Orange Polska S.A.
Bezpieczeństwo Systemów Teleinformatycznych / Wydział Operacji Bezpieczeństwa

Warszawa, 21.02.2016

Analiza złośliwego oprogramowania

Cleopatra

Poniższy raport opisuje szczegółowo analizę próbki, uzyskanej w trakcie analizy powłamaniowej w systemie jednego z klientów Orange Polska. Był to moduł PE32 o następujących skrótach kryptograficznych:

Md5	d07f1000d60f2a06eed08f30e42d73ab
Sha256	01e4365a53ecbf0d80e5b91df51219e0a40c6e3a914ef36151056a20e3be87d1
Sha512	abd3dcb6fc85553c80afc5f8b6f8615767c995e662f61f467e6a1171d9d8244660091a ff615cd454149af48e73c3dcc5179235c8421dd54bc491466ca816ba08

Do próbki nie udało się dopasować żadnej sygnatury przyporządkowującej ją do jednej ze znanych gatunków złośliwego oprogramowania, więc w niniejszym raporcie posłużono się roboczą nazwą Cleopatra. Próbka została dostarczona w październiku 2016 roku w ramach kampanii phishingowej, w której nadawca podszywał się pod firmę świadczącą usługi kurierskie. W mailu ofiara była namawiana do otworzenia linku rzekomo zawierającego oprogramowanie, które umożliwi mu odebranie niedostarczonej paczki.

Failed Delivery for Package # 415767662

usps@transitsystems.com
Wysłano: Pn 2016-11-14 19:43
Do: [redacted]

We tried but failed to deliver your package again today, because no one was present at the destination address.
On the delivery day, there must be someone present at the destination address to receive the parcel.

Shipping type: Priority 1day
Box size: Large Flat Rate box
Date : Nov 14th 2016
Delivery Notification : e-mail sent

To reschedule the parcel delivery, visit our nearest office, with a printed copy of the Delivery Notice Card.

An electronic copy of the Delivery Notice Card, in Microsoft Word format, can be downloaded from our website :
https://tools.usps.com/go/TrackConfirmAction?action=download&in voice=73118704512

The tracking number can be found on the Delivery Notice Card and can be used to track your parcel:
https://t ools.usps.com/go/TrackConfirmAction_input

Thanks for shipping with us

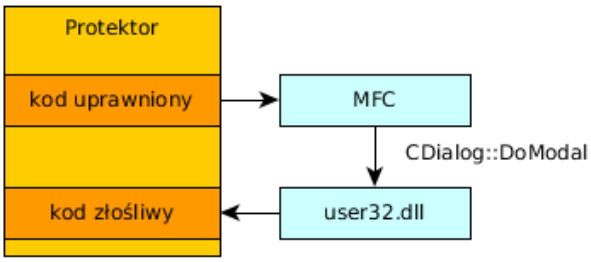
© 2016 United States Postal Service

1. Organizacja próbki

Kod analizowanej próbki w kontekście jego funkcji można podzielić na dwie części organizacyjne – protektor i dropper. Funkcją protektora jest zapobiec zaklasyfikowanie próbki jako złośliwe oprogramowanie przez zainstalowane w systemie oprogramowanie defensywne. Dodatkowo protektory utrudniają dokładną analizę funkcjonalności oprogramowania również na etapie drobiazgowej analizy próbek już zaklasyfikowanych jako złośliwe.

2. Protektor

Kod protektora nie jest wywoływany bezpośrednio, lecz przy wykorzystaniu wywołań bibliotecznych biblioteki MFC. W uproszczeniu, jest to metoda ukrywania złośliwych funkcji wykorzystująca założenie, że złośliwy kod nie może być wywołany przez uprawniony kod, co jest nieprawdą. Jeśli analityk lub oprogramowanie defensywne pominie analizę kodu wywoływanego przez kod biblioteczny MFC, jeśli nie posiada innych mechanizmów wykrywania złośliwych operacji, nie zarejestruje on wywołania złośliwego kodu i błędnie zaklasyfikuje próbkę jako uprawnioną.



