Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: **Cyberbezpieczeństwo (CBE)**
Specjalność: **Bezpieczeństwo w elektroenergetyce (CEN)**

PRACA DYPLOMOWA INŻYNIERSKA

Analiza złośliwego oprogramowania za pomocą narzędzi informatyki śledczej

Mikołaj Grzempa

Opiekun pracy
dr inż. Robert Czechowski

Słowa kluczowe: analiza złośliwego oprogramowania, informatyka śledcza, środowisko Windows

WROCŁAW, 2023

Streszczenie

W pracy przedstawiono możliwości analizy złośliwego oprogramowania WannaCry za pomocą automatycznych i półautomatycznych narzędzi informatyki śledczej oraz opisane zostały kroki pozwalające skonfigurować bezpieczne środowisko przy pomocy wirtualizacji wraz ze stworzoną izolowaną siecią. Podczas badania użyto narzędzi open-source, które pomagają w analizie statycznej i dynamicznej. Systemy wykorzystane w projekcie to Windows, na którym znajdować się będzie złośliwe oprogramowanie wraz z systemem Linux, który będzie pełnił rolę symulowanego serwera Command & Control. W celu uzyskania pełnego obrazu zachowania złośliwego oprogramowania przy pomocy dekompilacji zmieniony zostanie kod oprogramowania. Wynikiem pracy są znalezione podczas badania indykatory złośliwego oprogramowania, na podstawie których opracowane zostały metody blokowania ataków za pomocą sygnatur YARA. Wnioski przedstawione zostały w podsumowaniu.

Słowa kluczowe: analiza złośliwego oprogramowania, informatyka śledcza, środowisko Windows

Abstract

The thesis presents the possibilities of analyzing the WannaCry malware using automated and semi-automated computer forensics tools, and describes the steps to set up a secure environment using virtualization with an isolated network created. During the study, open-source tools were used to help with static and dynamic analysis. The systems used in the project are Windows, which will host the malware, along with a Linux system, which will act as a simulated Command and Control server. In order to get a complete picture of the malware's behavior using decompilation, the software code will be changed. The results of the work are the malware indicators found during the study, based on which methods were developed to block attacks using YARA signatures. The conclusions are included in the summary.

Keywords: malware Analysis, Computer forensics, Windows environment

Praca została wygenerowana w systemie LaTeX, w wersji LaTeX2e, przy użyciu edytora Overleaf.

Spis treści

Spis rysunków

Spis tabel

Spis listingów

Skróty

GPO (ang. *Group Policy Object*)
MZ (ang. *Mark Zbikowski*)
EDR (ang. *Endpoint Detection and Response*)
C2 (ang. *Command and Control*)
DHCP (ang. *Dynamic Host Configuration Protocol*)
URL (ang. *Uniform Resource Locator*)
API (ang. *Application Programming Interfacee*)
YARA (ang. *Yet Another Ridiculous Acronym*)
HTTP (ang. *Hypertext Transfer Protocol*)
ARP (ang. *Address Resolution Protocol*)
CPU (ang. *Central Processing Unit*)
SSDP (ang. *Simple Service Discovery Protocol*)
IP (ang. *Internet Protool*)
SMB (ang. *Server Message Block*)
RAM (ang. *Random Access Memory*)
RODO (ang. *General Data Protection Regulation*)
DNS (ang. *Domain Name System*)
FPU (ang. *Floating-Point Unit*)
LIFO (ang. *Last In First Out*)
PID (ang. *Process Identification*)
IOC (ang. *Indicator of compromise*)

Wprowadzenie

Cel pracy

Celem pracy jest przygotowanie odizolowanego środowiska, w którym w bezpieczny sposób przeprowadzona będzie analiza złośliwego oprogramowania pod kątem zachowania zaaplikowanego złośliwego kodu. Środowisko to będzie odseparowane od sieci zewnętrznej oraz głównego hosta za pomocą wirtualizacji. Głównym celem pracy jest przedstawienie najważniejszych i najskuteczniejszych narzędzi przeznaczonych do analizy malware-u, przedstawienie technik analizy, wyników porównawczych skuteczności narzędzi oraz technik analitycznych.

Zakres pracy

Zakres pracy obejmuje działania konieczne do poprawnego i efektywnego spełnienia celu pracy. Zadania zostały podzielone na 4 etapy, opisane w 4 kolejnych rozdziałach tej pracy.
Ich zawartość przedstawiono poniżej.

1. **Faza formułowania i analizy problemu:**
   - sformułowanie założeń projektowych i określenie metodologii badawczej,
   - wprowadzenie do złośliwego oprogramowania.
2. **Faza projektowania i implementacji.**
   - praktyczne zapoznanie się z instalacją i konfiguracją podstawowych narzędzi,
   - przygotowanie stanowiska laboratoryjnego (zwirtualizowanego środowiska testowego),
   - zdefiniowanie i aktywowanie złośliwego oprogramowania.
3. **Faza analizy:**
   - przeprowadzenie testów i badań próbki (materiału), również pamięci RAM, wielu systemów operacyjnych (Linux/Windows) i różnych formatów plików hibernacji i snapshotów,
   - zdefiniowanie listy uruchomionych procesów (PID, PPID), aktywnych połączeń, usług, *hash-y*, występowania szkodliwego oprogramowania.
4. **Wnioski i podsumowanie**
   - klasyfikacja znalezionych technik,
   - metody obrony przed złośliwym oprogramowaniem.

Po wykonaniu wszystkich wyżej wymienionych zadań, wyniki pracy zostaną podsumowane. Zostanie również zaprezentowana literatura wykorzystana podczas realizacji pracy.

Rozdział 1

Wstęp teoretyczny

1.1. Sformułowanie problemu

Złośliwe oprogramowanie na przestrzeni lat zyskało znaczną popularność. Na podstawie statystyk z 2022 roku przeprowadzonych przez firmę BlackFrog, można zauważyć wzrost zgłoszonych ataków ransomware o 29% względem roku 2021. Straty związane z konsekwencjami ataku szacuje się na 20 mld dolarów w roku 2021, a zgodnie z przewidywaniami liczba ta ma wzrosnąć nawet do 265 mld w przeciągu 10 lat [24].

Typy złośliwego oprogramowania możemy podzielić przynajmniej na pięć podstawowych kategorii, które zostały wymienione w **tabeli 1.1**.

Tab. 1.1: Wykaz najpopularniejszych kategorii złośliwego oprogramowania [9]

Typ	Przykład	Opis
`Ransomware`	WannaCry	Zadaniem tego oprogramowania jest uniemożliwienie użytkownikowi dostępu do plików i aplikacji, aby wymusić na nim okup.
`Spyware`	Pegasus	Spyware wykorzystywany jest do śledzenia i obserwowania zainfekowanych użytkowników.
`Adware`	Fireball	W sposób niekontrolowany wymusza pokazywanie reklam na ekranie użytkownika.
`Trojany`	Rakhni Trojan	Zadaniem trojanów jest podszywanie się pod znane aplikacje m. in. po to, żeby zmylić programy antywirusowe lub użytkowników.
`Robaki`	Morris Worm	Złośliwe oprogramowanie, mające na celu replikowanie na innych stacjach w tej samej sieci.
`Rootkits`	Uroburos Rootkit	Jego zadaniem jest pobranie informacji o użytkowniku i zdalny dostęp do sesji użytkownika.

Wzrost zainteresowania przestępców przełożył się na zwiększenie zapotrzebowania specjalistów w obszarze cyberbezpieczeństwa oraz konieczność stworzenia odpowiednich narzędzi do zapobiegania atakom. Jednakże, aby skutecznie bronić się przeciwko złośliwemu oprogramowaniu, należy je odpowiednio przeanalizować, zrozumieć oraz wdrożyć potrzebne zabezpieczenia.

1.2. Podstawowe założenia

Podstawowym założeniem projektu jest poddanie szczegółowej analizie wybranego złośliwego oprogramowania za pomocą narzędzi informatyki śledczej w bezpiecznym środowisku. W tym celu podzielono założenia według kategorii:

1. **Część konfiguracyjna środowiska:**
   - przygotowanie wirtualnych maszyn,
   - instalacja potrzebnych narzędzi,
   - odizolowanie sieci domowej,
   - symulacja sieci w obrębie skonfigurowanych maszyn.
2. **Część analityczna:**
   - analiza statyczna,
   - analiza dynamiczna,
   - analiza kodu.
3. **Klasyfikacja zagrożeń i metody obrony:**
   - klasyfikacja za pomocą bazy MITRE ATT&CK,
   - opis znanych narzędzi EDR,
   - utworzenie sygnatur wspomagających narzędzia EDR.

Rozdział 2

Założenia projektowe

Jednym z najbardziej znanych ataków powodujących zainfekowanie około 230.000 komputerów, był atak za pomocą WannaCry [11]. WannaCry to crypto ransomware, którego zadaniem jest szyfrowanie plików ofiar w celu wymuszenia okupu w postaci płatności za pomocą krypto waluty bitcoin. Posłuży on jako przykład do lepszego zrozumienia działania złośliwego oprogramowania. Na podstawie analizy jego zachowania stworzone zostaną skuteczne metody zapobiegania niepożądanej aktywności programu.

2.1. Wirtualizacja

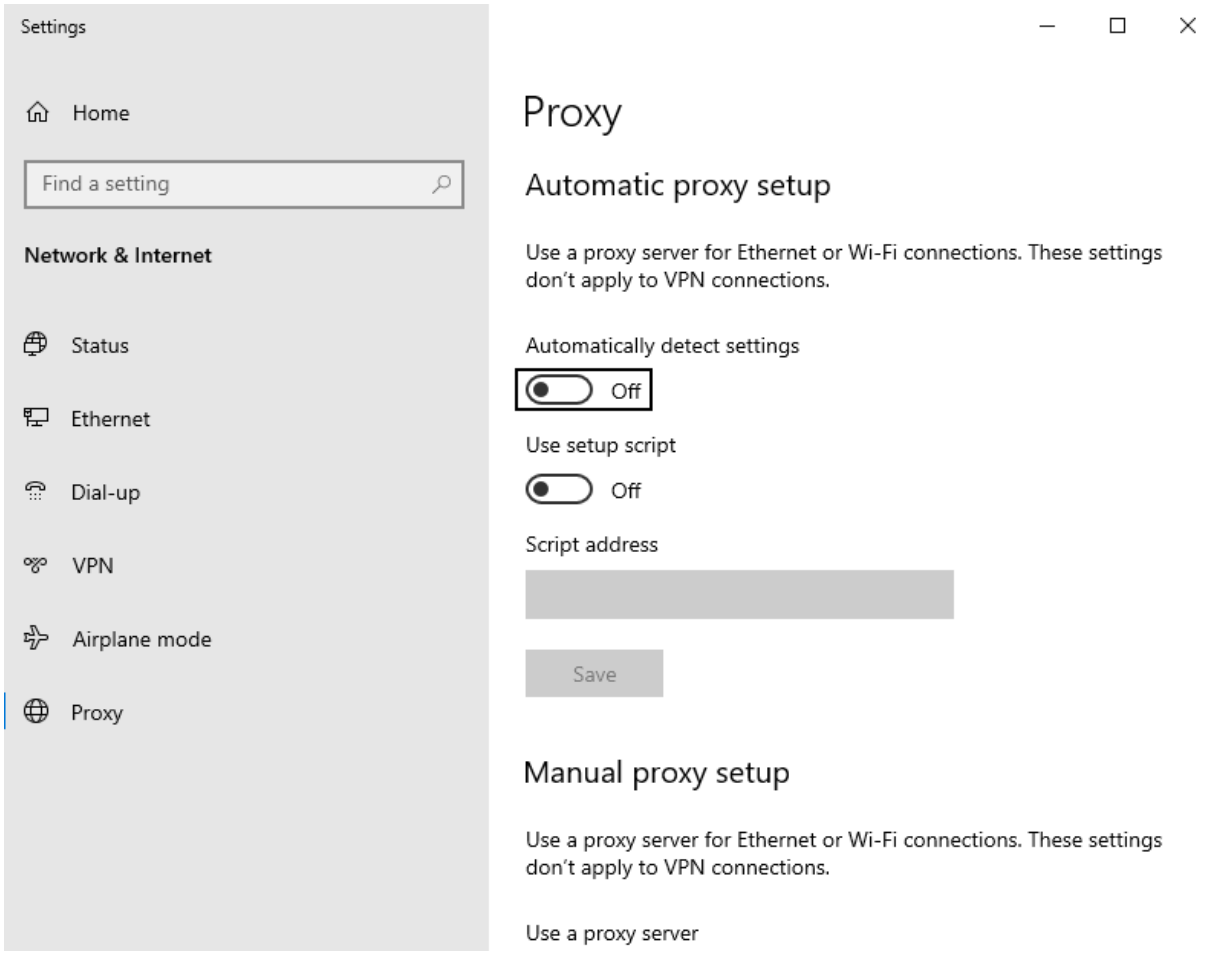
W celu bezpiecznego analizowania złośliwego oprogramowania, należy ocenić ryzyko i konsekwencje, jakie może spowodować nieodpowiednio skonfigurowane środowisko. Jedną z możliwości uniknięcia nieodwracalnych konsekwencji wprowadzenia złośliwego oprogramowania do swojego systemu jest wirtualizacja. Popularne narzędzie wykorzystujące wirtualizację to VirtualBox [28]. Zapewnia ono bezpieczeństwo, a dzięki wykorzystywaniu migawki, czyli zrzutu pamięci aktualnego stanu środowiska, użytkownik jest w stanie cofnąć niekontrolowane zmiany lub jeszcze raz przeprowadzić analizę.

2.2. Środowisko do analizy (Windows)

Ponieważ WannaCry został zaprojektowany dla środowiska Windows, w pracy wykorzystany zostanie obraz ISO dostępny na oficjalnej stronie Microsoft [15] w architekturze 64x. Podczas wyboru środowiska ważne jest, aby był to *Windows enterprise*, pomimo że niektóre z jego udostępnionych wersji wymagają dodatkowych kroków, takich jak instalacja *group policy*, z kolei nie pochodzi z oficjalnych źródeł Microsoft, przez co jest mniej zaufana i może być pozbawiona pewnych rozwiązań potrzebnych w dalszej części analizy. Inną możliwością przeprowadzenia badania było skorzystanie z licencji udostępnionej przez Politechnikę Wrocławską, jednak skutkowałoby to połączeniem prywatnego konta z domeną politechniki, wymuszając na użytkowniku stosowanie praktyk bezpieczeństwa, uniemożliwiając tym jednocześnie pełną analizę złośliwego oprogramowania.

