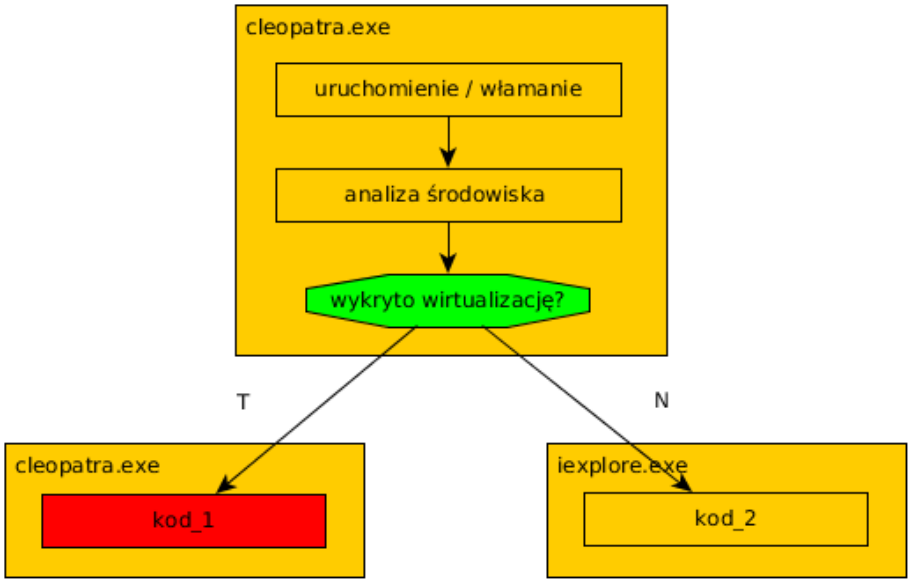
Raport – analiza oprogramowania Cleopatra: część II

W pierwszej części raportu przedstawiono analizę dropera oprogramowania nazwanego roboczo Cleopatra. Wspomniano w nim, że w zależności od wyników analizy swojego środowiska uruchomieniowego, próbka podejmuje różne decyzje dotyczące dalszego działania. Fundamentalną decyzją jest ta o wstrzyknięciu jednego z dwóch przygotowanych kodów. W niniejszym opracowaniu przeanalizowano scenariusz, w którym dropper po wykryciu wirtualizacji wstrzykuje kod oznaczony jako kod_1.

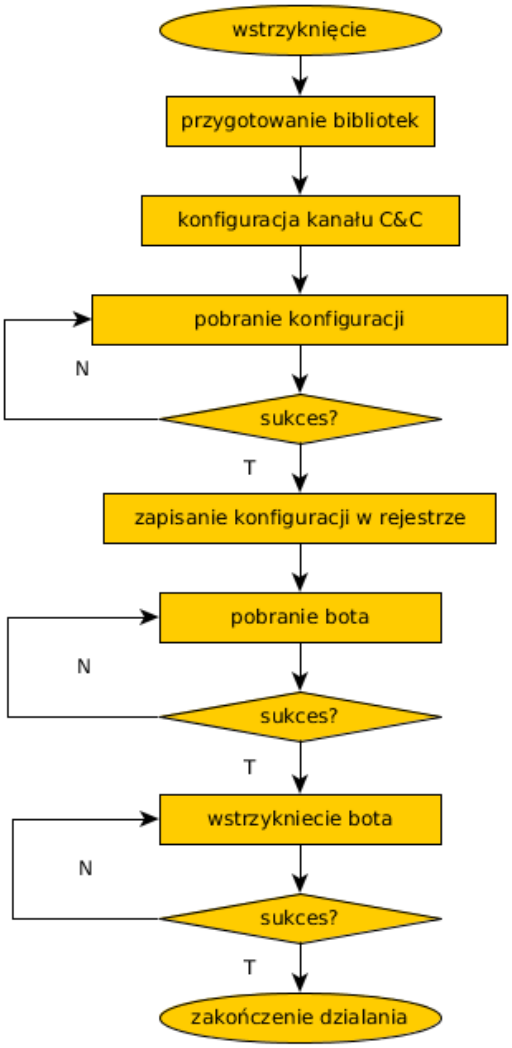


Sumy kontrolne próbki:

Md5	d07f1000d60f2a06eed08f30e42d73ab
Sha256	01e4365a53ecbf0d80e5b91df51219e0a40c6e3a914ef36151056a20e3be87d1
Sha512	abd3dcb6fc85553c80afc5f8b6f8615767c995e662f61f467e6a1171d9d824466009 1aff615cd454149af48e73c3dcc5179235c8421dd54bc491466ca816ba08

