strojového kódu pro tvorbu maligního vlákna.

U veškerých cizích převzatých materiálů obsažených na DVD jejich zahrnutí dovolují podmínky pro jejich šíření, nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce.

B Exportovaný obraz virtuálního počítače

V cloudu je umístěn export virtuálního počítače, který byl používán při práci. Počítač má nainstalován veškerý software potřebný k analýze. Zkoumaný vir je uložen na ploše. Pro přihlášení k uživatelskému účtu ve Windows je potřeba zadat heslo 12345. Adresa pro stažení http://goo.gl/kvTtbI

Bibliografie

[1] What is malware? MIT [online]. [cit. 2017-05-23]. Dostupné z: https://ist.mit.edu/security/malware

[2] LANDESMAN, M. A Brief History of Malware. LifeWire [online]. 2014. [cit. 2017-04-21]. Dostupné z: https://www.lifewire.com/brief-history-of-malware-153616

[3] BREWSTER, T. The evolution of botnet. IT PRO [online]. 2010. [cit. 2017-07-15]. Dostupné z: http://www.itpro.co.uk/627487/the-evolution-of-the-botnet

[4] ROUSH, W. Inside the Spyware Scandal. MIT Technology Review [online]. 2006. [cit. 2017-04-21]. Dostupné z: https://www.technologyreview.com/s/405741/inside-the-spyware-scandal/

[5] BARWISE, M. Quantity of malware booms. The H Security [online]. 2008. [cit. 2017-04-21]. Dostupné z: http://www.h-online.com/security/news/item/Quantity-of-malware-booms-735811.html

[6] KARYOTIS, V.; KHOUZANI, M.H.R. Malware Diffusion Models for Modern Complex Networks: Theory and Applications. Morgan Kaufmann [online]. 2016. [cit. 2017-04-21]. Dostupné z: https://books.google.cz/books?id=uLLsBAAAQBAJ

[7] COHEN, F.: Computer Viruses - Theory and Experiments. [online]. 1984. [cit. 2017-07-26]. Dostupné z: http://all.net/books/virus/index.html

[8] Introduction to Viruses. Computer Knowledge [online]. 2013. [cit. 2017-07-27]. Dostupné z: http://www.cknow.com/cms/vtutor/introduction-to-viruses.html

[9] What is a Keylogger? Kaspersky lab [online]. [cit. 2017-05-04]. Dostupné z: https://usa.kaspersky.com/resource-center/definitions/keylogger

[10] KREBS, B. Hacked Cameras, DVRs Powered Today's Massive Internet Outage. Krebs on Security [online]. 2016. [cit. 2017-05-05]. Dostupné z: https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/

[11] KREBS, B. Hacked Cameras, DVRs Powered Today's Massive Internet Outage. Krebs on Security [online]. 2017. [cit. 2017-07-28]. Dostupné z: https://krebsonsecurity.com/2017/03/dahua-hikvision-iot-devices-under-siege/

[12] SUROVIč, M. Static Behavioral Malware Detection over LLVM IR. [online]. Brno, 2016. Diplomová práce. Vysoké učení technické, Fakulta informačních technologií. Ústav inteligentních systémů. Dostupné z: https://dspace.vutbr.cz/bitstream/handle/11012/61919/18603.pdf

[13] GROSS, G. Rootkit Detection: Techniques and Best Practices. Alien Vault [online]. 2016. [cit. 2017-07-28]. Dostupné z: https://www.alienvault.com/blogs/security-essentials/rootkit-detection-techniques-and-best-practices

[14] NERENBERG, D. M. A study of Rootkit Stealth Techniques and Associated Detection Methods. [online]. Ohio, 2007. Diplomová práce. Wright-Patterson Air Force Base Ohio, Air Force Institute of Technology, Air University, Department of the Air Force. Dostupné z: http://www.dtic.mil/dtic/tr/fulltext/u2/a519999.pdf

[15] SQL Injection. Open Web Application Security Project. [online] 2016. [cit. 2017-05-04]. Dostupné z: https://www.owasp.org/index.php/SQL_Injection

[16] SQL Injection. W3 Schools [online]. [cit. 2017-07-27]. Dostupné z: https://www.w3schools.com/sql/sql_injection.asp

[17] Types of SQL Injection (SQLi). Acunetix [online]. [cit. 2017-08-07]. Dostupné z: https://www.acunetix.com/websitesecurity/sql-injection2/

[18] SQL (STRUCTURED QUERY LANGUAGE) INJECTION. Imperva [online]. [cit. 2017-08-07]. Dostupné z: https://www.incapsula.com/web-application-security/sql-injection.html

[19] ZAHARIA, A. What is Ransomware and 15 Easy Steps To Keep Your System Protected. Heimdal Security [online]. 2017. [cit. 2017-05-22]. Dostupné z: https://heimdalsecurity.com/blog/what-is-ransomware-protection/

[20] HÁK, I. Ransomware včera dnes a zítra. AEC a.s. Praha, 2017. Konference Security 2017.