2.2.1. Konfiguracja proxy

Pierwszym krokiem podczas konfigurowania środowiska jest całkowite wyłączenie proxy, działanie to jest niezbędne, by dokonać pełnej analizy ruchu sieciowego. Ustawienia widoczne na **rysunku 2.1**.



Rys. 2.1: Ustawienia proxy dla systemu Windows

2.2.2. Konfiguracja Windows Defender oraz FireWall

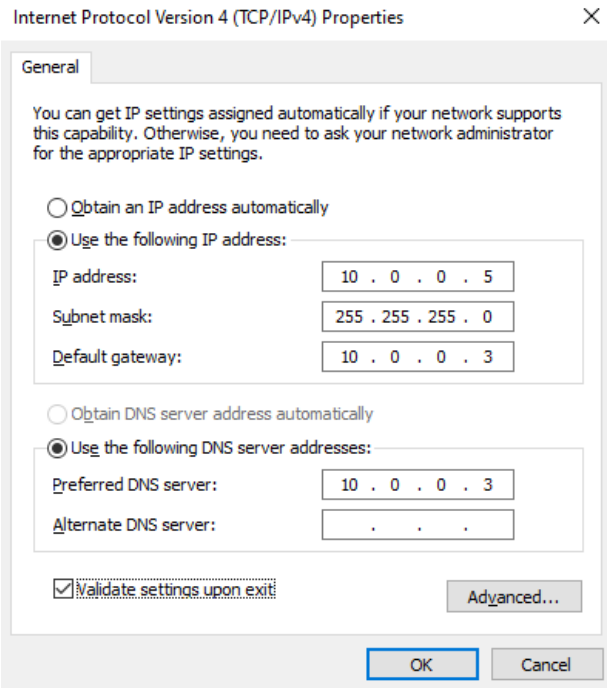
Współczesne antywirusy i *FireWall'e* wykorzystują sygnatury, które mogłoby uniemożliwić pełne zadziałanie złośliwego oprogramowania WannaCry, dlatego podczas konfiguracji środowiska należy je wyłączyć za pomocą *GPO*, czyli zbioru zasad, reguł i ustawień dla systemu, które określają, jak użytkownik lub komputer ma postępować w danej sytuacji [22]. Ścieżka prowadząca do wymaganych opcji:

GPO → Administrative Templates → Windows Components → Microsoft Defender Antivirus → Enable "Turn off Microsoft Defender Antivirus"

GPO → Administrative Templates → Network → Network Connections → Windows Defender Firewall → Domain Profile → Disable "Protect All Network Connections"

2.2.3. Konfiguracja sieci w systemie windows

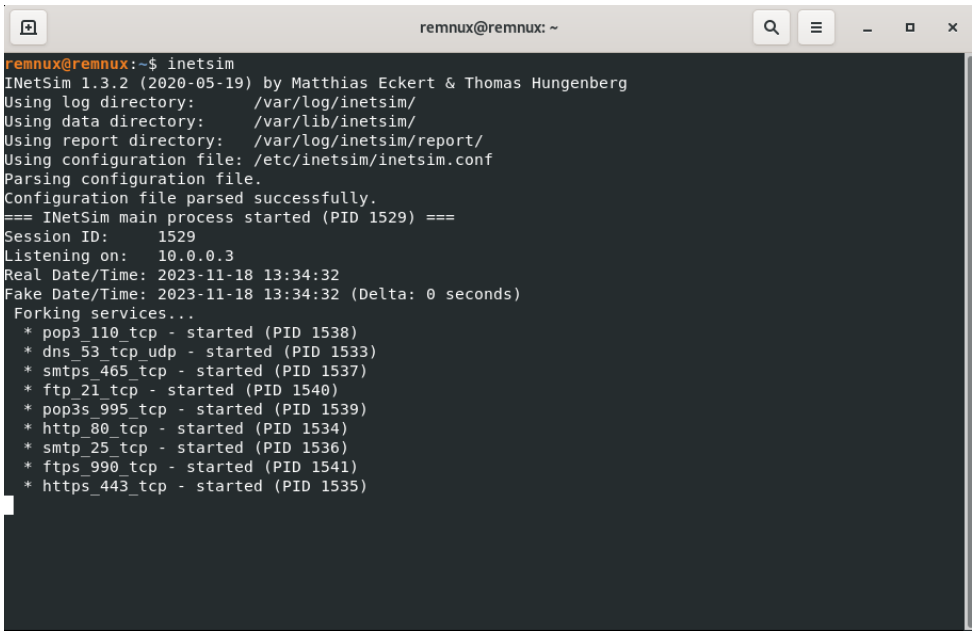
Ostatnim krokiem w konfiguracji środowiska jest statyczne ustawienie sieci IPv4, zgodnie z **rysunkiem 2.2**. Celem tego jest przekierowanie ruchu DNS na skonfigurowany w dalszej części pracy *INetSim* [26].



Rys. 2.2: Konfiguracja DNS dla systemu Windows

2.3. Środowisko symulujące *Command & Control* (Linux)

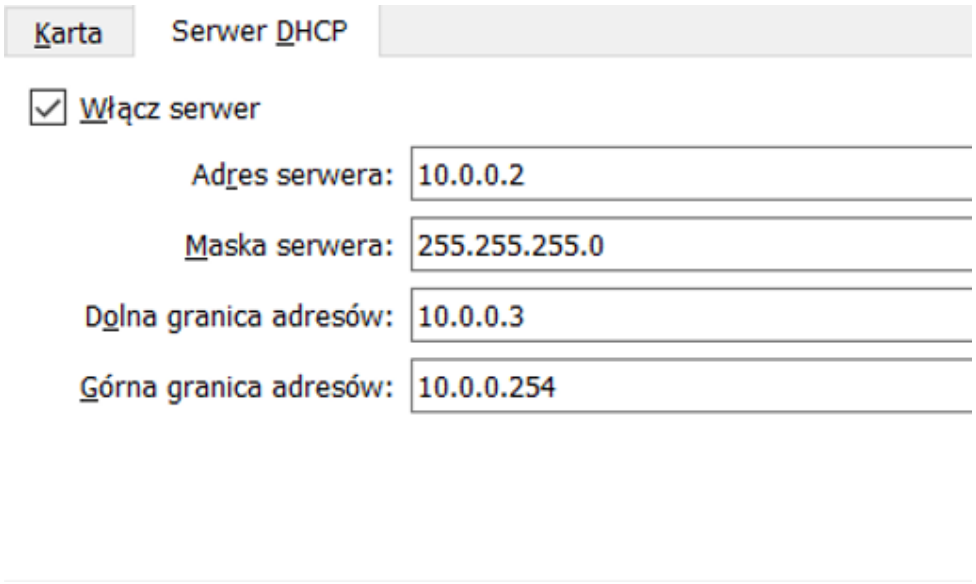
Serwery Command & Control, często określane jako C2, służą cyberprzestępcom do zarządzania komunikacją i kontrolą nad urządzeniami zainfekowanymi przez złośliwe oprogramowanie. W celu symulacji usług potrzebnych do prawidłowego działania złośliwego oprogramowania wykorzystany zostanie system *REMnux* [31] wraz z dostępnym oprogramowaniem symulującym usługi sieciowe *INetSim*. Potrzebne usługi widoczne są na **rysunku 2.3** Umożliwi to oszukanie złośliwego oprogramowania, symulując działający serwer C2 z którym będzie chciał się połączyć.



Rys. 2.3: Usługi wykorzystywane przez *Inetsim* dla środowiska *Remnux*

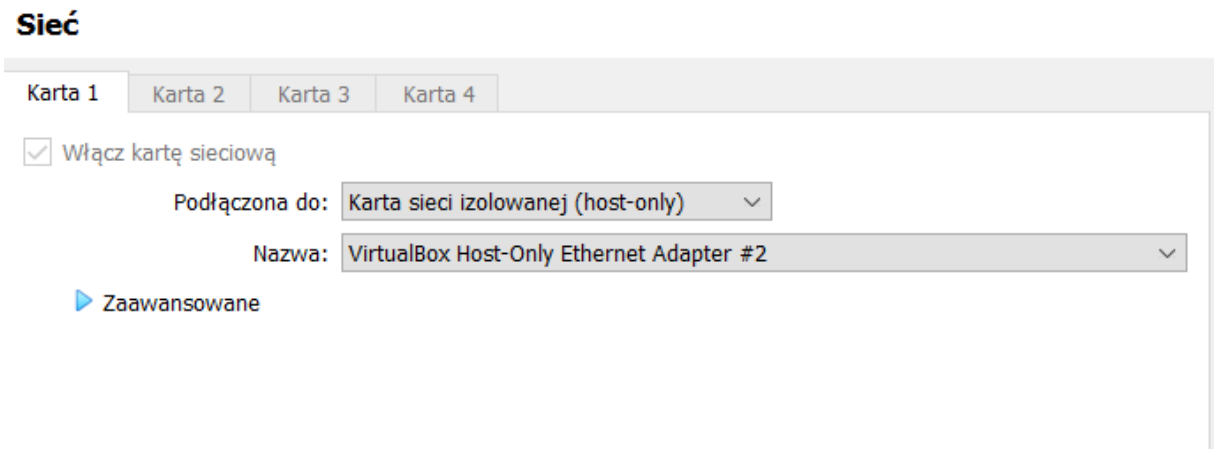
2.4. Konfiguracja karty sieciowej

Ostatnim elementem konfiguracji jest odizolowanie sieci domowej od stworzonych wcześniej środowisk. Izolacja ma za zadanie nie dopuścić do rozprzestrzenienia się złośliwego oprogramowania po dostępnych urządzeniach w sieci domowej. Wykorzystane zostaną do tego rozwiązania oferowane przez *VirtualBox*. Na potrzeby pracy wykorzystana zostanie pula adresów klasy A - 10.0.0.1/24. Prawidłowa konfiguracja serwera DHCP została pokazana na **rysunku 2.4**.



Rys. 2.4: konfiguracja serwera DHCP

Następnie należy ustawić stworzoną wcześniej kartę sieciową do obu środowisk oraz podłączyć ją do sieci izolowanej (host only). Rysunek poglądowy **2.5**



Rys. 2.5: Ustawienia sieci

2.5. Narzędzia wykorzystywane podczas analizy

Istnieje wiele narzędzi wykorzystywanych w informatyce śledczej, rozpoczynając od podstawowych narzędzi, takich jak edytory heksadecymalne, a kończąc na złożonych systemach do symulacji sieciowych i analizy behawioralnej, które są niezbędne w pracy każdego analityka

bezpieczeństwa. Wiedza o tych narzędziach i umiejętność ich efektywnego wykorzystania stanowią podstawę skutecznego reagowania na incydenty bezpieczeństwa. Programy wykorzystane w pracy to:

1. **Analiza statyczna:**
   - Sha256,
   - md5sum,
   - Floss,
   - PEview,
   - Pestudio,
   - VirusTotal,
   - Cutter,
   - Ida Pro.
2. **Analiza dynamiczna:**
   - Procmon,
   - WireShark,
   - process hacker 2,
   - x32,
   - tcpview,
   - volatility forensics.

2.6. Nakładka na system (FlareVM)

FlareVM [12] jest to nakładka o otwartym kodzie źródłowym przeznaczona dla zespołów security stworzona na wirtualne środowiska *Windows*, która automatycznie instaluje zestaw narzędzi umożliwiających analizę złośliwego oprogramowania. Instalacja wymaga interpretera poleceń *powershell* przynajmniej w wersji 5.0 oraz wyłączonego antywirusa.

Rozdział 3

Analiza złośliwego oprogramowania

Zgodnie z wcześniejszymi założeniami, część analityczna pracy zostanie podzielona na dwie kategorie, które opisane zostaną w dalszej części pracy. Przed rozpoczęciem badania należy upewnić się, że prawidłowo zostało skonfigurowane środowisko oraz wykonać zrzut pamięci wykorzystując z funkcji "zrób migawkę" oferowanej przez *VirtualBox*.

3.1. Analiza Statyczna

Analiza statyczna polega na szczegółowym przeglądzie kodu źródłowego lub binarnej struktury pliku, co pozwala na wykrycie specyficznych elementów, takich jak niejawne funkcje czy wykorzystywane słabości, bez konieczności uruchamiania samego oprogramowania [10].

3.1.1. Badanie Hash'y

Hash'em nazywamy wynik działania operacji matematycznej (nazywanej funkcją skrótu) na określonym ciągu znaków (np. na haśle lub pliku). Funkcja ta przekształca podane przez użytkownika dane wejściowe (np. hasło) na krótką, posiadającą stały rozmiar wartość znakową [4]. W celu analizy Hash'y wykorzystany zostanie VirusTotal [5], który umożliwia skanowanie poszczególnych plików, funkcji skrótu i linków i przedstawia wyniki pozwalające stwierdzić ewentualną infekcję złośliwym oprogramowaniem. Serwis ten wykorzystuje około 85 skanerów antywirusowych [30].
Jednym ze sposobów ukrycia złośliwego oprogramowania przed antywirusami, jest skorzystanie z kompresji plików. Porównując wyniki przedstawione na **rysunkach 3.1** oraz **3.2** można zauważyć różnicę w otrzymanych wartościach.