Działanie

Po podjęciu decyzji o wstrzyknięciu kodu 1 Cleopatra uruchamia droppera po raz drugi, jednak tym razem jest on zatrzymany (uruchomienie z flagą CREATE_SUSPENDED). Jego pamięć jest nadpisywana nowym kodem, a główny wątek procesu jest modyfikowany w taki sposób, by rozpoczął jego wykonywanie. Można powiedzieć, że oryginalna „powłoka" droppera jest nienaruszna, ale jego „wnętrze" jest zmienione. Jego nowym zadaniem jest nawiązać komunikację z kanałem C&C i wykonać instalację zadanych programów.



Rysunek 1: Ogólny zarys operacji kodu kod_1

Aby wykonać swoje zadania, w pierwszej kolejności Cleopatra ładuje niezbędne biblioteki (shellwapi, shell32). Następnie przygotowywana jest biblioteka WinInet oraz wykonywane są połączenia do zdalnych węzłów w celu pobrania potrzebnych danych. Węzły są opisane za pomocą pięciu adresów IPv4, które są zakodowane i na stałe zapisane w próbce (więcej nt. kodowania łańcuchów znakowych w dalszej części raportu). Jeśli próba połączenia się z wybranym adresem zakończy się niepowodzeniem, jest podejmowana kolejna próba połączenia się z kolejnym adresem z listy.

Kanał C&C

Cleopatra korzysta ze scentralizowanego kanału C&C opartego na protokole https. Do nawiązania połączenia wykorzystywane jest 5 adresów IP rozmieszczonych w różnych miejscach na świecie.

Adres	**Kraj**
92.xxx.xxx.59	Francja
176.xxx.xxx.53	Francja
213.xxx.xxx.171	Czechy
210.xxx.xxx.63	Wietnam
80.xxx.xxx.222	Włochy

Oparcie kanału na adresach IP w niektórych przypadkach komplikuje blokowanie złośliwego ruchu. W odróżnieniu od stosowania domen, które można w większości przypadków określić jako w całości złośliwe, blokowanie ruchu do węzła w zakresie całego adresu IP może spowodować zablokowanie ruchu do legalnie działających usług. W wyniku tego w niektórych przypadkach blokowanie ruchu jest nie tylko skomplikowane, ale w niektórych krajach może być nielegalne.

Z kolei certyfikat (własnoręcznie podpisany) prezentowany przez serwery C&C i wykorzystywany przez protokół SSL został wystawiony na organizację `Omeelage Tathagar SASU` z siedzibą w New Delhi, Indie. Został on zarejestrowany na czarnej liście certyfikatów portalu Abuse.ch.

O ile zebrane dane dot. Kanały C&C mogą posłużyć do korelowania z danymi z innych kampanii, należy pamiętać, że nie mogą stanowić żadnej konstruktywnej postawy do wskazania rzeczywistostego pochodzenia kampanii. Są one zwyczajnie zbyt łątwe do podrobienia.

Struktura danych udostępniana przez kanał C&C jest prosta. Pierwsze cztery bajty zawierają skrót datagramu. Reszta stanowi zaszyfrowane dane. Po każdorazowym pobraniu nowych danych próbka wyznacza skrót szyfrogramu i porównuje go ze skrótem zawartym w nagłówku. Jeśli skróty nie są równe, dane są odrzucane.

Rysunek 2: Struktura datagramu Cleopatra

Po pozytywnym zweryfikowaniu ingralności szyfrogramu, następuje rozszyfrowanie danych. Szczegółowe omówienie algorytmu szyfrującego znajduje się w dalszej części raportu.

Operacje

Pierwsza część pobranych danych (zasób /list) to konfiguracja Cleopatry, która po uprzednim zaszyfrowaniu jest zapisywana i przechowywana w rejestrze systemowym.