[21] BRANDOM, R. Ransomware victims have paid out more than $25 million, Google study finds. The Verge [online]. 2017. [cit. 2017-07-28]. Dostupné z: https://www.theverge.com/2017/7/25/16023920/ransomware-statistics-locky-cerber-google-research

[22] JOSHI, R.C.; SARDANA, A. Honeypots: A New Paradigm to Information Security. CRC Press [online]. 2014. [cit. 2017-05-06]. Dostupné z: https://books.google.cz/books?id=c_rRBQAAQBAJ

[23] GUDKOVA, D. Spam and phishing in 2016. SECURELIST [online]. 2016. [cit. 2017-05-05]. Dostupné z: https://securelist.com/analysis/kaspersky-security-bulletin/77483/kaspersky-security-bulletin-spam-and-phishing-in-2016/

[24] User Manual. Oracle Corporation [online]. 2017. [cit. 2017-05-05]. Dostupné z: https://www.virtualbox.org/wiki/Documentation

[25] Wireshark Wiki. Wireshark [online]. 2017. [cit. 2017-05-05]. Dostupné z: https://wiki.wireshark.org/

[26] THORNTON, B. Sandboxie 5.18. PC & Tech Authority [online]. 2017. [cit. 2017-05-05]. Dostupné z: http://downloads.pcauthority.com.au/article/2198-sandboxie

[27] RUSSINOVICH, M. Process Monitor v3.33. Windows Sysinternals. [online] 2017. [cit. 2017-05-04]. Dostupné z: https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx

[28] Malware Analysis by Cuckoo sandbox. Malwr [online]. 2017. [cit. 2017-05-05]. Dostupné z: https://malwr.com/analysis/ZjJiZTQ2MDI2ZWMzNDEyMDljNmFlZjllMWIxYzA0OTI/

[29] W97M.Dropper. Symantec [online]. 2016. [cit. 2017-05-15]. Dostupné z: https://www.symantec.com/security_response/

[30] Choose between the 64-bit or 32-bit version of Office. Microsoft [online]. [cit. 2017-05-05]. Dostupné z: https://support.office.com/en-us/article/Choose-between-the-64-bit-or-32-bit-version-of-Office-2dee7807-8f95-4d0c-b5fe-6c6f49b8d261

[31] LEACH, P. et al. A Universally Unique IDentifier (UUID) URN Namespace. Microsoft [online]. 2005. [cit. 2017-05-05]. Dostupné z: http://www.ietf.org/rfc/rfc4122.txt

[32] MSDN Library. Microsoft [online]. [cit. 2017-05-23]. Dostupné z: https://msdn.microsoft.com/en-us/library/ms310241

[33] ANWAR, P. Bypassing Windows User Account Control (UAC) and ways of mitigation. GreyHatHacker.NET [online]. 2014. [cit. 2017-05-02]. Dostupné z: https://www.greyhathacker.net/?p=796

[34] INFOBOX О компании. Infobox [online]. [cit. 2017-05-22]. Dostupné z: https://infobox.ru/about/

[35] Whois IP 109.120.170.116. WHOIS [online]. [cit. 2017-05-22]. Dostupné z: https://www.whois.com/whois/109.120.170.116

[36] lotihecter.com; WHOIS [online]. [cit. 2017-05-28]. Dostupné z: https://www.whois.com/whois/lotihecter.com

[37] NOWAK, T. The Undocumented Functions of NTDLL. NTinterlnals.net Team [online]. 2000. [cit. 2017-05-22]. Dostupné z: https://undocumented.ntinternals.net/

[38] ČERMÁK, M.Obfuskace a základní obfuskační techniky. Clever And Smart [online]. 2016. [cit. 2017-05-23]. Dostupné z: http://www.cleverandsmart.cz/obfuskace-a-zakladni-obfuskacni-techniky/