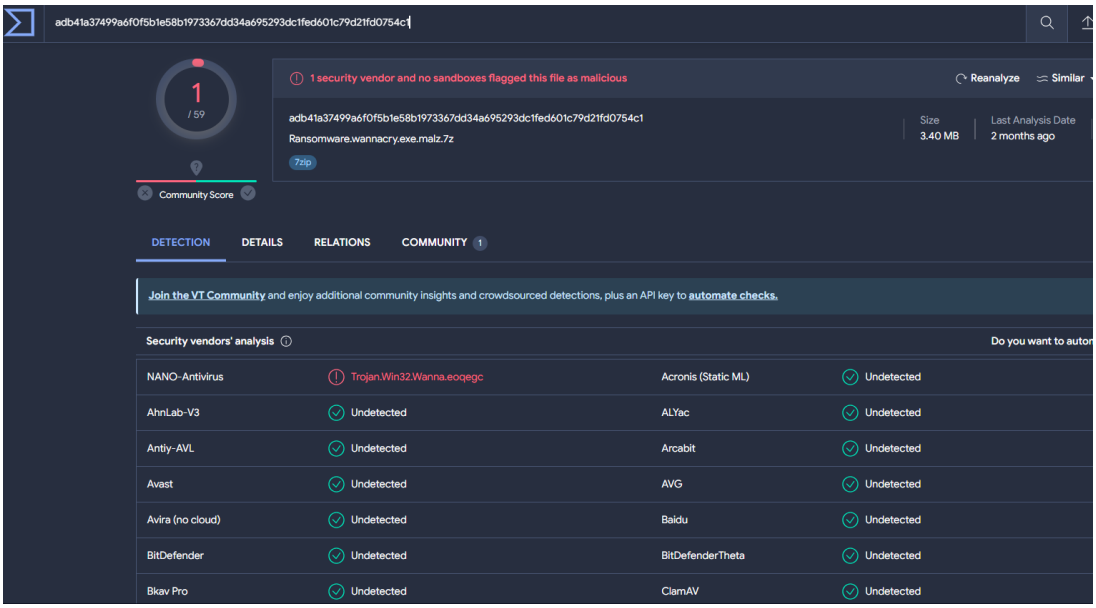
```
C:\Users\Miki\Desktop
λ md5sum Ransomware.wannacry.exe.malz.7z
4d9c771619255c9b937c34b4c50cec7e *Ransomware.wannacry.exe.malz.7z

C:\Users\Miki\Desktop
λ sha256sum Ransomware.wannacry.exe.malz.7z
adb41a37499a6f0f5b1e58b1973367dd34a695293dc1fed601c79d21fd0754c1 *Ransomware.wannacry.exe.malz.7z
```

Rys. 3.1: Wynik funkcji skrótu skompresowanego pliku

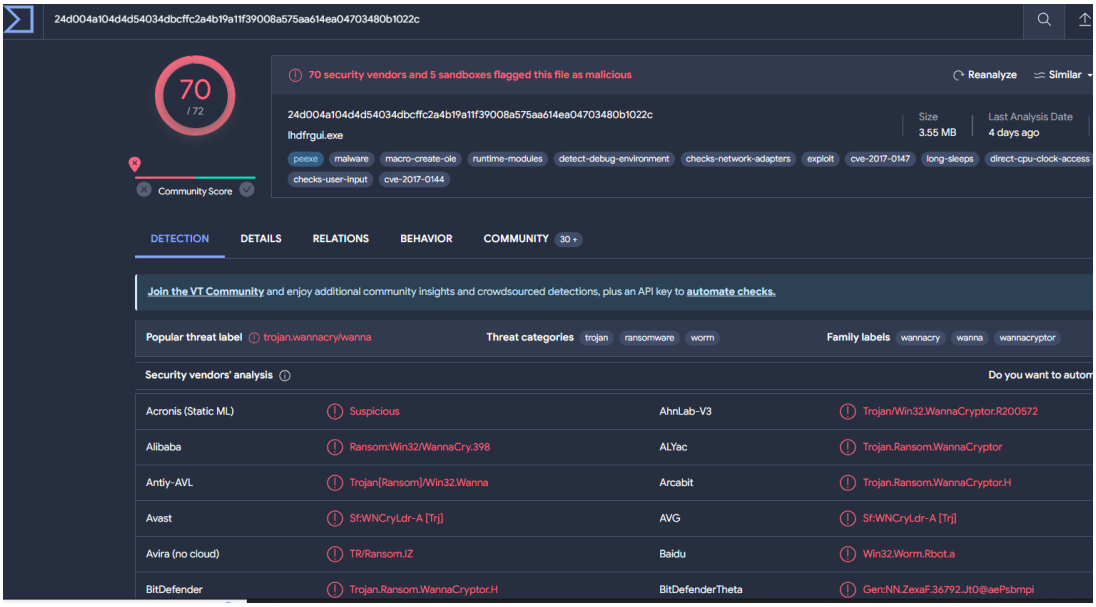
```
C:\Users\Miki\Desktop
λ md5sum Ransomware.wannacry.exe.malz
db349b97c37d22f5ea1d1841e3c89eb4 *Ransomware.wannacry.exe.malz

C:\Users\Miki\Desktop
λ sha256sum Ransomware.wannacry.exe.malz
24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c *Ransomware.wannacry.exe.malz
```

Rys. 3.2: Wynik funkcji skrótu oryginalnego pliku

Poprzez kompresję plików zmienia się *hash* pliku oraz następuje zaciemnienie kodu. Antywirusy, których wynik skanu bazuje tylko na sygnaturach, może okazać się niewystarczający do zachowania bezpieczeństwa. Wyniki skanu antywirusów na podstawie sygnatur zostały umieszczone na **rysunkach 3.3** i **3.4**.



Rys. 3.3: Skan skompresowanego oprogramowania WannaCry przez VirusTotal



Rys. 3.4: Skan oryginalnego oprogramowania WannaCry przez VirusTotal

Analizując wyniki skanów, można zauważyć, że tylko jeden z pięćdziesięciu dziewięciu skanerów, był w stanie określić skompresowany plik WannaCry jako niebezpieczny. Wynik skanu dla oryginalnego pliku jednoznacznie wskazuję na Ransomware.

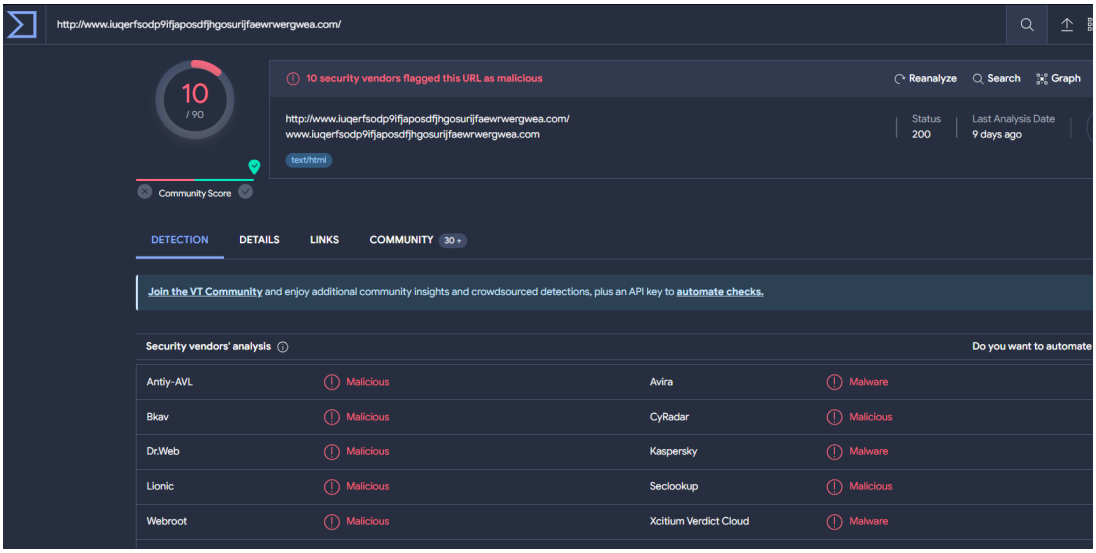
3.1.2. Analiza ciągów znaków

W celu zautomatyzowania i przyspieszenia szukania czytelnych znaków w pliku, wykorzystany w pracy zostanie program Floss [29].
Wynik narzędzia Floss zawiera w sobie wiele niepotrzebnych dla projektu informacji, dlatego te warte uwagi zostały opisane w **tablicy 3.1**. wraz ze wskazaniem, do czego może zostać wykorzystany dany ciąg znaków.

Tab. 3.1: Wykaz najpopularniejszych kategorii złośliwego oprogramowania

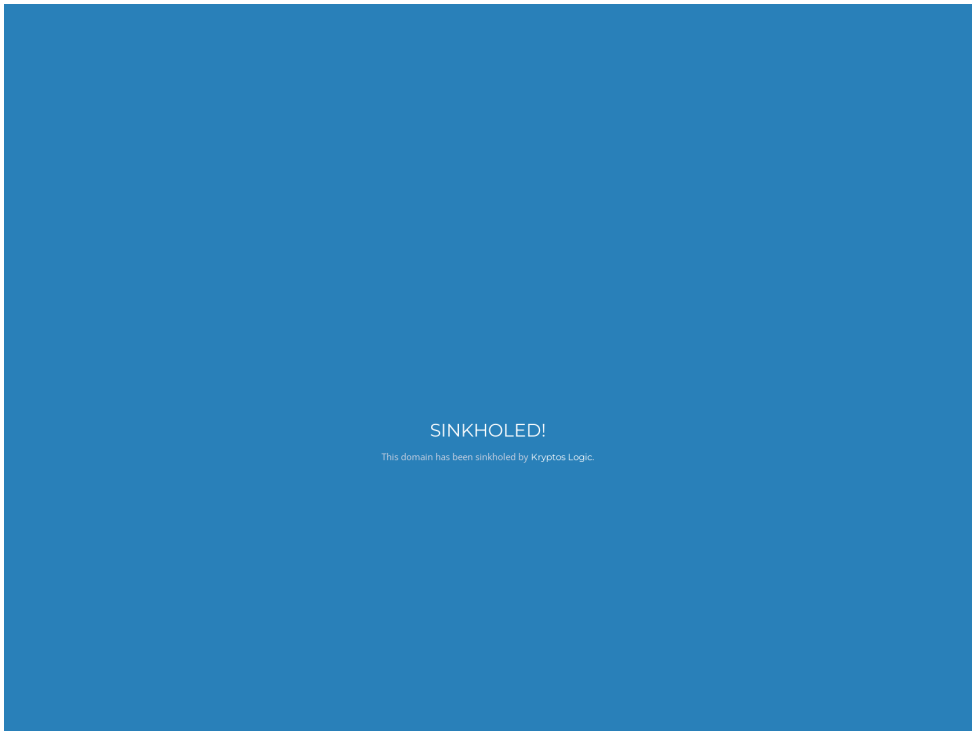
Ciąg znaków	Opis
`C:/%s/qeriuwjhrf`	Jest to ścieżka w której zostanie pobrany lub stworzony plik.
`http://www.iuqerfsow....com`	Adres URL z którym łączy się złośliwe oprogramowania.
`CryptDecrypt`	Połączenie API może posłużyć do szyfrowania danych.
`IsDebuggerPresent`	Połączenie API prawdopodobnie wykorzystane do zmiany zachowania pliku w momencie debugowania.
`cmd.exe /c "%s"115p7UMMngoj1pMvkrLn...`	Wywołanie komendy poprzez cmd.
`attrib +h`	Ukrywa daną ścieżkę lub plik.
`Sleep`	Połączenie API służy do opóźnienia działania programu w celu nierozpoznania przez systemy antywirusowe lub sandboxy.
`InternetCloseHandle`	Połączenie API do odczytania stanu połączenia internetowego.

URL znaleziony podczas analizy został sklasyfikowany przez VirusTotal jako podejrzany. Wynik analizy przedstawia **rysunek 3.6** .



Rys. 3.5: Skan URLa za pomocą VirusTotal

Korzystając z narzędzia Urlscan dostępnego pod linkiem `https://urlscan.io/` można stwierdzić, że strona ta została zawieszona przez `kryptoslogic.com`, czyli firmę zajmującą się bezpieczeństwem.



Rys. 3.6: Skan URL za pomocą Urlscan

Bazując na samych *stringach*, można wysunąć wniosek, że oprogramowanie to po połączeniu ze stroną internetowej pobiera dodatkową zawartość, jednocześnie szyfrując dane użytkownika. Hipoteza ta zostanie sprawdzona w dalszej części pracy.

3.1.3. Możliwości narzędzia PEstudio

Jednym z najlepszych narzędzi do statycznej analizy jest "PEstudio". Pierwszą rzeczą, którą możemy wywnioskować po przeanalizowaniu złośliwego oprogramowania przez PEstudio są dwa pierwsze bajty. Określają one rozszerzenie, z którego korzysta plik. W przypadku oprogramowania WannaCry jest to MZ, czyli plik wykonywalny z rozszerzeniem *.exe*. Rysunek poglądowy **3.7**.

first-bytes > hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes > text	M Z . . . . . @ . . . . .