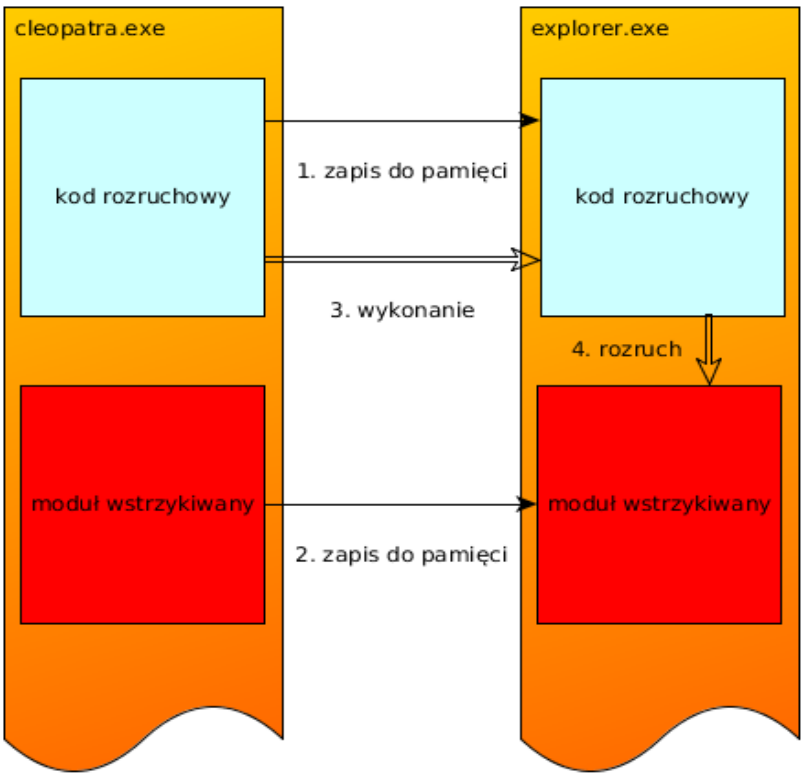
Następnie pobierany jest zasób /bot. Dane te stanowią w zasadzie dowolny moduł PE, który będzie wstrzykiwany do procesu explorer.exe. Po wykonaniu tego kroku następuje przeglądanie działających w systemie procesów w poszukiwaniu procesu explorer.exe. Odpowiednia funkcja dla każdego z zarejestrowanych procesów pobiera jego nazwę i wylicza sumę kontrolną. Suma ta jest porównywana z zapisaną w próbce sumą 0xBE037055, która odpowiada właśnie nazwie explorer.exe.

```
.exe:00403BDF mov     esi, eax
.exe:00403BE1 mov     eax, [esp+0A0h]
.exe:00403BE8 call    calculate_checksum
.exe:00403BED mov     [esp+0A0h], ebp
.exe:00403BF4 cmp     esi, 0BE037055h
.exe:00403BFA mov     [esp+0A4h], ebp
.exe:00403C01 jz      short continue_injection
.exe:00403C03 inc     edi
.exe:00403C04 cmp     edi, [esp+0ACh]
.exe:00403C0B jl      short loc_403B9A
.exe:00403C0D jmp     loc_403EE5
.exe:00403C12 ; ---------------------------------------
```

Rysunek 3: Weryfikacja nazwy procesu

Po dokonaniu wyboru procesu, wykonywane jest wstrzykiwanie kodu. Większość złośliwych progrmów do wstrzyknięcia przygotowuje konkretny kod, w którym wykonywane jest wiele przekształceń w oparciu o rozkład fragmentów pamięci w procesie procesu-gospodarza. Implementacja wstrzykiwania przez Cleopatrę jest o tyle ciekawa, że umożliwia wstrzyknięcie dowolnego programu, bez względu na wykorzystywane biblioteki i wywołania.

Pomysł opiera się na wstrzykiwaniu dwóch fragmentów kodu, z których pierwszy stanowi bot, a drugi – specjalny kod rozruchowy. Kod ten wykonuje w procesie explorer.exe część zadań loadera procesów systemu Windows. Odnajduje importowane przez program biblioteki i funkcje i ładuje je do działającego procesu. Określa również na podstawie nagłówków PE punkt wejściowy programu (ang. Entry point) i przekazuje tam sterowanie jednego z wątków procesu explorer.exe.



Rysunek 4: Wstrzykiwanie modułu PE

Po tych wszystkich drobnych modyfikacjach docelowy kod umieszczony w procesie gospodarza jest już gotowy do działania. Cleopatra tworzy w procesie nowy wątek, ktory rozpocznie wykonywanie pobranego programu.

Dekodowanie łańcuchów znakowych

Wykorzystywane w próbce łańcuchy znakowe jak zazwyczaj bywa w przypadku próbek złośliwego oprogramowania, nie są przechowywane w postaci jawnej.

