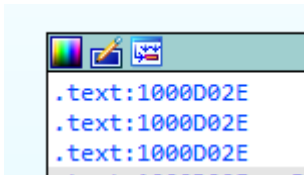


4.1

1)



.text:1000D02E

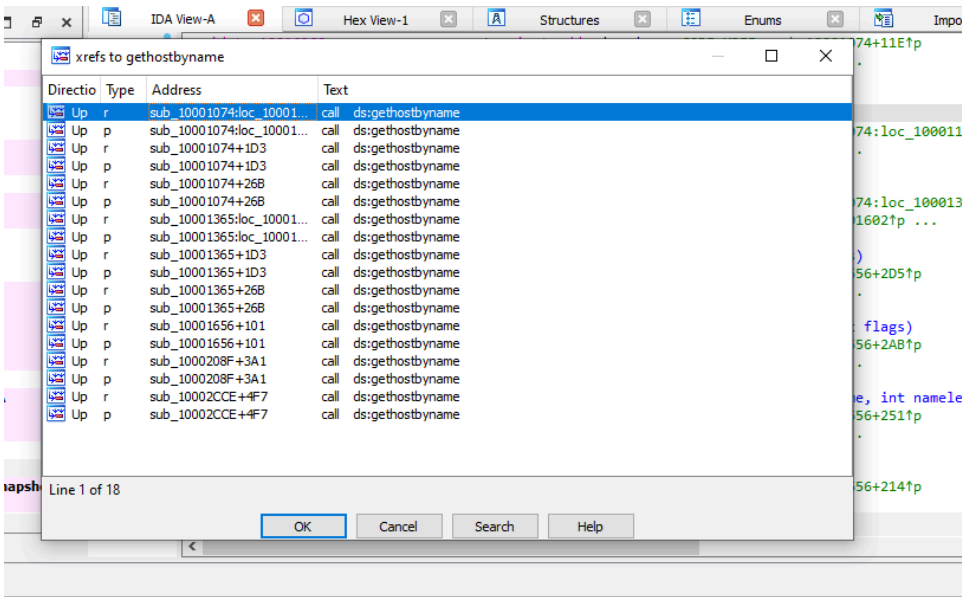
2)

Address	Ordinal	Name	Library
100163CC	52	gethostbyname	WS2_32

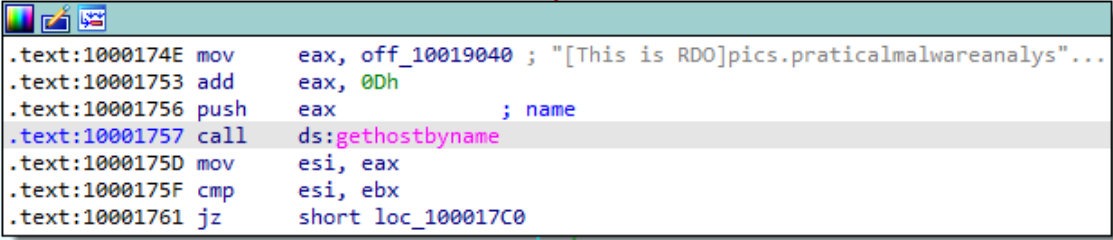
.idata:100163CC

3)

Jest wywoływany 9 razy przez 5 różnych funkcji.



4)



```
.text:1000174E mov    eax, off_10019040 ; "[This is RDO]pics.practicalmalwareanalys"...
.text:10001753 add    eax, 0Dh
.text:10001756 push   eax                ; name
.text:10001757 call   ds:gethostbyname
.text:1000175D mov    esi, eax
.text:1000175F cmp    esi, ebx
.text:10001761 jz     short loc_100017C0
```

pics.practicalmalwareanalysis.com

~~~~~  
Ładuje adres dword\_10019040 do rejestru eax, wskazując na ciąg znaków w 10019194: '[This is RDO]pics.practicalmalwareanalysis.com',0. Dodaje 0Dh do eax, przesuwając wskaźnik o 13 bajtów do przodu, aby zaczynał się od 'pics...'. Wrzuca wskaźnik na stos i wywołuje gethostbyname, przekazując argument: 'pics.practicalmalwareanalysis.com'.  
~~~~~