Rys. 3.7: Dwa pierwsze bajty oprogramowania WannaCry

Kolejną ważną rzeczą znalezioną przy pomocy narzędzia jest indykator trzech innych plików, do których odwołuje się złośliwe oprogramowanie. Nazwy plików przedstawione są na **rysunku 3.8**

indicator (36)	detail	level
file > embedded	signature: executable, location: .data, offset: 0x0000B020, size: 5263716 (...	1
file > embedded	signature: executable, location: .data, offset: 0x0000F080, size: 5297524 (...	1
file > embedded	signature: executable, location: .rsrc, offset: 0x000320A4, size: 3514368 (...	1

Rys. 3.8: Odwołania do plików

Narzędzie PEstudio zawiera też funkcjonalności narzędzi wymienionych wcześniej, takie jak analiza stringów wraz z przyporządkowaniem do odpowiedniej taktyki opisanej w MITRE ATT&CK lub też informację dotyczące *Hash'a* dla algorytmu SHA256. Odwołania do nagłówków zostały przestawione na **rysunku 3.9**

footprint (10)	value
general	
file > sha256	24D004A104D4D54034DBCFFC2A4B19A11F39008A575AA614EA04703480B1022C
dos-stub > sha256	57CA1D7508C252D07E1780DCBA940B59B5044B132756318FDCACD3C11B4A7351
dos-header > sha256	6C382A1C16DBA41B4FF6F0D728E9E92AEBFE2EE0C7FEB30A0E63B15EAF6C4B44
rich-header > sha256	D4496034DE1F5AF97B361FCDC86EB5D939978830DFF8BF01B6AB3C93961AA425
section > .text > sha256	7609ECC798A357DD1A2F0134F9A6EA06511A8885EC322C7ACD0D84C569398678
section > .rdata > sha256	532E9419F23EAF5EB0E8828B211A7164CBF80AD54461BC748C1EC2349552E6A2
section > .data > sha256	6F93FB1B241A990ECC281F9C782F0DA471628F6068925AAF580C1B1DE86BCE8A
section > .rsrc > sha256	1EFE677209C1284357EF0C7996A1318B7DE3836DFB11F97D85335D6D3B8A8E42
resource > executable > sha256	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA
version > sha256	2F3FC51546ADA848DFC8E775554C0DE3689D6FAE7BA4BF3D40E3C8DEC68B277B

Rys. 3.9: Odwołania do nagłówków

Można również zauważyć wartości nagłówków. Opis znajduje się w **tabeli 3.4**. Porównując wartości poszczególnych plików wierszy "raw-size"oraz "virtual size"można też określić, czy pliki zostały skompresowane. Wartości te nie powinny się od siebie znacząco różnić. Wartości nagłówków przedstawia **rysunek 3.12**

property	value	value	value	value
headers	header[0]	header[1]	header[2]	header[3]
name	.text	.rdata	.data	.rsrc
footprint > sha256	7609ECC798A357DD1A2F01...	532E9419F23EAF5EB0E8828B...	6F93FB1B241A990ECC281F9...	1EFE677209C1284357EF0C79...
entropy	6.135	3.504	6.100	7.995
file-ratio (99.89%)	0.99 %	0.11 %	4.29 %	94.50 %
raw-address (begin)	0x00001000	0x0000A000	0x0000B000	0x00032000
raw-address (end)	0x0000A000	0x0000B000	0x00032000	0x0038D000
raw-size (3719168 bytes)	0x00009000 (36864 bytes)	0x00001000 (4096 bytes)	0x00027000 (159744 bytes)	0x0035B000 (3518464 bytes)
virtual-address	0x00001000	0x0000A000	0x0000B000	0x00310000
virtual-size (6718034 bytes)	0x00008BCA (35786 bytes)	0x00000998 (2456 bytes)	0x0030489C (3164316 bytes)	0x0035A454 (3515476 bytes)

Rys. 3.10: Pamięć nagłówków prezentowana przez PEstudio

Kolejną z opcji programu PEstudio jest "libaries". Zakładka ta pozwala na zapoznanie się z bibliotekami, z których korzysta przez złośliwe oprogramowanie oraz oznacza te często wykorzystywane są w sposób podejrzany. Złośliwe oprogramowanie użytkuje biblioteki wbudowane w system, takie jak *KERNEL32.DLL*, tworzyć własne, korzystać z bibliotek innego oprogramowania np. *msedge_elf.dll* lub je modyfikować wstrzykując niebezpieczny kod. Informację te będą klucze w budowaniu sygnatur YARA.

library (7)	duplicate (0)	flag (3)	first-thunk-original (INT)	first-thunk (IAT)	type (1)	imports (91)	group	description
KERNEL32.dll	-	-	0x0000A2B0	0x0000A030	implicit	32	-	Windows NT BASE API Client
ADVAPI32.dll	-	-	0x0000A280	0x0000A000	implicit	11	-	Advanced Windows 32 Base API
WS2_32.dll	-	x	0x0000A3C4	0x0000A144	implicit	13	network	Windows Socket Library
MSVCP60.dll	-	-	0x0000A334	0x0000A0B4	implicit	2	-	Windows C Runtime Library
iphlpapi.dll	-	x	0x0000A3FC	0x0000A17C	implicit	2	network	IP Helper API
WININET.dll	-	x	0x0000A3B4	0x0000A134	implicit	3	network	Internet Extensions for Win32 Library
MSVCRT.dll	-	-	0x0000A340	0x0000A0C0	implicit	28	-	Microsoft C Runtime Library

Rys. 3.11: Wykorzystywane biblioteki przez złośliwe oprogramowanie WannaCry

Zakładka "imports" zawiera w sobie połączenia API i tak jak w przypadku zakładki "libaries" oznacza te, które mogą zostać użyte w sposób niewłaściwy oraz przypisuje je do odpowiedniej taktyki MITRE ATT&CK. Opis poszczególnych połączeń można znaleźć pod linkiem `https://malapi.io` [18].

imports (91)	flag (28)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (16)	technique (8)	type (1)	ordinal (13)	library (7)
ChangeServiceConfig2A	x	0x0000A6C0	0x0000A6C0	52 (0x0034)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
CloseServiceHandle	-	0x0000A672	0x0000A672	62 (0x003E)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
OpenSCManagerA	-	0x0000A69A	0x0000A69A	429 (0x01AD)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
StartServiceA	-	0x0000A662	0x0000A662	585 (0x0249)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
CreateServiceA	x	0x0000A688	0x0000A688	100 (0x0064)	services	T1543 | Create or Modify System Process	implicit	-	ADVAPI32.dll
OpenServiceA	-	0x0000A714	0x0000A714	431 (0x01AF)	services	T1543 | Create or Modify System Process	implicit	-	ADVAPI32.dll
SetServiceStatus	-	0x0000A6AC	0x0000A6AC	580 (0x0244)	services	T1543 | Create or Modify System Process	implicit	-	ADVAPI32.dll
Sleep	-	0x0000A408	0x0000A408	854 (0x0356)	execution	T1497 | Sandbox Evasion	implicit	-	KERNEL32.dll
GetTickCount	-	0x0000A410	0x0000A410	479 (0x01DF)	reconnaissance	T1124 | System Time Discovery	implicit	-	KERNEL32.dll
RegisterServiceCtrlHandlerA	-	0x0000A6D8	0x0000A6D8	524 (0x020C)	services	T1106 | Execution through API	implicit	-	ADVAPI32.dll
MoveFileExA	x	0x0000A576	0x0000A576	623 (0x026F)	file	T1105 | Remote File Copy	implicit	-	KERNEL32.dll
GetCurrentThreadId	x	0x0000A524	0x0000A524	326 (0x0146)	execution	T1057 | Process Discovery	implicit	-	KERNEL32.dll
CryptAcquireContextA	x	0x0000A638	0x0000A638	133 (0x0085)	cryptography	T1027 | Obfuscated Files or Information	implicit	-	ADVAPI32.dll
CryptGenRandom	x	0x0000A650	0x0000A650	150 (0x0096)	cryptography	T1027 | Obfuscated Files or Information	implicit	-	ADVAPI32.dll
rand	x	0x0000A824	0x0000A824	678 (0x02A6)	cryptography	T1027 | Obfuscated Files or Information	implicit	-	MSVCRT.dll
srand	x	0x0000A852	0x0000A852	692 (0x02B4)	cryptography	T1027 | Obfuscated Files or Information	implicit	-	MSVCRT.dll

Rys. 3.12: Połączenia API wykorzystywane przez złośliwe oprogramowanie

Tab. 3.2: Wykaz nagłówków wraz z opisem [19]

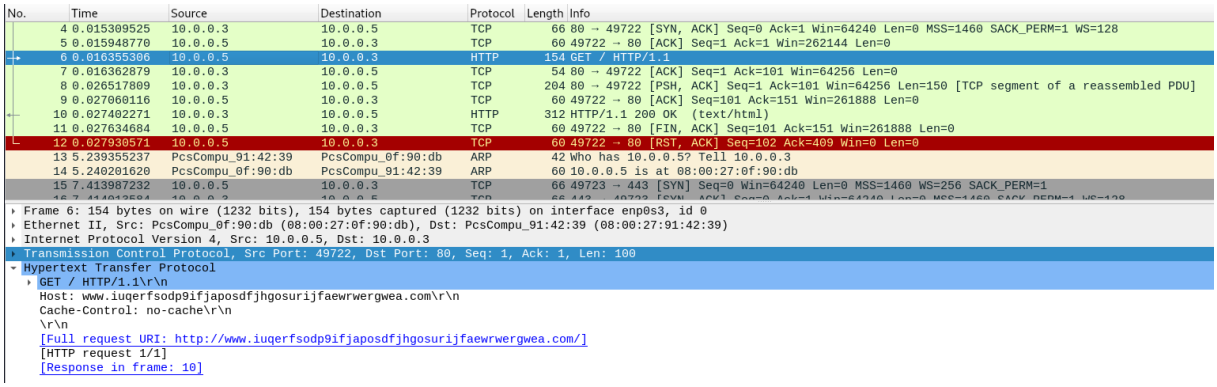
Ciąg znaków	Opis
`.text`	Zawiera instrukcję wykonywane przez CPU, powinno być to jedyne miejsce w którym trzymany jest kod
`.rdata`	Zawiera importowane i eksportowane informację w statusie read-only, z tego pliku najczęściej pobierane są informację wykorzystywane przez PEStudio
`.rsrc`	Przetrzymuje informacje takie jak: ikony, obrazy, string'i
`.data`	Umieszczone są w nim globalne dane, dostępne w każdym etapie programu

3.2. Analiza Dynamiczna

Analiza dynamiczna jest nazywana również „analizą behawioralną", ponieważ ocenia malware na podstawie jego złośliwego zachowania. Ten typ analizy wykrywa zmiany wprowadzane w systemach „na żywo" – jest także bardziej niebezpieczny, gdyż dochodzi w nim do wykonania złośliwego kodu. Malware jest uruchamiany w celu zebrania cennych informacji o jego aktywności, w tym utworzonych plikach i folderach, otwieranych portach i wywoływanych adresach URL. Badane w tym przypadku są również takie aspekty jak wykorzystywane funkcje i biblioteki, odwołania do innych aplikacji/narzędzi, modyfikacje ustawień systemowych, uruchamiane procesy i usługi, zmiany w rejestrach, tworzone tunele komunikacyjne, wykorzystywane nowe wektory infekcji kolejnych ofiar i tym podobne [1].

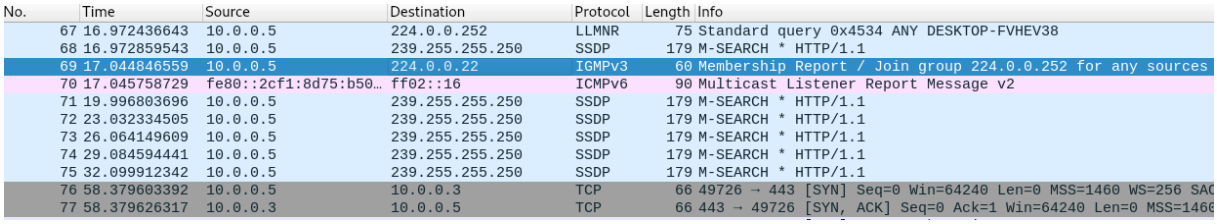
3.2.1. Pierwsze uruchomienie złośliwego oprogramowania

Pierwsze uruchomienie złośliwego oprogramowania na pierwszy rzut oka pozornie nie zadziałało, dlatego należy przeanalizować ruch sieciowy, który zgodnie z założeniami określonymi w rozdziale drugim pracy, ma za zadanie symulowanie sieci oraz usług w celu oszukania złośliwego oprogramowania. W tym celu zostanie wykorzystany Wireshark [8]. Wireshark jest to oprogramowanie pozwalające na analizę pakietów ruchu sieciowego, stworzone przez Geralda Combsa.



Rys. 3.13: Ruch sieciowy pobrany z narzędzia WireShark dla oprogramowania WannaCry (cz. 1)

Analizując ruch sieciowy przedstawiony na **rysunku 3.14** po uruchomieniu złośliwego oprogramowania, możemy zauważyć udaną próbę stabilizacji połączenia do adresu 10.0.0.5 na port 80 z portu 49722. Jest to wystawiona usługa HTTP stworzona przez INetSim. Mówią o tym pakiety 3, 4 i 5. Jest to tak zwany 3-Way Handshake.
Następny interesujący pakiet to połączenie HTTP o numerze pakietu 6. Badając ramkę pakietu można zauważyć połączenie do URL http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwe.com/, który został znaleziony i opisany w części statycznej analizy.



Rys. 3.14: Ruch sieciowy pobrany z narzędzia WireShark dla oprogramowania WannaCry (cz. 2)

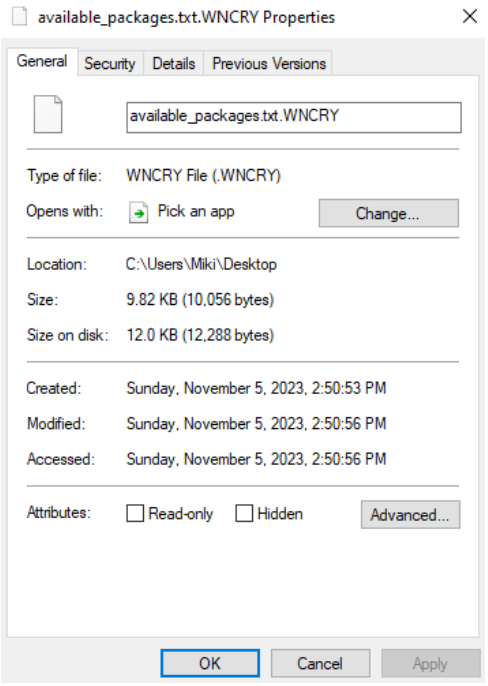
Połączenia z pakietów 71-77 na adres 239.255.255.250 po protokole SSDP, który najczęściej służy poszukiwania urządzeń w sieci, może wskazywać na próbę zainfekowania innych maszyn lub routerów. W dalszej części wykonywane są połączenia ARP, które mają za zadanie zidentyfikować dostępne adresy IP.

3.2.2. Druga próba uruchomienia złośliwego oprogramowania

Ponieważ nie udało się poprawie uruchomić złośliwego oprogramowania poprzez wykorzystanie *INetSim*, należy sprawdzić. co się stanie w momencie wyłączenia usługi. W tym celu wykorzystać należy wcześniej zrobiony zrzut pamięci, aby przywrócić stan przed rozpoczęciem analizy.

Rys. 3.15: Pogląd pulpitu po uruchomieniu złośliwego oprogramowania

Po wyłączeniu dodatkowych usług oferowanych przez INetSim, można zauważyć zadziałanie złośliwego oprogramowania WannaCry. Podgląd usług widoczny na **rysunku 3.15**. Uwagę przykuwa wyświetlające się dodatkowe okienko, w którym opisano działania umożliwiające odszyfrowanie danych oraz podano adres portfela na, który należy wpłacić okup. Zgodnie z opisem w okienku, prywatne pliki zostały zaszyfrowane, przez co są niedostępne dla użytkownika. Podgląd pliku widoczny na **rysunku 3.17**. Rozszerzenia plików zmieniły się na *WNCRY*.