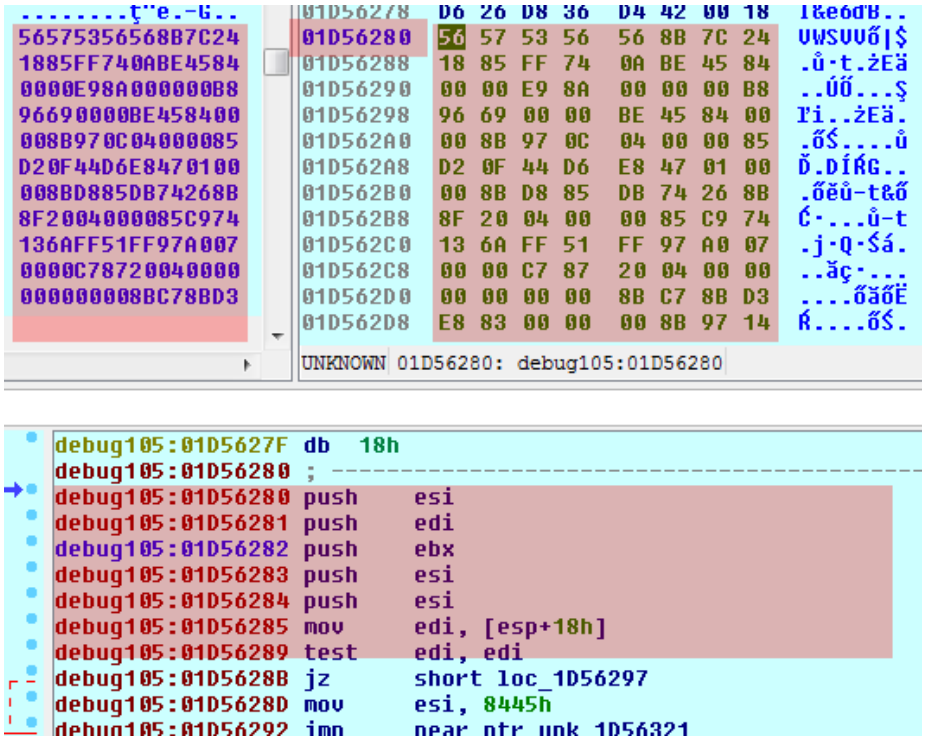
Zamiast zwykłych łańcuchów są wykorzystywane specjalne, zaszyfrowane algorytmem RC4 obiekty na stałe zapisane w próbce. Każdy z tych obiektów posiada swój identyfikator. Kiedy powstaje konieczność skorzystania z łańcucha znakowego, program odnosi się do obiektu o odpowienim identyfikatorze. Obiekt jest rozszyfrowywany i udostępniany jest łańcuch w postaci jawnej.

```
01D4D9A8  00 00 00 00 00 00 00 00  EB 27 D8 0A C4 47 00 18  ........ë'Ø.ÄG..
01D4D9B8  00 00 00 00 20 00 00 00  00 00 00 00 01 00 00 00  ....·...........
01D4D9C8  00 00 00 01 03 00 00 00  AB AB AB AB AB AB AB AB  ........««««««««
01D4D9D8  00 00 00 00 00 00 00 00  EE 22 D8 0A D1 47 00 18  ........î"Ø.ÐG..
01D4D9E8  3C 63 66 67 20 6E 65 74  3D 22 25 64 22 3E 3C 73  <cfg·net="%d"><s
01D4D9F8  74 61 72 74 75 70 3E 25  73 3C 2F 73 74 00 00 00  tartup>%s</st...
01D4DA08  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
01D4DA18  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
01D4DA28  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
01D4DA38  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
01D4DA48  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Rysunek 5: Dekodowany łańcuch

Dzięki temu rozwiązaniu zawarte w próbce łańcuchy nie są dostępne w analizie statycznej ani w trakcie zwykłego działania aż do momentu, w którym sięga po nie próbka.