Rysunek 1: Wywołanie kodu za pośrednictwem bibliotek MFC

Mechanizm ten został opisany już w poprzednich publikacjach CERT Orange Polska (m.in. w raporcie z analizy próbki Dyre). W przypadku próbki Cleopatra, złośliwy kod wywoływany jest za pośrednictwem wywołania CDialog::DoModal.

Funkcja ta w ramach wywołania zwrotnego (ang. Callback), rozpocznie wykonywanie kodu, który jest odpowiedzialny za przygotowanie głównego kodu odpowiedzialnego za wypakowanie droppera. Kod ten jest przechowywany na stosie procesu, co oznacza, że najprawdopodobniej został zadeklarowany jako zmienna lokalna funkcji.

```
Stack[00000DA8]:0012BC3E ; ------------------------------
Stack[00000DA8]:0012BC3E
Stack[00000DA8]:0012BC3E loc_12BC3E:
EIP Stack[00000DA8]:0012BC3E push    ebp
Stack[00000DA8]:0012BC3F mov     ebp, esp
Stack[00000DA8]:0012BC41 mov     eax, 5950h
Stack[00000DA8]:0012BC46 call    near ptr unk_12C49C
Stack[00000DA8]:0012BC4B push    ebx
Stack[00000DA8]:0012BC4C push    esi
Stack[00000DA8]:0012BC4D push    edi
Stack[00000DA8]:0012BC4E call    near ptr unk_12BC3D
Stack[00000DA8]:0012BC53 xor     ebx, ebx
Stack[00000DA8]:0012BC55 mov     byte ptr [ebp-24h], 6Bh
Stack[00000DA8]:0012BC59 mov     byte ptr [ebp-23h], 65h
```

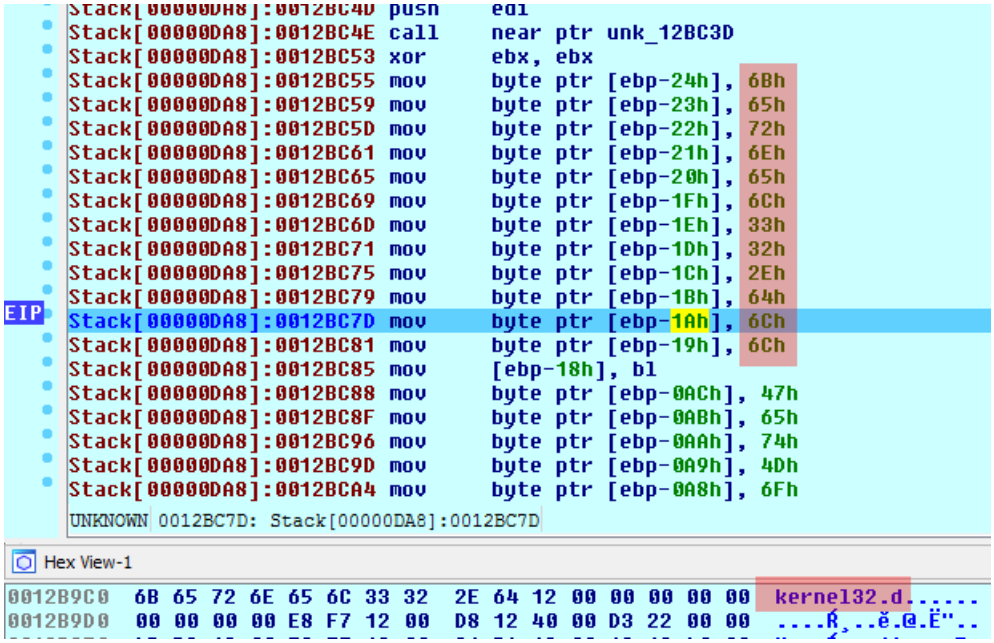
Rysunek 2: Wykonywanie kodu na stosie

W większości uprawnionych aplikacji traktowanie zmiennych (czyli zapis i odczyt do nich) jako kod (czyli wykonywanie) nie ma miejsca. Zachowanie takie należy traktować jako anomalię, która w większości przypadków oznacza, że mamy do czynienia ze złośliwą funkcją.