Rys. 3.16: Pogląd plików zaszyfrowanych przez złośliwe oprogramowanie

Pojawił się również plik wykonywalny o nazwie @WanaDecryptor@.exe oraz plik graficzny @WanaDecryptor@.bmp.

3.2.3. Analiza procesów narzędziem ProcMon

Process Monitor [16] to zaawansowane narzędzie do monitorowania systemu Windows, które pokazuje w czasie rzeczywistym aktywność systemu plików, rejestru i procesów/wątków.
Łączy w sobie funkcje starszych narzędzi Sysinternals, Filemon i Regmon, i dodaje obszerną listę ulepszeń, w tym bogate i nieniszczące filtrowanie, kompleksowe właściwości zdarzeń, takie jak identyfikatory sesji i nazwy użytkowników, wiarygodne informacje o procesach, pełne stosy wątków ze zintegrowaną obsługą symboli dla każdej operacji, jednoczesne rejestrowanie do pliku i wiele więcej. Wyjątkowo zaawansowane funkcje sprawiają, że ProcMon jest podstawowym narzędziem w zestawie narzędzi do rozwiązywania problemów systemowych i polowania na złośliwe oprogramowanie [17].
W celu lepszego zrozumienia zmiany w rejestrze należy zapoznać się z podstawowymi operacjami na nim. Funkcje te są ważne, aby dokładniej zrozumieć zmiany, które zaszły w ustawieniach systemu oraz bezpieczeństwie.

Tab. 3.3: Podstawowe ścieżki dla group policy

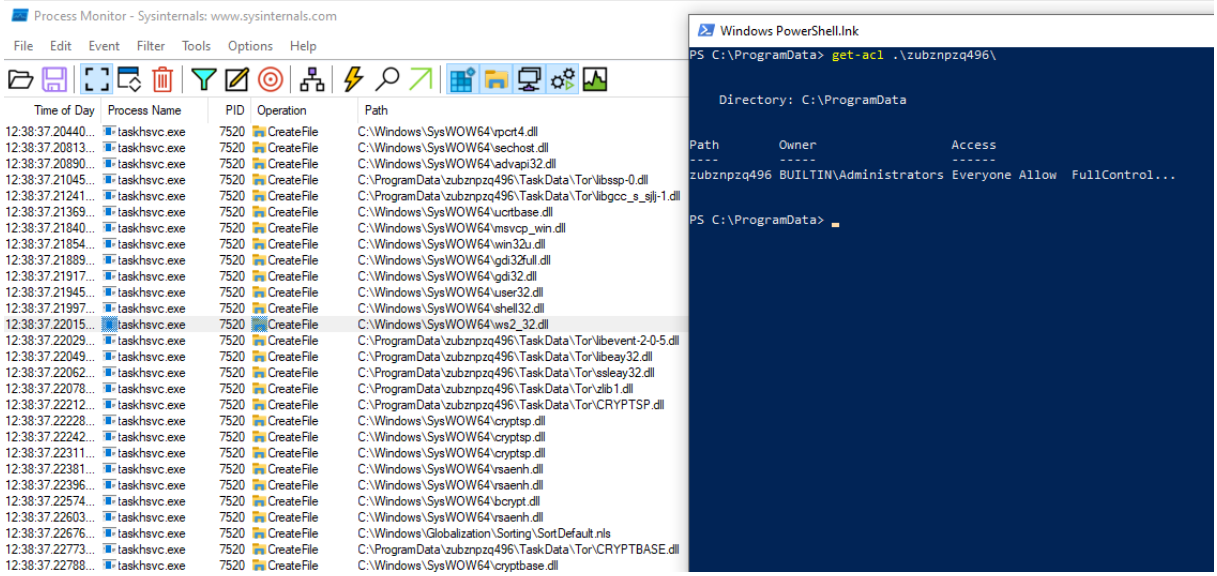
Nazwa funkcji	Opis
HKEY_local_Machine	Zawiera ustawienia dla systemu
HKEY_CURRENT_USER	Zawiera ustawienia dla bieżącego użytkownika
HKEY_CLASSES_ROOT	Zawiera informacje o rozszerzeniach plików, identyfikator programowy i dane identyfikatora interfejsu
HKEY_CURRENT_CONFIG	Zawiera aktualne ustawienia hardware'u
HKEY_USERS	Definiuje ustawienia dla użytkowników

Poprzez funkcję RegOpenKey otworzony został rejestr ustawień internetu znajdującego się pod ścieżką "HKLM/SOFTWARE/Policies/Microsoft/Windows/CurrentVersion/Internet Settings", a za pomocą RegSetValueEx zmieniona została wartość tego rejestru.

Tab. 3.4: Podstawowe funkcje rejestru

Nazwa funkcji	Opis
RegOpenKey	Umożliwia edytowanie rejestru
RegSetValue	Dodaje nową wartość do rejestru
RegGetValue	Zwraca dane rejestru
RegCreateKey	Tworzy nowy rejestr

Analizując logi ProcMon'a, można również zauważyć proces taskhsvc.exe, który stworzył nowy folder widoczny tylko dla administratora o nazwie *zubznpzq496*, najprawdopodobniej znajdują się tam odwołania do sieci TOR. Utworzone zostały tam również pliki wykonywalne *taskdl.exe, tasksche.exe, taskse.exe*, prawdopodobnie mające imitować prawdziwy proces *task.exe* w celu ukrycia się. Wykorzystując kolejne narzędzie PowerShell za pomocą komendy *Get-Filehash* wyciągnięte zostały hash'e plików widoczne w **tabeli 3.5**.
Każdy z nich został przeskanowany narzędziem VirusTotal, który wykazał potencjalnie niebezpieczne oprogramowanie. Pozostałe pliki utworzone to min. *b.wncry, c.wncry, f.wncry, r.wncry, s.wncry, t.wncry, u.wncry, 00000000.res, 00000000.pky, 00000000.eky.*



Rys. 3.17: Nowy Folder złośliwego oprogramowania

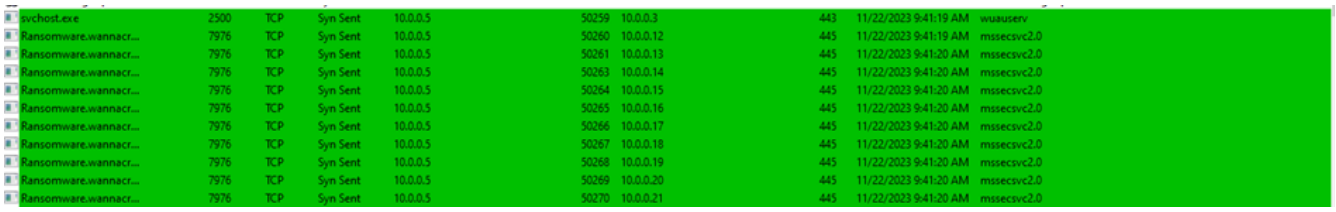
Tab. 3.5: Wykaz nagłówków wraz z opisem

Pliki	Opis
`taskse.exe`	2CA2D550E603D74DEDDA03156023135B38DA3630CB014E3D00B1263358C5F00D
`taskdl.exe`	4A468603FDCB7A2EB5770705898CF9EF37AADE532A7964642ECD705A74794B79
`tasksche.exe`	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA

Wykorzystując znalezione pliki wykonywalne należy w miarę możliwości poszukać inne podejrzane aktywności. Tasksche.exe tworzy nowy wpis w rejestrze o nazwie *WanaCrypt0r* oraz plik o nazwie *@WannDecryptor@.exe* pod ścieżką *"C:/ProgramData/zubznpzq496/@WanaDecryptor@.exe*, skrót utworzył się również na pulpicie. Plik *taskdl* wykonuję dużą ilość operacji na plikach tymczasowych, najprawdopodobniej tworząc je lub usuwając.

3.2.4. Analiza sieciowa przy pomocy TCPView

Chcąc zobaczyć aktywne połączenia sieciowe oraz procesy, które je wywołują, można użyć narzędzia TCPView [14] stworzonego przez Marka Russinovicha.



Rys. 3.18: Aktywne połączenia WannaCry widoczne w narzędziu TCPview

Po uruchomieniu pliku WannaCry nastąpiła próba połączenia się do innych urządzeń w sieci po protokole SMB na port 445. Jest to cecha charakterystyczna dla złośliwego oprogramowania typu Robak. Analizując dostępne informacje w internecie można stwierdzić, że WannaCry

korzystał z podatności EternalBlue przeznaczonej na protokół SMB w wersji 1.0, dzięki tej podatności jest w stanie wykonywać zdalnie kod na komputerze ofiary, co może prowadzić do dalszych szkód. Opisany ruch widoczny jest na **rysunku 3.18**, Po próbie ekspansji zostało jedno niestandardowe połączenie na port 9050 wywołane przez proces *taskhsvc.exe*. Korzystając z przycisku "Check payment", który wyświetla się w okienku z okupem, również nastąpiło nowe połączenie wykonane przez proces *wannacry.exe* na 9050. Port ten najczęściej służy do komunikowania się z siecią TOR. Sieć TOR została stworzona w celu zachowania anonimowości, natomiast wykorzystywana jest również w celu popełnienia różnego rodzaju przestępstw.

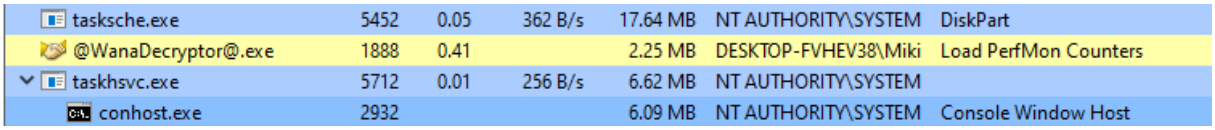
3.2.5. ProcessHacker

Narzędzie ProcessHacker [25] zaawansowane oprogramowanie do monitorowania i zarządzania procesami systemowymi. Aplikacja ta umożliwia szczegółowy wgląd i manipulację w działające procesy oraz usługi systemowe, a także pozwala na monitorowanie wykorzystania takich zasobów jak CPU i pamięć RAM.

Rys. 3.19: Zużycie zasobów przed uruchomieniem złośliwego oprogramowania

Rys. 3.20: Zużycie zasobów po uruchomieniu złośliwego oprogramowania

Porównując wykresy przed uruchomieniem złośliwego oprogramowania **3.19** oraz po jego zadziałaniu **3.20** zauważalne jest znaczne zwiększenie wartości CPU, które w najwyższym punkcie uzyskało aż 60%. Wykres I/O (wejścia/wyjścia) przedstawia dwie wartości:
Read (R): Odnosi się do danych odczytywanych z dysków. Na wykresie reprezentuje to ilość danych przesłanych do pamięci RAM z dysku lub innych nośników pamięci w danym czasie.
Write (W): Odnosi się do danych zapisywanych na dyskach. Na wykresie przedstawia to ilość danych przesłanych z pamięci RAM do trwałego magazynu danych.
Po uruchomienia pliku WannaCry obie te wartości znacząco wzrosły, co sugeruje sporą aktywność na plikach.



Rys. 3.21: Aktywne procesy widoczne w narzędziu ProcessHacker

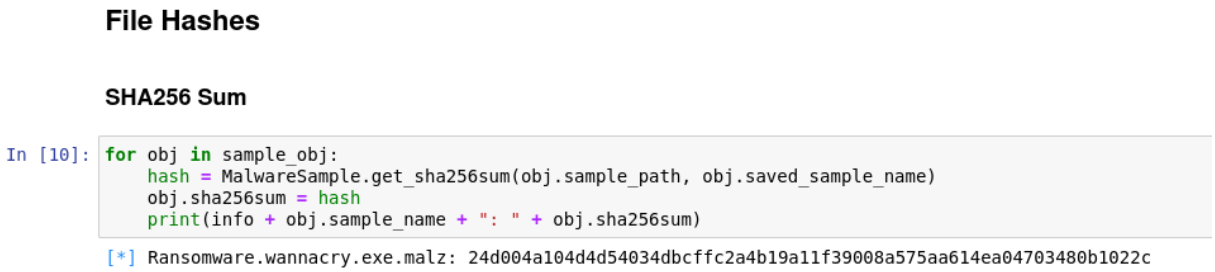
Rysunek 3.27 przedstawia wykaz aktywnych procesów wygenerowanych przez złośliwe oprogramowanie, wraz z ich podrzędnymi procesami, które są kreowane przez główny proces.

3.3. Automatyzacja analizy

Istotnym aspektem pracy analitycznej jest zdolność do automatyzacji pewnych wykonywanych zadań, co nie tylko oszczędza czas, ale także pomaga wyłapać elementy, które mogły zostać nieumyślnie przeoczone przez analityka.

3.3.1. Jupyter

Jupyter [20] to narzędzie stworzone przez Michaela Taggarta, pomaga ono między innymi w analizie statycznej. Jedną z funkcji *Jupyera* jest automatyczne szukanie ciągów znaków, konwertowanie plików do zipa oraz liczenie sum kontrolnych pokazanych na **rysunku 3.22**. Zebrane dane następnie za pomocą API wysyła do VirusTotal i sporządza raport.