Ciekawostkę stanowi fakt, różne fragmenty kodu, z których korzysta próbka, również są przechowywane w postaci zaszyfrowanych łańcuchów znakowych. Po odszyfrowaniu tych łańcuchów otrzymujemy liczbowy zapis kolejnych bajtów kodu, które są za pomocą specjalnej funkcji konwertowane do swojej binarnej postaci i wykonywane.

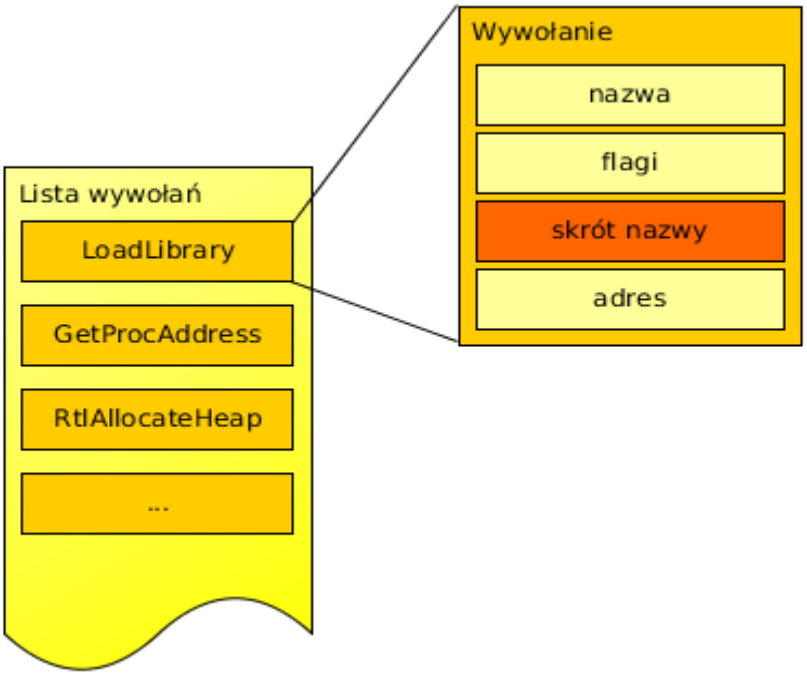


Rysunek 6: Konwersja kodu z postaci ASCII do binarnej

Ukrywanie wywołań bibliotecznych

Podobnie, jak łancuchy znakowe, próbka ukrywa wykorzystywane wywołania biblioteczne. W wyjściowej postaci próbka nie zawiera żadnych informacji na temat nazw wykorzystywanych wywołań. Nie wykaże ich statyczna analiza importowanych wywołań pliku ani poszukiwanie łańcuchów znakowych. Jednak zamiast szyfrowania łańcuchów nazw wywołań bibliotecznych, zastosowano inne rozwiązanie.

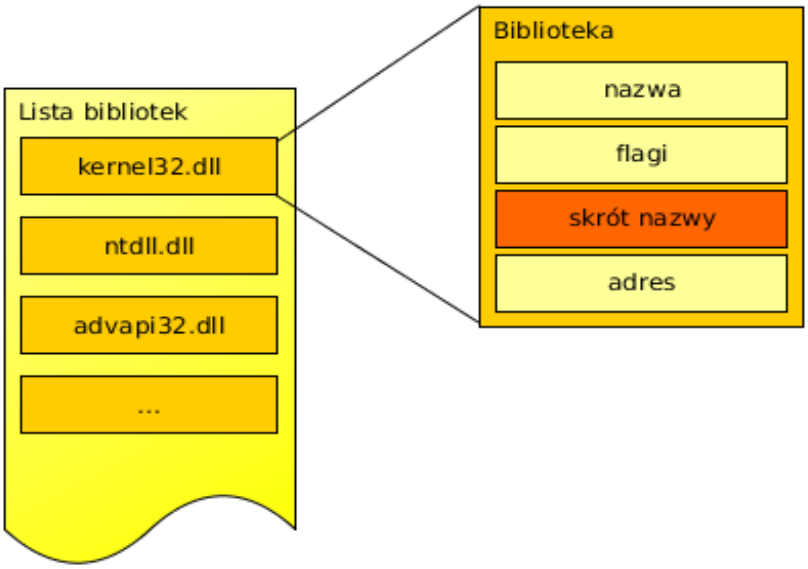
Każde z wywołań identyfikowane jest za pomocą pary skrótów o długości czterech bajtów. Pierwszy odpowiada za nazwę biblioteki, a drugi – za nazwę funkcji. Na swoje potrzeby Cleopatra utrzymuje listę wywołań i każde wywołanie jest do niej dodawane dopiero w momencie pierwszego użycia.



Rysunek 7: Lista wywołań