5)

23 zmienne

6)

Jeden - lpThreadParameter

7)

Będzie to xdoord_d:10095B34

8)

Program:

1. Przechowuje ciąg znaków "cmd.exe /c" w pamięci.
2. Czyści bufor (prawdopodobnie przygotowując go do odbierania danych).
3. Otwiera gniazdo sieciowe w trybie odbioru (receiving network socket).
4. Odbiera zdalne polecenie przez to gniazdo.
5. Wykonuje odebrane polecenie.

W skrócie: program przygotowuje się do odbierania zdalnych poleceń przez sieć i wykonuje je na lokalnym systemie.

9)

```
Up w sub_10001656+22 mov dword_1008E5C4, eax
```

Zmienna **dword_1008E5C4** jest używana przez malware do przechowywania informacji o wersji systemu operacyjnego. Na podstawie tej informacji malware decyduje, jakiego interpretera poleceń użyć do wykonywania dalszych działań.

- ~~~~~
- Funkcja `sub_10003695` pobiera informacje o wersji systemu operacyjnego przy użyciu `GetVersionExA()`.
 - Wynik tej funkcji jest porównywany z `VER_PLATFORM_WIN32_NT` (0x02)
 - Jeśli system operacyjny to Windows NT lub nowszy, funkcja ustawia rejestr `AL` na 1.
 - Wartość zwrócona przez `sub_10003695` (czyli 1 lub 0) jest przechowywana w zmiennej `dword_1008E5C4`.

Wykorzystanie zmiennej `dword_1008E5C4`:

- W kodzie znajdują się trzy referencje do zmiennej `dword_1008E5C4`, z czego jedna jest typu "w" (zapis pod adresem `10001687`).
- W dalszej części kodu, wartość `dword_1008E5C4` jest używana do podjęcia decyzji o tym, którego interpretera poleceń użyć:
 - Jeśli wartość `dword_1008E5C4` wynosi 1 (co oznacza, że system to Windows NT lub nowszy), malware używa `cmd.exe`.
 - Jeśli wartość `dword_1008E5C4` wynosi 0 (co oznacza, że system to starsza wersja Windows), malware używa `command.exe`.

~~~~~

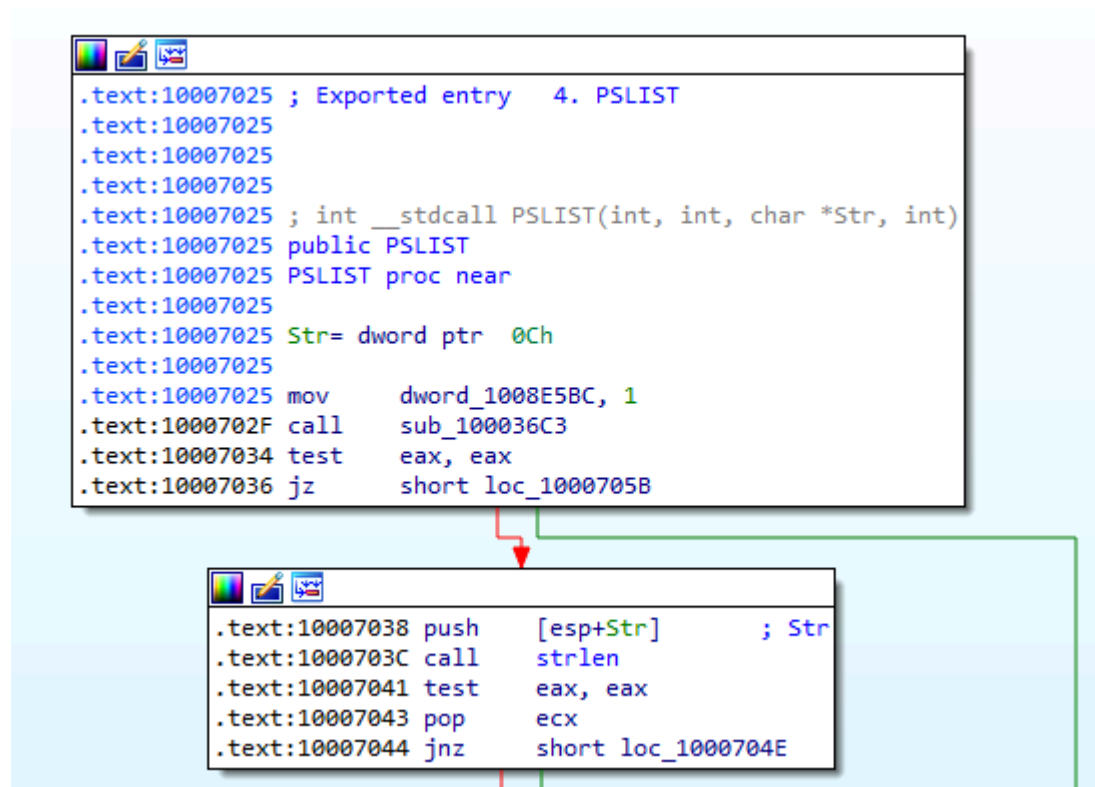
10)