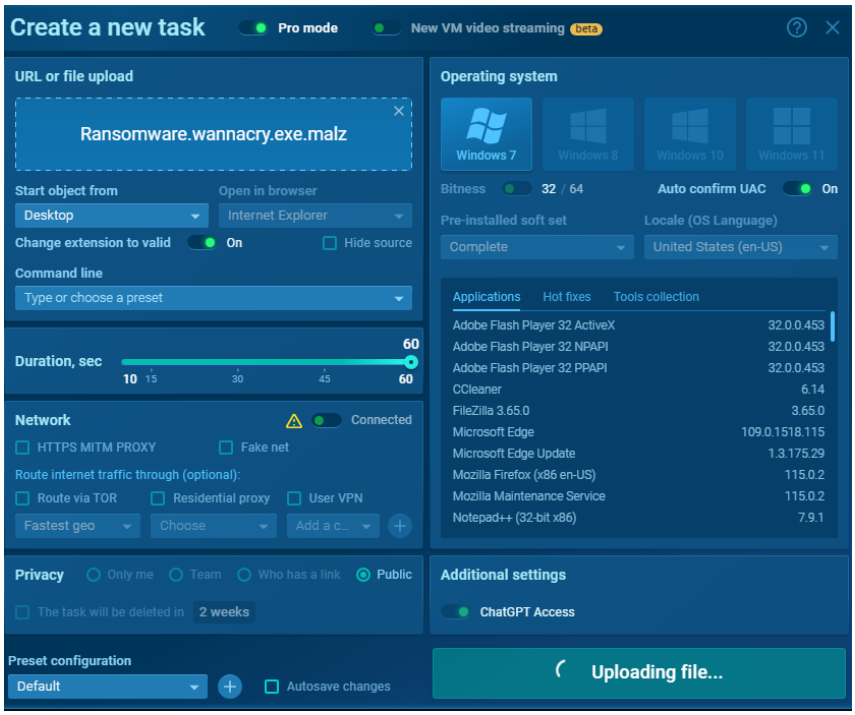


Rys. 3.22: Funkcjonalność Jupyter - suma kontrolna

3.3.2. Analiza plików w piaskownicy

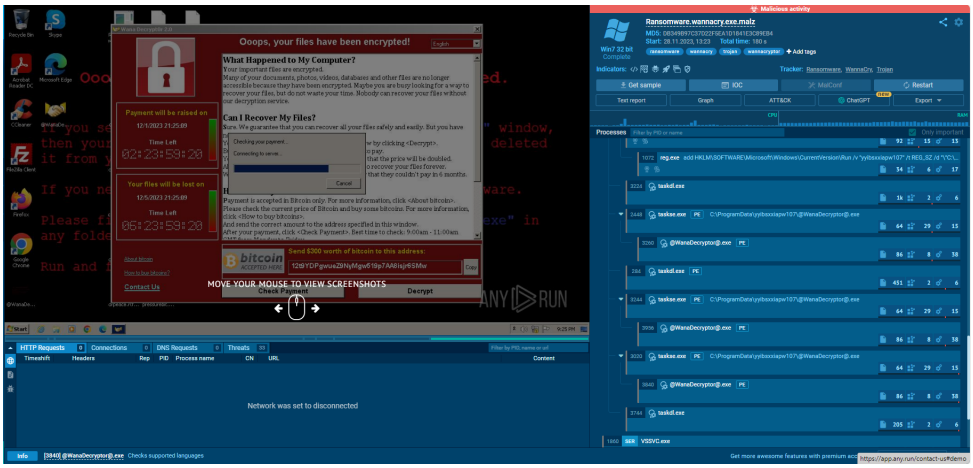
Sandbox jest to tak zwana piaskownica, czyli odizolowane środowisko dające możliwość uruchomienia złośliwego oprogramowania w sposób bezpieczny. Jednym z najlepszych *sandboxów* jest any.run [7]. Ponieważ analizy wykonywane tym narzędziem obywają się poprzez aplikację webową, badanie niesie to za sobą pewne ryzyka i ograniczenia. Wykorzystane

podczas badania skany te są publiczne dostępne dla każdego lub przechowywane są na serwerach dostawcy, podczas analizy może okazać się, że próbka przez nas analizowana posiada skradzione dane (takie jak informację wrażliwe, imiona i nazwiska klientów) lub dane poufne dla firmy (np. faktury). Przez co zgodnie ze standardami takimi jak RODO może zostać to zgłoszone jako wyciek danych. Drugą wadą tego rozwiązania jest prawdopodobieństwo rozpoznania przez złośliwe środowisko piaskownicy, może to poskutkować innym, mylącym zachowaniem i przebiegiem programu. Niemniej możliwości platformy any.run są rozległe.
Wersja premium między innymi pozwala na wybór systemu operacyjnego, konfigurację sieci, ustawienie prywatności, a także na zainstalowanie narzędzi. Okno konfiguracji piaskownicy przedstawia **rysunek 3.23**



Rys. 3.23: Funkcjonalności any.run - konfiguracja środowiska

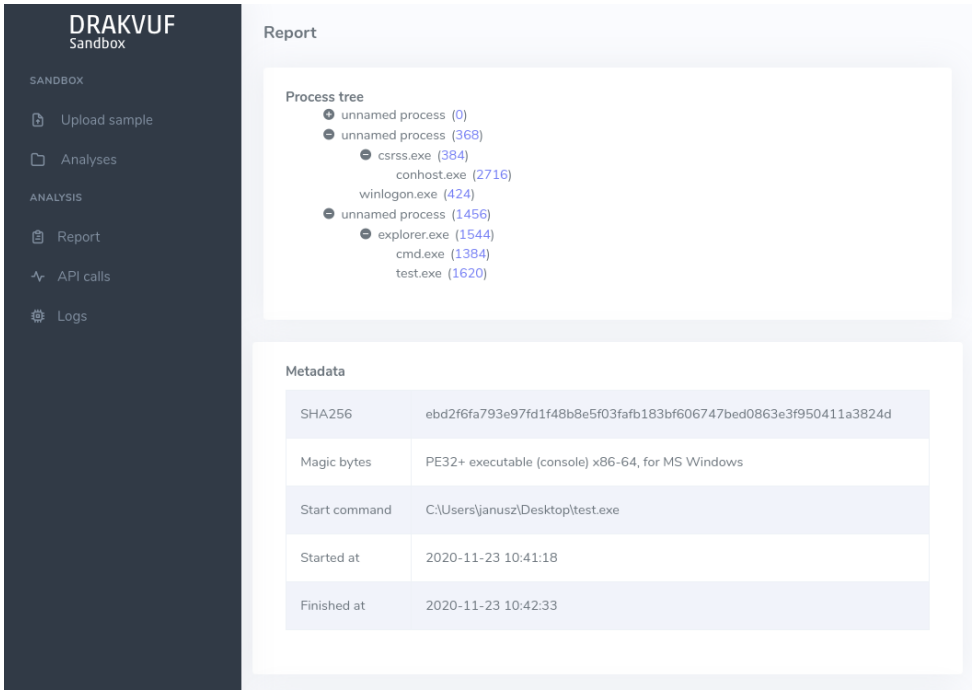
Any.run pozwala śledzić ruch sieciowy w tym ruch DNS, określić związane z nim niebezpieczeństwa, a także proces, który wykonał dane połączenie. Pokazuje również zmienione i utworzone pliki w trakcie ataku, śledzi wykonywane procesy, klasyfikuje ataki na podstawie MITRE ATT&CK oraz korzysta z technologii ChatGPT. Podgląd na **rysunku 3.24**



Rys. 3.24: Funkcjonalności any.run - panel główny

3.3.3. DRAKVUF Sandbox

DRAKVUF sandbox [3] jest polskim projektem o dostępnym kodzie źródłowym, który można zainstalować lokalnie na komputerze. Służy, podobnie jak any.run, do przyśpieszenia pracy analitykom. Narzędzie to wyróżnia się bezagentową analizą, co utrudnia złośliwemu oprogramowaniu wykrycie jego obecności, monitoruje zachowania na poziomie jądra systemu, a także pozwala na jednoczesną analizę wielu sampli.



Rys. 3.25: Podgląd narzędzia DRAKVUF

3.3.4. EnCase Forensic

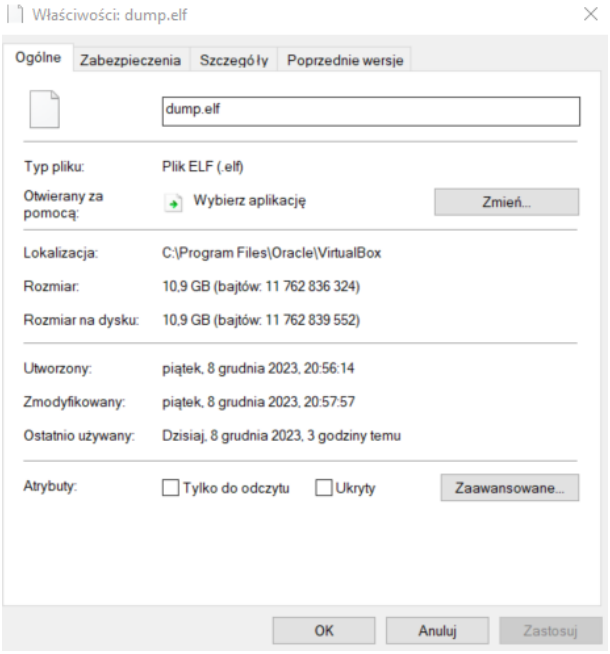
Narzędzie EnCase Forensic [13] firmy MediaRecovery to również polski produkt pomagający podczas informatyki śledczej. Jego główne funkcje to przeszukiwanie, gromadzenie, analiza danych, przeszukiwanie poczty i Internetu (np. po słowach kluczowych), analizę systemu, w tym rejestrów, przeglądanie zdjęć, rozpoznawanie niektórych systemów szyfrujących i wiele innych. Oprogramowanie jest zaprojektowane w taki sposób, aby zachować integralność danych oraz zgodność z normami sądowymi. To oznacza, że dowody zgromadzone za pomocą EnCase mogą być prezentowane w sądzie, ponieważ procesy i procedury używane do ich zbierania są zgodne z wymogami prawnymi.

3.4. Analiza wirtualnego dysku

Podczas incydentów bezpieczeństwa najważniejszą rzeczą jest zabezpieczenie logów związanych z atakiem złośliwego oprogramowania. W tym celu kopiuje się zawartość dysku, który jest istotnym elementem podczas ewentualnej rozprawy w sądzie lub może zostać wykorzystany do analizy powłamaniowej.

3.4.1. Utworzenie kopii zapasowej

Pierwszym krokiem podczas analizy powłamaniowej jest utworzenie kopii zawartości dysku wirtualnego na którym uruchomione zostało złośliwe oprogramowanie. W tym celu należy wykorzystać komendę[2]:
vboxmanage debugvm "nazwa_maszyny" dumpvmcore filename "Nazwa_pliku".elf
Utworzony w ten sposób plik widoczny jest na rysunku **3.26**.



Rys. 3.26: Podgląd zrzutu pamięci dysku wirtualnego

3.4.2. Analiza za pomocą narzędzia Volatility

Votalility [6] to narzędzie open-source wykorzystywane do zaawansowanej analizy pamięci RAM stworzone przez organizacje The Volatility Foundation. **Rysunek 3.27** przedstawia widok aktywnych procesów wraz z ich numerami PID, znalezionych przy pomocy oprogramowania Volatility, które zostały znalezione we wcześniejszej części pracy. Widoczny jest również czas w którym poszczególne procesy powstały.

```
7076    696     svchost.exe     0x88018e13b080  3       -       0       False   2023-12-08 19:11:09
1680    696     Ransomware.wan  0x88018e3770c0  131     -       0       True    2023-12-08 19:11:44
7300    7328    tasksche.exe    0x88018e3bb0c0  7       -       0       True    2023-12-08 19:11:45
1676    7824    taskhsvc.exe    0x88018ebd7080  1       -       0       True    2023-12-08 19:13:32
4612    1676    conhost.exe     0x88018ebd4080  2       -       0       False   2023-12-08 19:13:32
7328    7468    @WanaDecryptor  0x88018e69e340  1       -       1       True    2023-12-08 19:14:00
468     1212    taskhostw.exe   0x88018c7d2080  3       -       1       False   2023-12-08 19:35:38
7100    696     svchost.exe     0x88018c899080  1       -       0       False   2023-12-08 19:35:39
```

Rys. 3.27: Procesy wylistowane przy pomocy narzędzia Volatility

Korzystając z flagi *dlllist* wraz ze znalezionymi numerami PID, wylistowane zostały biblioteki z których korzysta dany proces. Analizując wynik komendy przedstawiony na **rysunku 3.28** można zauważyć aktywny proces *svhost.exe* wykorzystujący bibliotekę *bcryptPrimitives.dll*, która wykorzystuje funkcjonalności związane z szyfrowaniem i deszyfrowaniem. Najprawdopodobniej któryś z programów stworzonych przez WannaCry wykorzystuje ten proces do szyfrowania plików.

```
7328    @WanaDecryptor  0x7ffaf95f0000  0x59000 wow64.dll       C:\Windows\System32\wow64.dll   2023-12-08 19:14:0
7328    @WanaDecryptor  0x7ffaf8ea0000  0x83000 wow64win.dll    C:\Windows\System32\wow64win.dll        2023-12-08
7328    @WanaDecryptor  0x77220000      0xa000  wow64cpu.dll    C:\Windows\System32\wow64cpu.dll        2023-12-08
468     taskhostw.exe   0x7ffaec710000  0x2b1000        iertutil.dll    C:\Windows\System32\iertutil.dll        20
7100    svchost.exe     0x7ff605580000  0x10000 svchost.exe     C:\Windows\system32\svchost.exe 2023-12-08 19:35:3
7100    svchost.exe     0x7ffafa6f0000  0x1f8000        -       -       2023-12-08 19:35:39.000000      Disabled
7100    svchost.exe     0x7ffaf9000000  0xbd000 KERNEL32.DLL    C:\Windows\SYSTEM32\KERNEL32.DLL        2023-12-08
7100    svchost.exe     0x7ffaf80f0000  0x2ce000        KERNELBASE.dll  C:\Windows\SYSTEM32\KERNELBASE.dll      20
7100    svchost.exe     0x7ffaf9ee0000  0x9c000 sechost.dll     C:\Windows\SYSTEM32\sechost.dll 2023-12-08 19:35:3
7100    svchost.exe     0x7ffaf91e0000  0x125000        RPCRT4.dll      C:\Windows\SYSTEM32\RPCRT4.dll  2023-12-08
7100    svchost.exe     0x7ffaf7eb0000  0x100000        ucrtbase.dll    C:\Windows\SYSTEM32\ucrtbase.dll        20
7100    svchost.exe     0x7ffaf9b80000  0x354000        combase.dll     C:\Windows\SYSTEM32\combase.dll 2023-12-08
7100    svchost.exe     0x7ffaf5d00000  0x12000 kernel.appcore.dll      C:\Windows\SYSTEM32\kernel.appcore.dll  20
7100    svchost.exe     0x7ffafa190000  0x9e000 msvcrt.dll      C:\Windows\system32\msvcrt.dll  2023-12-08 19:35:3
7100    svchost.exe     0x7ffaf8430000  0x82000 bcryptPrimitives.dll    C:\Windows\SYSTEM32\bcryptPrimitives.dll
```