Tak więc, w przypadku zaistnienia potrzeby skorzystania z konkretnego wywołania, zamiast pary nazw biblioteki i wywołania, podawana jest właśnie para skrótów tych nazw. Program przystępuje do przeglądania istniejącej listy zarejestrowanych wywołań w poszukiwaniu zapisu ze skrótem zgodnym z zadanym. Jeśli zostanie on odnaleziony, wywołanie jest wykonywane od razu. Jeśli nie, program najpierw musi dodać je do listy.

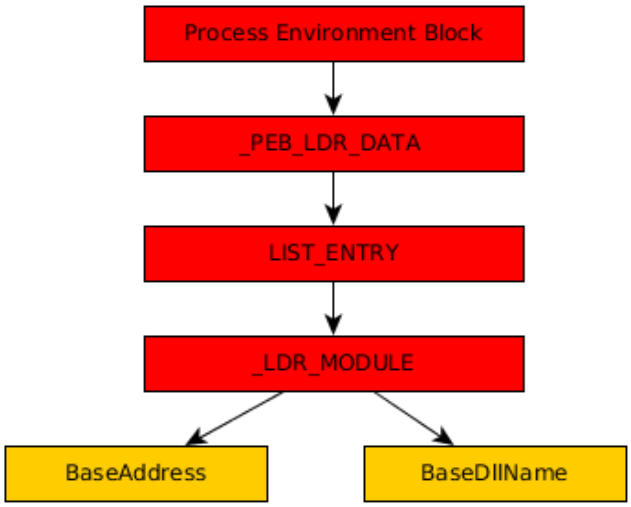
Dodawanie niezarejestrowanego wywołania do listy odbywa się w następujący sposób. Najpierw wyszukiwana jest biblioteka, której nazwa odpowiada zadanemu skrótowi nazwy biblioteki. Program przechowuje te nazwy na liście bardzo podobnej do listy wywołań.



Rysunek 8: Lista bibliotek

W przypadku braku wymaganej biblioteki na liście, program rozpoczyna jej wyszukiwanie w pamięci.

W pierwszym kroku sięga po strukturę systemową opisującą proces złośliwego programu, mianowicie strukturę Process Environment Block, który w każdym procesie jest przechowywany w pamięci pod adresem FS:[0x30].

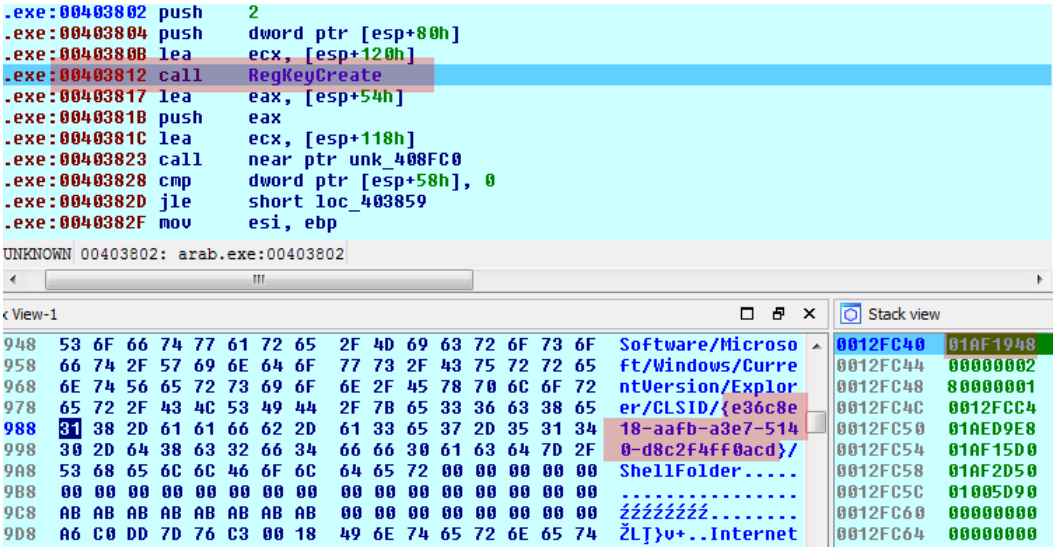


W strukturze tej znajduje się wykaz załadowanych przez proces do pamięci bibliotek. Wykorzystując wspomniane struktury, odnajduje wszystkie załadowane biblioteki i pobiera ich nazwy. Wylicza skrót nazwy każdej z bibliotek i porównuje ze skrótem zadanym w wywołaniu. Jeśli skrót się zgadza, struktura opisująca bibliotekę (jej nazwa, flagi, skrót, oraz adres) jest dołączana do listy.