Wykonywany na stosie kod rozpoczyna proces importowania potrzebnych do działania wywołań bibliotecznych. Działanie to nazywane jest późnym wiązaniem (ang. late binding) w przeciwieństwie do wczesnego wiązania (ang. early binding). Jest ono stosowane zarówno w uprawnionych jak i złośliwych aplikacjach, przy czym w tych ostatnich ma na celu ukrycie listy stosowanych przez program wywołań do ostatniej chwili. Późne importowanie stosuje się, by ukryć wykorzystanie podejrzanych wywołań i zapobiec wzbudzeniu podejrzeń na etapie analizy statycznej próbki (przed uruchomieniem).

W trakcie analizy statycznej wciąż istnieje możliwość wykrycia podejrzanych wywołań za pomocą wyszukiwania łańcuchów znaków. Przykładowo, jeśli w próbce występuje łańcuch „CreateToolhelp32Snapshot” (wywołanie służące do przeglądania innych procesów działających w systemie), to nawet jeśli jego wykorzystanie nie jest deklarowane przez próbkę, można przypuszczać, że jest ono importowane za pomocą późnego wiązania i następnie wykorzystywane.

Aby ukryć nazwy funkcji przed analizą łańcuchów, Cleopatra przechowuje je w postaci kodu posługującego się pojedynczymi znakami. W ten sposób, znak po znaku, przygotowuje nazwy wywołań do późnego wiązania, ale uniemożliwia wykrycie łańcuchów w trakcie analizy statycznej.



Rysunek 3: Zakodowane nazwy bibliotek

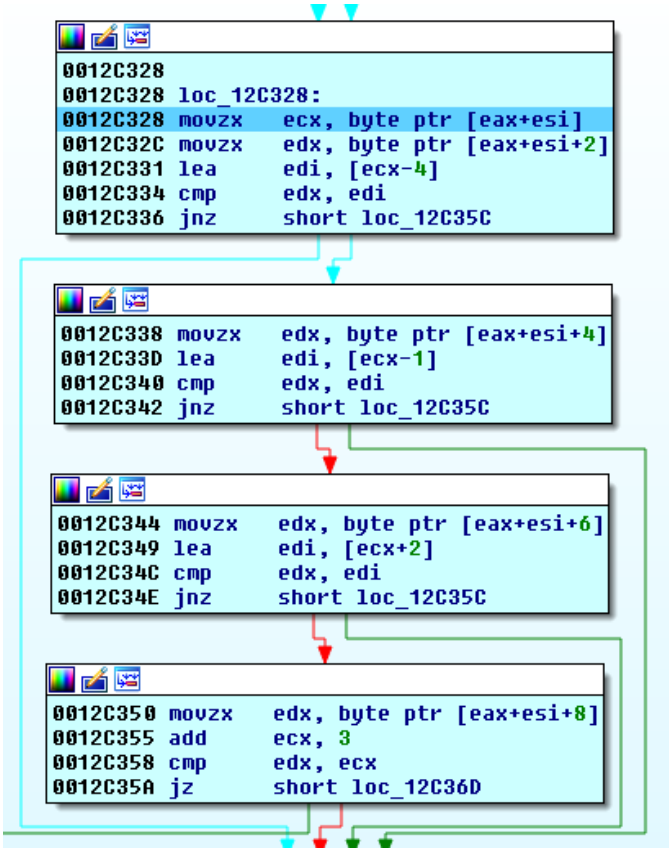
Po załadowaniu wszystkich potrzebnych wywołań, protektor przystępuje do wypakowania z próbki kodu droppera. W większości przypadków protektor odnajduje w próbce zasób (ang. resource) lub ładuje jakiś fragment z określonej lokalizacji próbki. W przypadku Cleopatra jednak żadna lokalizacja nie jest podana. Przeznaczony do załadowania kod jest ukryty w próbce i musi zostać odnaleziony według pewnego algorytmu.