Rys. 3.28: Biblioteki wylistowane przy pomocy narzędzia Volatility

Przy użyciu komendy *python3 vol.py -f /home/remnux/Downloads/dump.elf windows.handles | grep "7328"*, gdzie numer "7238" to PID procesu *@WanaDecrytor*, znalezione zostało odwołanie do pliku: /Device/HarddiskVolume2/Windows/Fonts/StaticCache.dat. Plik *StaticCache.dat* najprawdopodobniej zawiera czcionki, które zostały wykorzystane w oknie dialogowym pokazanym na **rysunku 3.15**. Opisana aktywność sugeruje, że odpowiedzialny za wyświetlanie okna dialogowego z okupem jest plik *@WanaDecrytor*.
Uprawnienia pliku *@WanaDecrytor*, które zostały pokazane na rysunku **3.29** pozwalają między innymi tworzenie, usuwanie i manipulowanie plikami, nadawanie uprawnień, zmianę wyświetlanego czasu, otrzymywanie powiadomień o zmianach w plikach lub ścieżkach.

```
7328    @WanaDecryptor  2       SeCreateTokenPrivilege          Create a token object
7328    @WanaDecryptor  3       SeAssignPrimaryTokenPrivilege           Replace a process-level token
7328    @WanaDecryptor  4       SeLockMemoryPrivilege           Lock pages in memory
7328    @WanaDecryptor  5       SeIncreaseQuotaPrivilege                Increase quotas
7328    @WanaDecryptor  6       SeMachineAccountPrivilege               Add workstations to the domain
7328    @WanaDecryptor  7       SeTcbPrivilege          Act as part of the operating system
7328    @WanaDecryptor  8       SeSecurityPrivilege             Manage auditing and security log
7328    @WanaDecryptor  9       SeTakeOwnershipPrivilege                Take ownership of files/objects
7328    @WanaDecryptor  10      SeLoadDriverPrivilege           Load and unload device drivers
7328    @WanaDecryptor  11      SeSystemProfilePrivilege                Profile system performance
7328    @WanaDecryptor  12      SeSystemtimePrivilege           Change the system time
7328    @WanaDecryptor  13      SeProfileSingleProcessPrivilege         Profile a single process
7328    @WanaDecryptor  14      SeIncreaseBasePriorityPrivilege         Increase scheduling priority
7328    @WanaDecryptor  15      SeCreatePagefilePrivilege               Create a pagefile
7328    @WanaDecryptor  16      SeCreatePermanentPrivilege              Create permanent shared objects
7328    @WanaDecryptor  17      SeBackupPrivilege               Backup files and directories
7328    @WanaDecryptor  18      SeRestorePrivilege              Restore files and directories
7328    @WanaDecryptor  19      SeShutdownPrivilege     Present Shut down the system
7328    @WanaDecryptor  20      SeDebugPrivilege                Debug programs
7328    @WanaDecryptor  21      SeAuditPrivilege                Generate security audits
7328    @WanaDecryptor  22      SeSystemEnvironmentPrivilege            Edit firmware environment values
7328    @WanaDecryptor  23      SeChangeNotifyPrivilege Present,Enabled,Default Receive notifications of changes to files or directories
7328    @WanaDecryptor  24      SeRemoteShutdownPrivilege               Force shutdown from a remote system
7328    @WanaDecryptor  25      SeUndockPrivilege       Present Remove computer from docking station
7328    @WanaDecryptor  26      SeSyncAgentPrivilege            Synch directory service data
7328    @WanaDecryptor  27      SeEnableDelegationPrivilege             Enable user accounts to be trusted for delegation
7328    @WanaDecryptor  28      SeManageVolumePrivilege         Manage the files on a volume
7328    @WanaDecryptor  29      SeImpersonatePrivilege          Impersonate a client after authentication
7328    @WanaDecryptor  30      SeCreateGlobalPrivilege Default Create global objects
7328    @WanaDecryptor  31      SeTrustedCredManAccessPrivilege         Access Credential Manager as a trusted caller
7328    @WanaDecryptor  32      SeRelabelPrivilege              Modify the mandatory integrity level of an object
7328    @WanaDecryptor  33      SeIncreaseWorkingSetPrivilege   Present Allocate more memory for user applications
7328    @WanaDecryptor  34      SeTimeZonePrivilege     Present Adjust the time zone of the computer's internal clock
7328    @WanaDecryptor  35      SeCreateSymbolicLinkPrivilege           Required to create a symbolic link
```

Rys. 3.29: Uprawnienia programu @WannDecryptor

Rozdział 4

Analiza kodu

W tym rozdziale za pomocą dekompilacji oraz deasemblacji częściowo omówiony zostanie kod złośliwego oprogramowania za pomocą narzędzia cutter [23] oraz znalezienie innych indykatorów i wskazówek poprzez zmianę kodu w narzędziu 32dbg.

4.1. Dekompilacja głównej części programu

Dekompilacja programu polega na zmianie widoku kodu z nisko-poziomowego na wysoko-poziomowy. Przykładem takiej zamiany może być przejście z języka assembler na język C++. Przykład dla oprogramowania WannaCry został umieszczony w **listingach 4.1** oraz **4.2**.

Listing 4.1: Kod złośliwego oprogramowania w dekompilatorze

```
/* jsdec pseudo code output */
/* C:\Users\Miki\Desktop\Ransomware.wannacry.exe.malz @ 0x40815c */
#include <stdint.h>

int32_t main (void) {
    int32_t var_64h;
    int32_t var_50h;
    int32_t var_17h;
    int32_t var_13h;
    int32_t var_fh;
    int32_t var_bh;
    int32_t var_7h;
    int32_t var_3h;
    int32_t var_1h;
    ecx = 0xe;
    esi = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com";
    edi = &var_50h;
    eax = 0;
    do {
        *(es:edi) = *(esi);
        ecx--;
        esi += 4;
        es:edi += 4;
    } while (ecx != 0);
    *(es:edi) = *(esi);
    esi++;
    es:edi++;
    eax = InternetOpenA (eax, 1, eax, eax, eax, eax, eax, eax, ax, al);
    ecx = &var_64h;
    esi = eax;
    eax = InternetOpenUrlA (esi, ecx, 0, 0, 0x84000000, 0);
```

```
    edi = eax;
    esi = imp.InternetCloseHandle;
    if (edi == 0) {
        void (*esi)() ();
        void (*esi)(uint32_t) (0);
        eax = fcn_00408090 ();
        eax = 0;
        return eax;
    }
    void (*esi)() ();
    eax = void (*esi)(uint32_t) (edi);
    eax = 0;
    return eax;
}
```

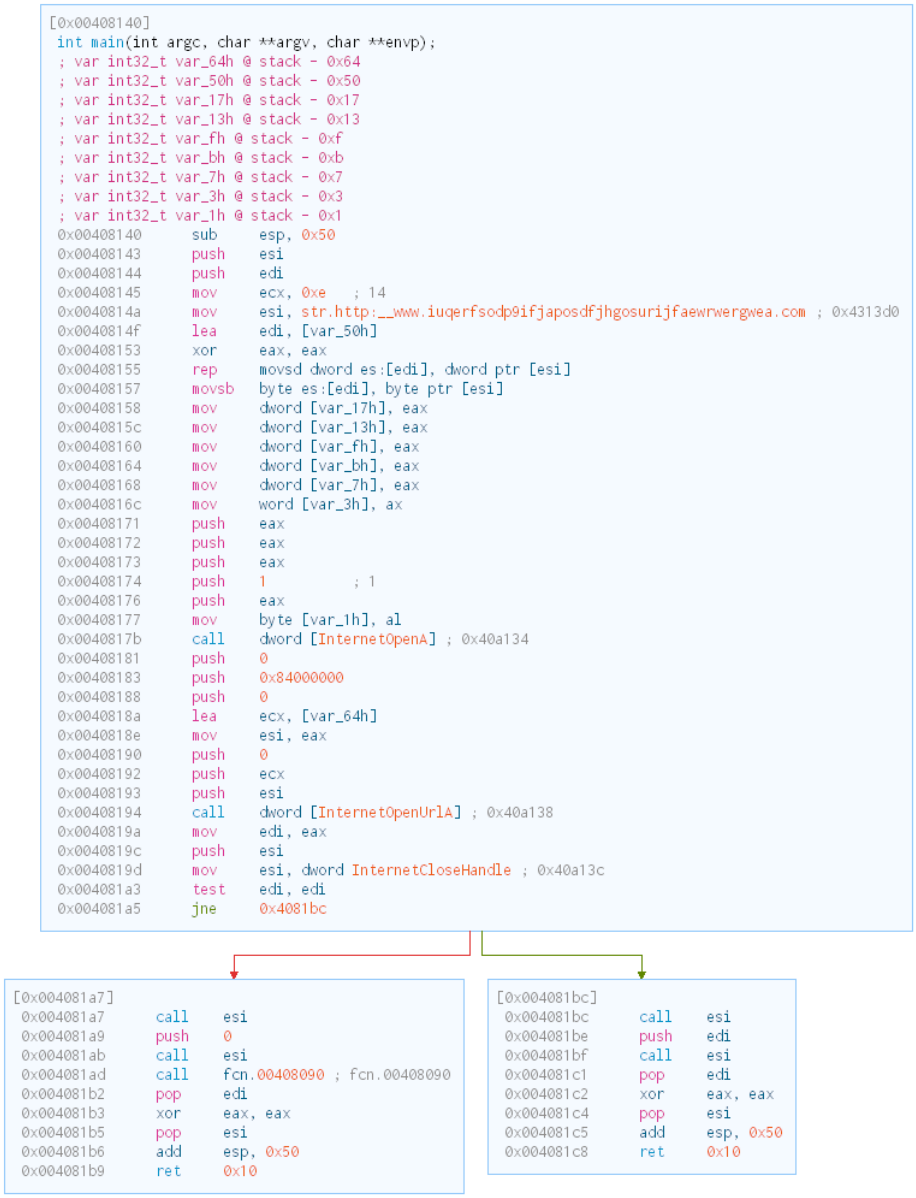
Listing 4.2: Funkcja fcn_0040890 oprogramowania w dekompilatorze

```
/* jsdec pseudo code output */
/* C:\Users\Miki\Desktop\Ransomware.wannacry.exe.malz @ 0x408090 */
#include <stdint.h>

uint32_t fcn_00408090 (void) {
    const char * var_3ch;
    const char * var_38h;
    int32_t var_34h;
    int32_t var_30h;
    int32_t var_2ch;
    const char * lpServiceStartTable;
    int32_t var_24h;
    int32_t var_20h;
    int32_t var_1ch;
    GetModuleFileNameA (0, data.0070f760, 0x104);
    eax = p_argc ();
    if (*(eax) < 2) {
        fcn_00407f20 ();
        return eax;
    }
    eax = OpenSCManagerA (0, 0, 0xf003f, edi);
    edi = eax;
    if (edi != 0) {
        eax = OpenServiceA (edi, "mssecsvc2.0", 0xf01ff, esi, ebx);
        ebx = imp.CloseServiceHandle;
        esi = eax;
        if (esi != 0) {
            fcn_00407fa0 (esi, 0x3c);
            void (*ebx)(uint32_t) (esi);
        }
        void (*ebx)(uint32_t) (edi);
    }
    eax = &lpServiceStartTable;
    StartServiceCtrlDispatcherA (eax, "mssecsvc2.0", data.00408000, 0, 0);
    return eax;
}
```

4.1.1. Statyczna analiza głównej części programu

Zakładka *Graph* narzędzia *cutter* pozwala zobaczyć różne drogi, którymi może poruszać się złośliwe oprogramowanie. Wybrana droga zależy od zwróconej wartości zapisanej w pamięci, czyli przebiegu działania programu. W celu ułatwienia analizy kodu, wykorzystana

zostanie zakładka *Graph* widoczna na **rysunku 4.1** wraz z dekompilowanym kodem umieszczonym w **listingu 4.1**. Pierwszą rzeczą, na którą warto zwrócić uwagę, jest przekazanie do pamięci za pomocą instrukcji *mov esi* znalezionego podczas analizy statycznej URL. Kolejnym krokiem jest załadowanie pozostałych argumentów do stosu pamięci, aby następnie za pomocą instrukcji *call dword [internetOpenUrlA]* wykonać połączenie internetowe. Zgodnie z dokumentacją Microsoft, API call *internetOpenURLA* jest *boolem*, czyli zwraca wartość "1" lub "0". Wartość ta zapisywana jest najpierw do *eax*, a potem do *edi*. Pozostała część kodu sprawdza za pomocą instrukcji *if* wartość ma *edi* i na podstawie wyniku wykonuje się lewa droga kodu lub prawa na grafie. Prawa część grafu wykonuje się w momencie, kiedy połączenie jest udane, czyszczone są wartości na stosie i program się zamyka. Lewa część grafu również czyści wartości na stosie, ale wykonuje się dalsza część programu poprzez odwołanie do funkcji w miejscu *call fcn.00508090*. Po odwołaniu do wspomnianej funkcji program wykorzystuje *GetModuleFileNameA*, aby pobrać zawartość ścieżki pliku, sprawdza liczbę argumentów i w zależności od wyniku, idzie do kolejnych funkcji *fcn_004070f20* lub za pomocą Windows API *OpenSCManager* otwiera menadżera usług, aby stworzyć proces *mssecsvc2.0*.



Rys. 4.1: Zakładka graf narzędzia cutter

4.2. Analiza dynamiczna kodu

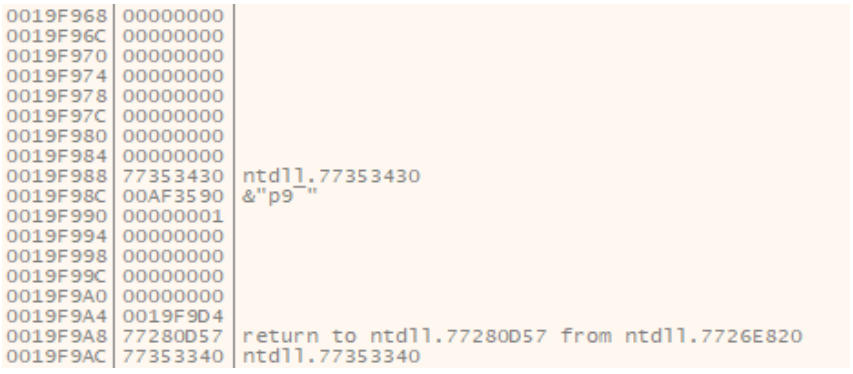
Analiza dynamiczna kodu polega na sprawdzeniu zachowania programu w trakcie jego wykonywania. Dzięki niej można odkryć inne funkcjonalności oraz doprowadzić do nich.