Po zidentyfikowaniu poszukiwanej biblioteki przeglądane są nazwy wszystkich eksportowanych przez nią wywołań. Dla każdej z tych nazw jest wyznaczany skrót i porównywany ze skrótem zadanym w wywołaniu. W przypadku zgodności skrótów – wywołanie jest rejestrowane na liście.

Generowanie nazw plików i kluczy

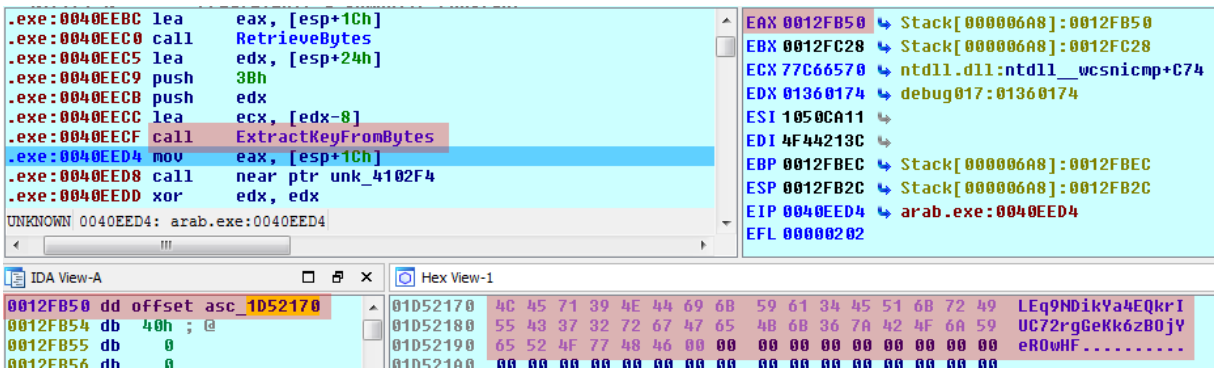
Wszelkie nazwy plików, mutexów oraz kluczy rejestrów wykorzystywanych przez próbkę generowane są w sposób pseudolosowy. Ziarnem algorytmu jest data zainstalowania systemu Windows przechowywana w kluczu.



Rysunek 9: Przykładowa nazwa klucza rejestru wygenerowana przez Cleopatrę

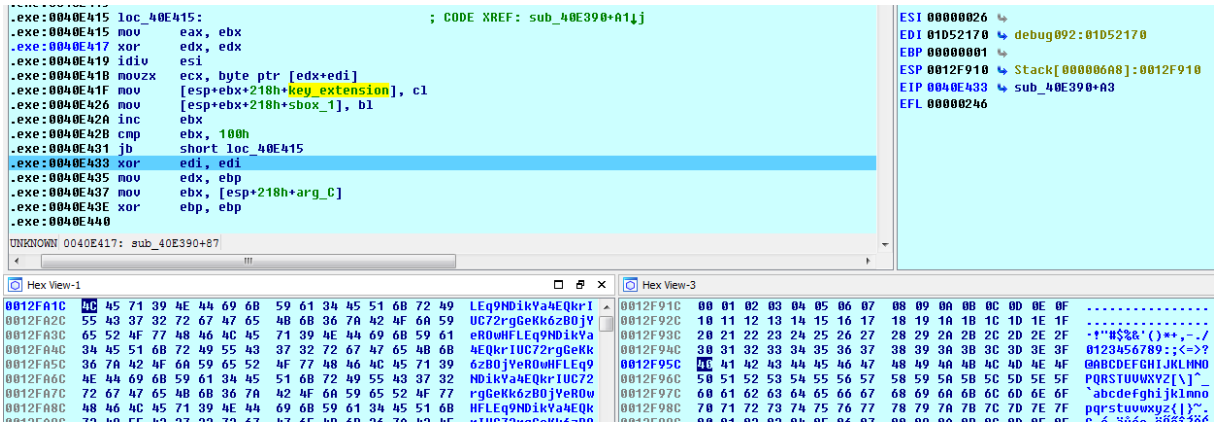
Algorytmy szyfrowania

Do szyfrowania i deszyfrowania danych odczytywanych i zapisywanych Cleopatra wykorzystuje algorytm oparty o RC4. Do zapisywania i odczytywania wykorzystywane są dwa różne klucze.



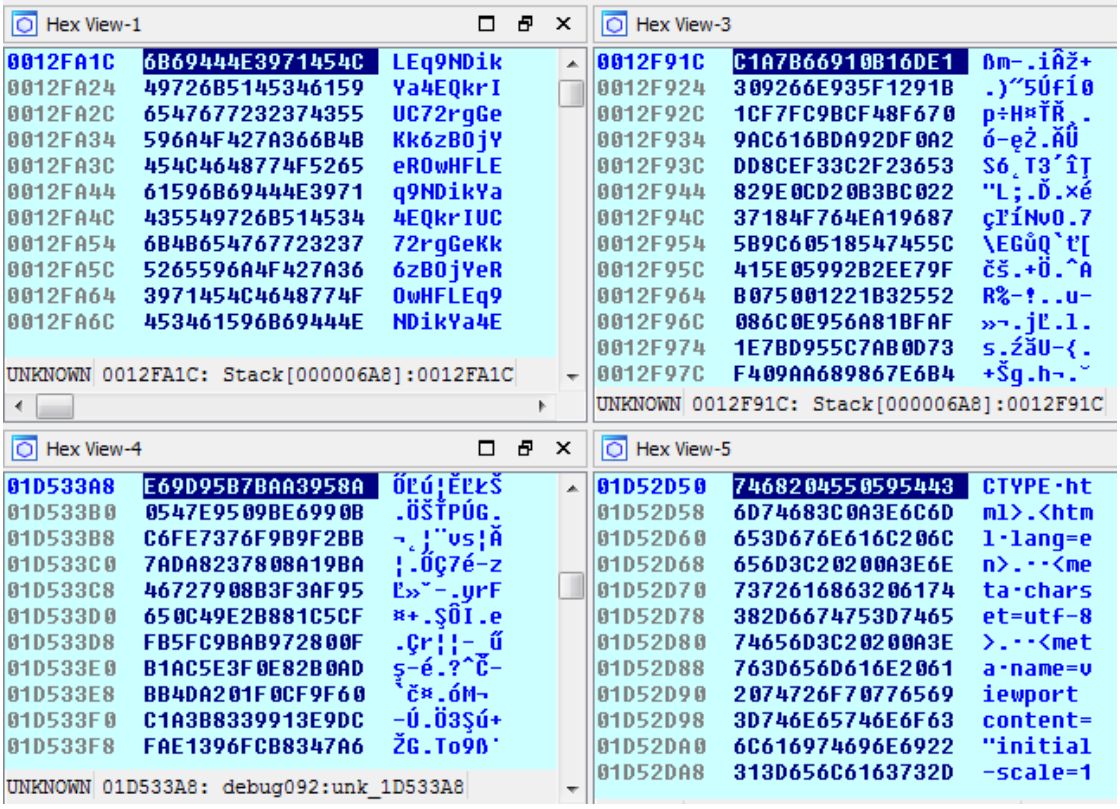
Rysunek 10: Odzyskany z zaszyfrowanego obiektu klucz

Po wyodrębnieniu odpowiedniego klucza jest on przedłużany do długości 256 bajtów. Podczas tego procesu tworzona jest również pomocnicza tablica bajtów, wypeniona wartościami od 1 do 256, które w miara inicjowania i działania szyfru jest mieszana.



Rysunek 11: Proces deszyfrowania RC4

Strumień szyfrujących bajtów jest poddawany działaniu operacji xor z tekstem jawnym lub szyfrogramem.



Rysunek 12: Deszyfrowanie danych. Klucz (lewo, na górze), mieszana tablica (prawo, na górze), szyfrogram (lewo, na dole), tekst jawny (prawo, na dole)