Cleopatra lokalizuje kod droppera po specjalnej sygnaturze. Sygnatura ta spełnia ściśle określone zależności arytmetyczne. Otóż, poprawna sygnatura kodu to sygnatura, w której:

- pierwszy i trzeci bajt różnią się o 4
- pierwszy i piąty bajt różnią się o 1
- pierwszy i siódmy bajt różnią się o 2
- pierwszy i dziewiąty bajt różnią się o 3

Aby odnaleźć ukryty kod, należy załadować do pamięci całą próbkę i analizując każdy bajt, od początku do końca, sprawdzać, czy spełniona jest zależność.

Takie podejście ma na celu utrudnienie wyznaczenia konkretnej sygnatury kodu droppera, po której można by zidentyfikować próbkę. Sygnatur spełniających powyższe równości jest bardzo dużo, ale nie jest ona na tyle popularna, żeby protektor popełnił błąd i załadował zły kod.



Rysunek 4: Algorytm dopasowania sygnatury

Po zidentyfikowaniu kodu jest on odszyfrowywany i wykonywany.

3. Dropper

Podstawowym zadaniem droppera jest po pierwszym uruchomieniu programu w systemie ofiary rozmieścić komponenty złośliwej aplikacji w systemie (de facto zainstalować ją) i dopilnować, by były one uruchamiane przy każdym uruchomieniu systemu. Dropper w próbce Cleopatra posiada dość złożony algorytm instalacji i szeroki wachlarz funkcji.

Dość nietypowa jest również organizacja kodu. Dropper posługuje się tylko jedną funkcją, która sama siebie wywołuje z różnymi parametrami. W zależności od parametru, wykonywana jest konretna procedura instalacji. Poniżej zestawiono niektóre ze stosowanych procedur.

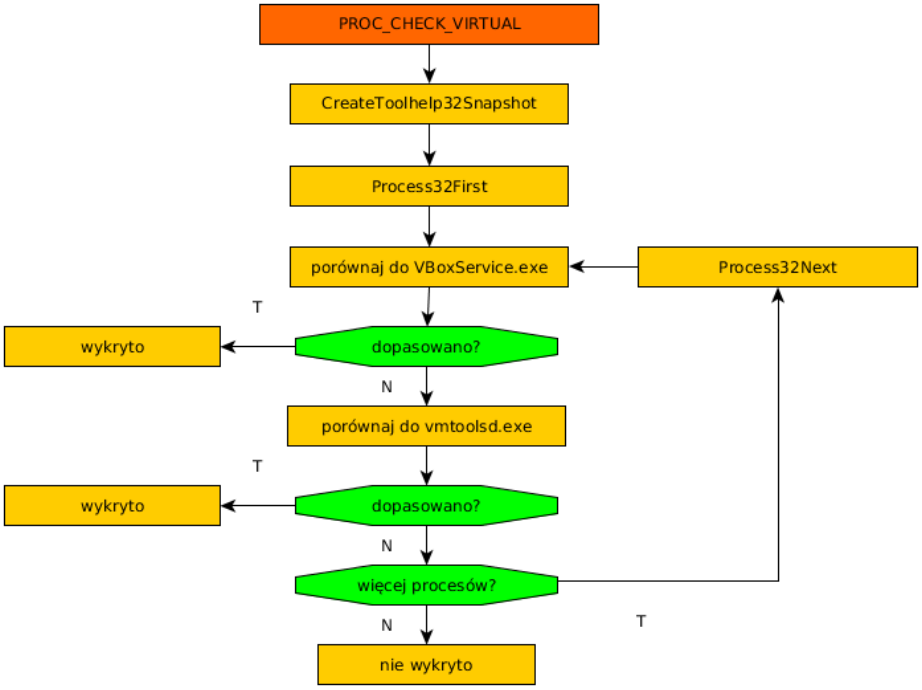
Parametr	Procedura
PROC_EXIT (0x0)	Zakończ działanie procesu
PROC_CHECK_PERMISSIONS (0x4)	Sprawdź, z jakimi uprawnieniami został uruchomiony dropper
PROC_CHECK_SAMPLE(0x5)	Sprawdź, z jakimi parametrami można otworzyć plik droppera
PROC_CHECK_VIRTUAL(0x6)	Sprawdź, czy proces droppera działa w maszynie wirtualnej