4.2.1. Opis działania narzędzia

Najważniejszą cechą narzędzia x32dbg jest okno "FPU" widoczny na **rysunku 4.2**, dzięki niemu jest w stanie kontrolować pamięć rejestru, oraz dzięki "EIP", podglądać kolejne wykonywane funkcję programu.

Rys. 4.2: Okno "FPU" w programiex32dbg.Ink

Stos jest to wydzielona część pamięci, która korzysta z algorytmu LIFO (Last-In-First-Out). Podgląd w programie *x32dbg* pokazany został na **rysunku 4.3**. Kontrola stosu potrzebna jest do zarządzania pamięcią. Okno w programie znajduje się w dolnej części. Pozostałe okna nie różnią się od tych widocznych w programie *cutter*.



Rys. 4.3: Okno stosu w programie x32dbg

4.2.2. Uruchomienie programu korzystając z *INetSim*

Pierwsza próba uruchomienia złośliwego oprogramowania opisana we wcześniejszym rozdziale nie powiodła się. Podczas analizy statycznej okazało się, że kiedy następuje udane połączenie do URL to program nie uruchamia się. W analizie dynamicznej pokazane zostanie, w jaki sposób osiągnąć pozytywny wynik mimo udanego połączenia. Początek programu najłatwiej znaleźć korzystając z funkcji *search* po znalezionym wcześniej URL. Przechodząc do wykonania instrukcji *test*, która zgodnie z analizą statyczną zawiera wynik wcześniejszej próby połączenia, można zauważyć wartość *EDI 00CC000C* w okienku *FPU*, co oznacza prawidłowe wykonanie połączenia. Instrukcja została zawarta na **rysunku 4.4**.

Rys. 4.4: Wartości pamięci po utworzeniu połączenia

Po przejściu do instrukcji *jne* można zauważyć zmianę wartości flagi *ZF* w oknie *FPU* z jedynki na zero, podgląd dostępny na **rysunku 4.5**. Zmieniając z powrotem wartość flagi na jedynkę, program nie przejdzie do prawej części drogi widocznej w **4.1**, tylko do lewej. W ten sposób, mimo włączonego narzędzia *INetSim* i udanego połączenia, działanie to wymusiło to aktywowanie złośliwego oprogramowania.

Rys. 4.5: Wartości pamięci po zmianie flagi

Rozdział 5

Metody obrony

Najważniejszym elementem po skończonej analizie jest odpowiednie sklasyfikowanie znalezionych dowodów, a także stworzenie odpowiednich mechanizmów obrony mających na celu rozpoznawanie i blokowanie złośliwego oprogramowania.

5.1. Techniki znalezione podczas analizy

W ramach pracy nad lepszym skategoryzowaniem i zrozumieniem znalezionych zachowań złośliwego oprogramowania, zostanie wykorzystana baza wiedzy MITRE, ATT&CK stworzona przez amerykańskie stowarzyszenie [27]. Poszczególne taktyki znalezione podczas analizy zostaną przypisane do odpowiedniej kategorii i opisane.

Tab. 5.1: Sklasyfikowane i opisane znalezione metody ataku

ID	Kategoria	Opis
T1486	Data Encrypted for Impact	Szyfrowanie danych
T1573	Encrypted Channel: Asymmetric Cryptography	Utworzenie kanału komunikacyjnego za pomocą narzędzia TOR.
11210	Exploitation of Remote Services	Rozprzestrzenienie się złośliwego oprogramowania za pomocą protokołu SMB.
T1222	Hide Artifacts: Hidden Files and Directories	Wykorzystuje komendę Attrib +h do ukycia plików.
T1016	System Network Configuration Discovery	Skan sieci w celu zainfekowania pozostałych urządzeń.

5.2. Sygnatury dla narzędzi *EDR*

Narzędzia *EDR* (Endpoint Detection and Response) to zaawansowane rozwiązania bezpieczeństwa zaprojektowane do monitorowania, wykrywania i reagowania na podejrzane działania i zagrożenia na urządzeniach końcowych, takich jak komputery, laptopy i serwery.
Najbardziej znane rozwiązania *EDR* to: CrowdStrike Falcon, SentinelOne, FireEye i Symantec. Narzędzia te wykorzystują między innymi sygnatury do badania plików. Są to specjalnie napisane pliki tekstowe zawierające zachowania złośliwego oprogramowania, a także cechy charakterystyczne. Na ich podstawie *EDR* lub antywirus podejmuje decyzje, czy taki plik jest niebezpieczny. Jedynymi z często wykorzystywanych rozwiązań są rule YARA [21].

Listing 5.1: Sygnatura Yara dla WannaCry

```
rule WannaCry_Yara {
    meta:
        author = "Mikolaj Grzempa"
        description = "Yara Rule for WannaCry"
    strings:


        $taskdl = "taskdl.exe" fullword  ascii
        $tasksche = "tasksche.exe"  fullword ascii
        $hash1 = "2
            ↪ CA2D550E603D74DEDDA03156023135B38DA3630CB014E3D00B1263358C5F00D
            ↪ " ascii
        $hash2 = "4
            ↪ A468603FDCB7A2EB5770705898CF9EF37AADE532A7964642ECD705A74794B79
            ↪ " ascii
        $hash3 = "
            ↪ ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA
            ↪ " ascii
        $URL = "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii
    condition:
        any of them
}

rule WannaCry_Yara_2 {
    meta:
        author = "Mikolaj Grzempa"
        description = "Yara Rule for WannaCry"
    strings:
        $PlikPE = {4D 5A}
        $FileName = "WannaCryptor"
        $FileName2 = "WANNACRY"
    condition:
    $PlikPE at 0 and ($FileName or $FileName2)
}
```

Opisany kod w **listingu 5.1** ma za zadanie powstrzymać oprogramowanie WannaCry przez zainfekowaniem komputera użytkownika oraz innych urządzeń w sieci.

Funkcja *WannaCry_Yara* zawiera listę nazw procesów, sum kontrolnych oraz adresów URL, z którymi złośliwe oprogramowanie próbowało się połączyć. W przypadku wykrycia przez *EDR* jakiegokolwiek z tych indykatorów, system natychmiast zatrzyma działanie podejrzanego pliku poprzez jego usunięcie lub przeniesienie do piaskownicy.

W funkcji *WannaCry_Yara_2* zdefiniowano warunek aktywacji ochrony. System reaguje, jeśli pierwsze bajty pliku rozpoczynają się od sekwencji *4D 5A*, co wskazuje na plik wykonywalny, a drugi warunek to nazwa utworzonego pliku *WannaCryptor* lub *WANNACRY*. Jeżeli obydwa warunki zostaną spełnione EDR zadziała.

Rozdział 6

Podsumowanie

Celem pracy było przedstawienie narzędzi informatyki śledczej do analizy złośliwego oprogramowania w odizolowanym środowisku z wykorzystaniem wirtualizacji. Cel został w pełni zrealizowany dla próbki WannaCry.
Za pomocą wbudowanych narzędzi programu VirtualBox skonfigurowana została wirtualna karta sieciowa, dzięki której oddzielony od sieci domowej został system Windows oraz Linux. Konfiguracja ta pozwoliła na stworzenie izolowanego środowiska, w którym w bezpieczny sposób przeprowadzone zostało badanie. W pracy, w rozdziale 2 zostały również opisane poszczególne kroki, mające na celu umożliwić uruchomienie złośliwego oprogramowania bez ograniczeń narzuconych przez antywirusa i FireWall'a.

Jako pierwsza została wykonana analiza statyczna, dzięki której odnalezione zostały podejrzane indykatory, takie jak połączenia API mogące umożliwić szyfrowanie danych, ukrywanie plików, oraz odwołanie do strony internetowej. Za pomocą narzędzi informatyki śledczej obliczona została suma kontrolna analizowanego programu, którą następnie przy użyciu dostępnych publicznych skanerów i baz danych sklasyfikowano jako znane złośliwe oprogramowanie.
Przeprowadzona analiza dynamiczna pozwoliła dokładniej zrozumieć badaną próbkę. Pojawiły się zachowania sugerujące, że opisywana próba wykorzystuje protokół SMBv1 do komunikacji z innymi urządzeniami w sieci w celu zainfekowania pozostałych urządzeń, a także na podstawie odnalezionych procesów udało się odnaleźć wskaźniki kompromitacji. W pracy zrealizowana została również podstawowa analiza wirtualnego dysku na którym znajdowało się badane złośliwe oprogramowanie. Wykorzystując narzędzia wymienione w podrozdziale numer 3.3 można w znacznej części przyspieszyć analizę złośliwego oprogramowania, co pozwoli podczas realnego incydentu na szybszą i dokładniejszą reakcję.

Statyczna analiza kodu wyjaśniła, dlaczego pierwsza próba odpalenia złośliwego oprogramowania pozornie nie zadziałała, natomiast przy pomocy analizy dynamicznej kodu, udało się w pełni uruchomić badany program. Na podstawie znalezionych zebranych danych udało się sklasyfikować taktyki wykorzystane podczas ataku, co jest kluczowym elementem w budowaniu czytelnego i zrozumiałego raportu powłamaniowego. Dzięki wyżej wymienionym czynnościom stworzone zostały reguły YARA, wykorzystywane przez narzędzia typu EDR, które są w stanie skutecznie zablokować podobne ataki.

Literatura

[1] D.W. Obsługa incydentów bezpieczeństwa związanych z malware. https://www.altkomakademia.pl/baza-wiedzy/blog/bezpieczenstwo-malware/. ost. dost. 10 marca 2022.

[2] A.F. How to extract a ram dump from a running virtualbox machine. https://andreafortuna.org/2017/06/23/how-to-extract-a-ram-dump-from-a-running-virtualbox-machine/. ost. dost. 4 grudnoa 2023.

[3] A. H. K. A.W, M.L. automated black-box malware analysis system. https://github.com/CERT-Polska/drakvuf-sandbox. ost. dost. 1 grudnia 2023.

[4] Bezpieczny blog. Co to jest hash? https://bezpieczny.blog/co-to-jest-hash/. ost. dost. 21 października 2019.

[5] Chronicle Security. Virustotal. https://www.virustotal.com. ost. dost. 20 października 2023.

[6] T. V. Foundation. Advanced memory analysis tool. https://www.volatilityfoundation.org. ost. dost. 6 grudnoa 2023.

[7] A. FZCO. Piaskownika do analizy plików. https://any.run/. ost. dost. 1 grudnia 2023.

[8] G.C. analizator pakietów. https://www.wireshark.org/. ost. dost. 3 grudnia 2023.

[9] K.B. The 12 most common types of malware. https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/. ost. dost. 28 stycznia 2023.

[10] Laboratorium Cyberbezpieczeństwa DKWOC. Proces analizy malware. https://www.wojsko-polskie.pl/woc/articles/publikacje-r/proces-analizy-malware/. ost. dost. 1 grudnia 2023.

[11] Malwarebytes. What was wannacry? | wannacry ransomware | malwarebytes. https://www.malwarebytes.com/wannacry. ost. dost. 25 października 2023.

[12] Mandiant. Flarevm. https://www.mandiant.com/resources/blog/flare-vm-the-windows-malware. ost. dost. 6 grudnoa 2022.

[13] MediaRecovery. Encase forensic. https://mediarecovery.pl/product/encase-forensic/. ost. dost. 1 grudnia 2023.

[14] Microsoft. Tcpview. https://learn.microsoft.com/pl-pl/sysinternals/downloads/tcpview. ost. dost. 7 października 2023.

[15] Microsoft Corporation. Obraz środowiska windows. https://www.microsoft.com/en-us/evalcenter/download-windows-10-enterprise. ost. dost. 21 listopada 2023.

[16] M.R. Monitor procesów. https://learn.microsoft.com/pl-pl/sysinternals/downloads/procmon. ost. dost. 1 grudnia 2023.

[17] M.R. Process monitor. https://learn.microsoft.com/en-us/sysinternals/downloads/procmon. ost. dost. 3 września 2023.

[18] mrd0x. Malicious api. https://malapi.io. ost. dost. 1 grudnia 2023.

[19] M.S, A.H. Practical malware analysis. Wydawnictwo Random House Lcc Us,2012.

[20] M.T. Automatyzacja analizy złośliwego oprogramowania. https://github.com/HuskyHacks/blue-jupyter. ost. dost. 1 grudnia 2023.

[21] N.F. Yara rules guide: Learning this malware research tool. https://www.varonis.com/blog/yara-rules. ost. dost. 1 czerwca 2023.

[22] R.D. Protokuł synchronizacji danych watermelondb. https://pushsec.pl/gpo-windows/. ost. dost. 8 stycznia 2021.

[23] Rizin. Cutter is a qt and c++ gui powered by rizin. https://cutter.re/. ost. dost. 1 grudnia 2023.

[24] S.C. Global ransomware damage costs predicted to exceed 265 billion by 2031. https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/. ost. dost. 7 lipca 2023.

[25] Sourceforge. monitor system resources, debug software and detect malware. https://processhacker.sourceforge.io. ost. dost. 4 grudnoa 2023.

[26] T.H, M.E. Symulacja podstawowych serwisów internetowych. https://www.inetsim.org/. ost. dost. 21 listopada 2020.

[27] The MITRE Corporation. Mitre attck. https://attack.mitre.org. ost. dost. 2 grudnia 2023.

[28] Virtualbox. Wirtualizacja. https://www.virtualbox.org. ost. dost. 17 listopada 2023.

[29] M. R. W.B. Floss. https://www.mandiant.com/. ost. dost. 21 czerwca 2022.

[30] Wikipedia. Virustotal. https://pl.wikipedia.org/wiki/VirusTotal. ost. dost. 20 października 2023.

[31] Zeltser Security Corp. A linux toolkit for malware analysis. https://remnux.org. ost. dost. 2 grudnia 2023.