Malware jest zaprogramowany do zbierania określonych danych systemowych i ich przesyłania do zdalnego serwera po udanym porównaniu łańcuchów.

Jeśli `memcmp` zwróci 0 podczas porównywania łańcuchów z "robotwork", program:

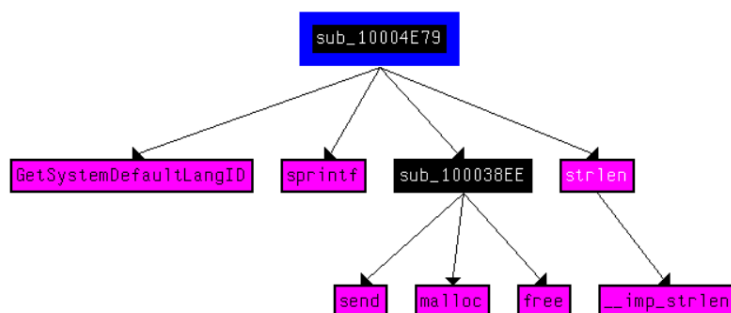
- Odczyta wartość `WorkTimes` z rejestru.
- Przeformatuje ją w ciąg znaków o określonym formacie.
- Wyśle sformatowany ciąg znaków przez sieć za pomocą socketu TCP/IPv4.

11)



- Tworzy zrzut uruchomionych procesów (CreateToolhelp32Snapshot).
- Wysyła nazwę procesu i PID przez socket.
- Otwiera proces (OpenProcess).
- Pobiera moduły procesu (EnumProcessModules).
- Wysyła PID, ścieżkę exe i liczbę wątków przez socket.
- Zapisuje informacje o procesach do pliku xinstall.dll.
- Przechodzi do następnego procesu i powtarza kroki.

12)



Na podstawie dostarczonego grafu, funkcja sub\_10004E79 może wywołać następujące funkcje API:

1. GetSystemDefaultLangID
2. sprintf
3. send
4. malloc
5. free
6. strlen
7. \_\_imp\_strlen (implementacja strlen)

Bazując na tych funkcjach API, sub\_10004E79 prawdopodobnie wykonuje następujące działania:

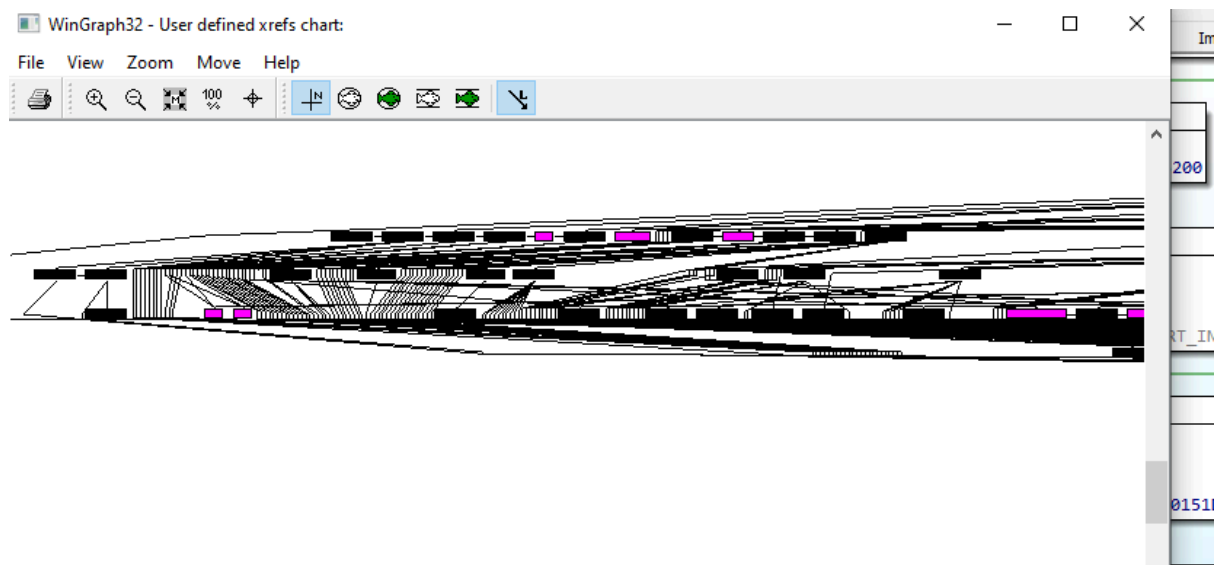
1. Pobiera domyślny identyfikator języka systemu (GetSystemDefaultLangID).
2. Formatuje ciąg znaków przy użyciu sprintf.
3. Wysyła dane przez sieć (send).
4. Alokuje i zwalnia pamięć (malloc i free).
5. Oblicza długość ciągów znaków (strlen i \_\_imp\_strlen).

Możliwa nazwa tej funkcji to:

**SendSystemLanguageInfo()**

13)

Wywołuje 4 funkcje, a dla głębokości 2 - zdecydowanie więcej.



14)

```
.text:10001358          call     ds:Sleep          ; Sleep 30s.  
30 sekund
```

15)

1, 2, 6

16)

Tak, w pliku występuje funkcja wykorzystująca instrukcję in do wykrywania VMware.

Dodatkowe znaki, takie jak nietypowa instrukcja **vpcext** i ciąg wskazujący na wykrywanie maszyn wirtualnych, potwierdzają próbę wykrywania środowisk wirtualnych.

17)

Dane pod adresem 1001D988 wyglądają na zakodowane lub zaszyfrowane.

Są serią znaków drukowalnych, przerywaną nieczytelnymi znakami.

```
data:1001D986 - 0  
data:1001D987 - 0  
data:1001D988 - a1UUU7461Yu2u1  
db '-1:',27h,'u<&u!=<6u746'>1:',27h,'yu&',27h,'<;2u106:101u3:',27h,'u',5,27h,'46<6'  
db '49u',18h,'49"4',27h,'6u',14h,'49,&6u',19h,'47u1|dgfa'  
data:1001D998 - 0  
data:1001D999 - 0
```

## 4.2

1)

Główna funkcja znajduje się pod adresem 00401040. Funkcja następnie wywołuje funkcję pod adresem 00401000, która zawiera konstrukcję warunkową If. Ta z kolei wywołuje funkcję **InternetGetConnectedState()**.

```

push    offset aSuccessInterne ; "Success: Internet Connection\n"
call    sub_40105F
add     esp, 4
mov     eax, 1
jmp     short loc_40103A

```

```

loc_401028:
push    offset aError11NoInter ; "Error 1.1: No Internet\n"
call    sub_40105F
add     esp, 4
xor     eax, eax

```

W zależności od jakości połączenia internetowego wywoływana jest funkcja z sub\_40105F.