Podsumowanie

Najłatwiejszą metodą wstrzykiwania kodu jest jego przygotowanie w procesie droppera, który wstrzykuje już gotowy do działania kod, z określonymi adresami wymganych do działania bibliotek i wywołań. Metoda stosowana w analizowanej próbce, wstrzykuje cały plik PE. Dzięki temu umożliwia zupełną niezależność autora Cleopatry i autora wstrzykiwanej próbki i może służyć do wstrzyknięcia dowolnego programu. Przykładem znanego z przeszłosci złośliwego programu o zbliżonej funkcjonalności jest Piptea, który służył do instalowania m.in. oprogramowania Virut.

Analizowana próbka w analizowanym wariancie (wstrzyknięcie kodu kod_1) prawdopodobnie realizuje funkcje platformy instalującej inne złośliwe oprogramowanie na zasadzie pay-per-install (PPP). Świadczy o tym ograniczona funkcjonalność oraz specyficzna metoda wstrzykiwania pobieranego kodu.

Poziom zagrożenia w sieci Orange został określony jako niski. Klienci sieci Orange Polska są chronieni przed zagrożeniem Cleopatra, ponieważ złośliwy ruch do serwerów C&C jest blokowany. Niezależnie od środków, które stosujemy w celu ochrony naszych klientów, CERT Orange Polska rekomenduje zainstalowanie i utrzymywanie aktualnych rozwiązań antywirusowych, które pozwolą uchronić system przed infekcjami podobnym oprogramowaniem w przyszłości.