2)

Jest to najprawdopodobniej funkcja **printf**. Program IDA nie jest już tak skuteczny jak w pierwszym zadaniu, dlatego trudniej jest znaleźć rzetelne informacje.

3)

| Offset | Name         | Func. Count | Bound? | OriginalFirstThun | TimeDateStamp | Forwarder |
|--------|--------------|-------------|--------|-------------------|---------------|-----------|
| 64C4   | WININET.dll  | 1           | FALSE  | 65B0              | 0             | 0         |
| 64D8   | KERNEL32.dll | 43          | FALSE  | 6500              | 0             | 0         |

| KERNEL32.dll [ 43 entries ] |                   |         |                |       |           |      |
|-----------------------------|-------------------|---------|----------------|-------|-----------|------|
| Call via                    | Name              | Ordinal | Original Thunk | Thunk | Forwarder | Hint |
| 6000                        | VirtualFree       | -       | 6780           | 6780  | -         | 2BF  |
| 6004                        | CloseHandle       | -       | 68C0           | 68C0  | -         | 1B   |
| 6008                        | GetCommandLi...   | -       | 65E0           | 65E0  | -         | CA   |
| 600C                        | GetVersion        | -       | 65F2           | 65F2  | -         | 174  |
| 6010                        | ExitProcess       | -       | 6600           | 6600  | -         | 7D   |
| 6014                        | TerminateProcess  | -       | 660E           | 660E  | -         | 29E  |
| 6018                        | GetCurrentProc... | -       | 6622           | 6622  | -         | F7   |

Analizując importy w PE Bear i poprzednie podpunkty:

- Ten program sprawdza połączenie z internetem.
- Sprawdza, czy system jest 32- lub 64-bitowy.
- Otwiera uchwyt pliku.
- Zapisuje znaki do pliku.

|      |               |   |
|------|---------------|---|
| 6050 | GetVersionExA | - |
| 6054 | HeapDestroy   | - |
| 6058 | HeapCreate    | - |
| 605C | HeapFree      | - |

Na podstawie importowanych modułów można też przypuszczać, że program operuje na poziomie niskopoziomowym, prawdopodobnie wykonuje operacje na pamięci, wcześniej wspomnianych plikach, procesach i stercie (heap).

## 4.3

1)

Jest to ta sama funkcja co w poprzednim zadaniu. Sprawdza połączenie internetowe. Zwraca wartość true oraz zwraca wartość false w zależności od połączenia.

2)



```
printf proc near
```

to będzie po prostu printf()

3)

Ten program nawiązuje połączenie z witryną **practicalmalwareanalysis.com**, aby pobrać plik cc.htm. Czyta jego zawartość partiami po 200 bajtów. Następnie parsuje pobraną zawartość, szukając komentarzy zaczynających się od '<!--'. Jeśli znajdzie taki komentarz, kolejny znak po komentarzu jest interpretowany jako polecenie do wykonania przez program.

```
|push    offset szUrl    ; "http://www.practicalmalwareanalysis.com"..
```

4)

- <http://www.practicalmalwareanalysis.com/cc.htm> (**URL**)
- Internet Explorer 7.5/pma (**User-agent**)



5)

Ten program wysyła żądanie GET do strony kontrolnej z użyciem specjalnego identyfikatora user agent, aby się zidentyfikować. Następnie parsuje pierwsze 512 bajtów odpowiedzi, szukając komentarza HTML. Jeśli znajdzie komentarz, pobiera z niego zakodowane polecenie.

Może zostać wykorzystany do sprawdzenia stanu połączenia zaatakowanego systemu, a także do uzyskania poleceń z serwera (na podstawie podanego adresu URL i znajdującego się w nim) i wyświetlania ich na konsoli.