PROC_INJECT (0x14)	Wstrzyknij zadany kod do innego procesu
PROC_INJECT_ESCALATED(0x9)	Podejmij próbę wstrzyknięcia kodu do procesu iexplore.exe
PROC_REMOVE(0xa)	Usuń próbkę (zastąp ją jednym z uprawnionych plików systemu Windows)
PROC_INSTALL (0x1)	Przeprowadź instalację aplikacji

Jako pierwsza jest wykonywana procedura PROC_INSTALL. Do podjęcia decyzji przy poszczególnych etapach instalacji wykorzystuje ona informacje pozyskane przez inne podprocedury diagnostyczne.

4. Inne metody wykrywania środowiska analitycznego

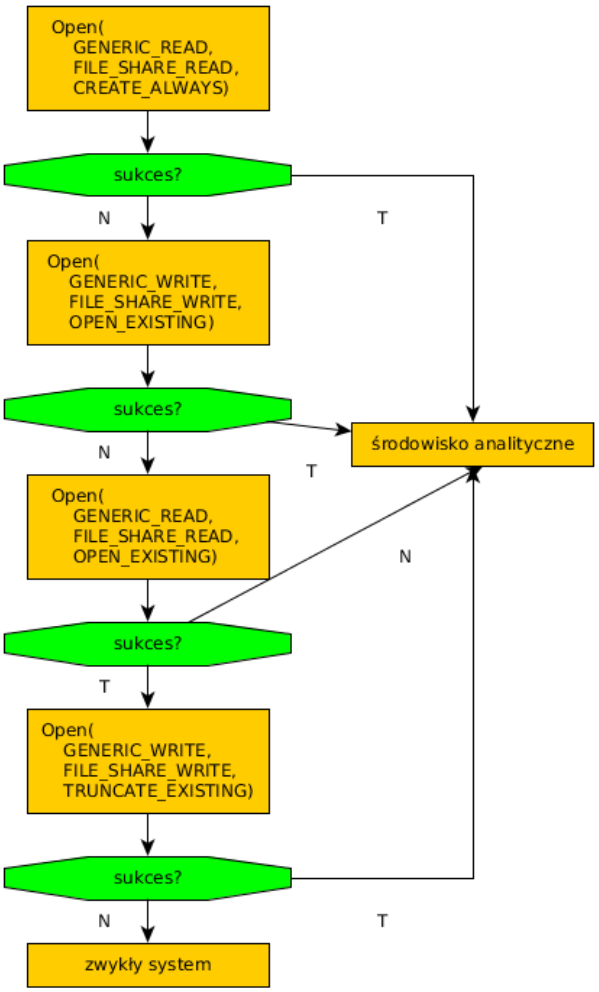
Jedną z diagnoz, które może przeprowadza Cleopatra, jest sprawdzenie, czy próbka została uruchomiona w wirtualnym środowisku. Przegląda ona wszystkie uruchomione w systemie procesy i porównuje ich nazwy z powszechnie znanymi nazwami platform wirtualizacji.



Rysunek 5: Algorytm sprawdzania wirtualizacji

Kolejnym badaniem jest podjęcie szeregu prób otworzenia własnego modułu z różnymi ustawieniami uprawnień dotyczących zapisu i odczytu, dzielenia pliku między procesami oraz trybu otwarcia. Cleopatra próbuje otworzyć plik z czterema kombinacjami tych parametrów. Ze względu na właściwości systemu Windows w zwykłym systemie trzy z tych prób powinny zakończyć się niepowodzeniem z kodem błędu 0x20 – naruszenie zasad współdzielenia pliku, a

jedna – powodzeniem. Jeśli jest inaczej – występuje podejrzenie, że program działa na platformie analitycznej.



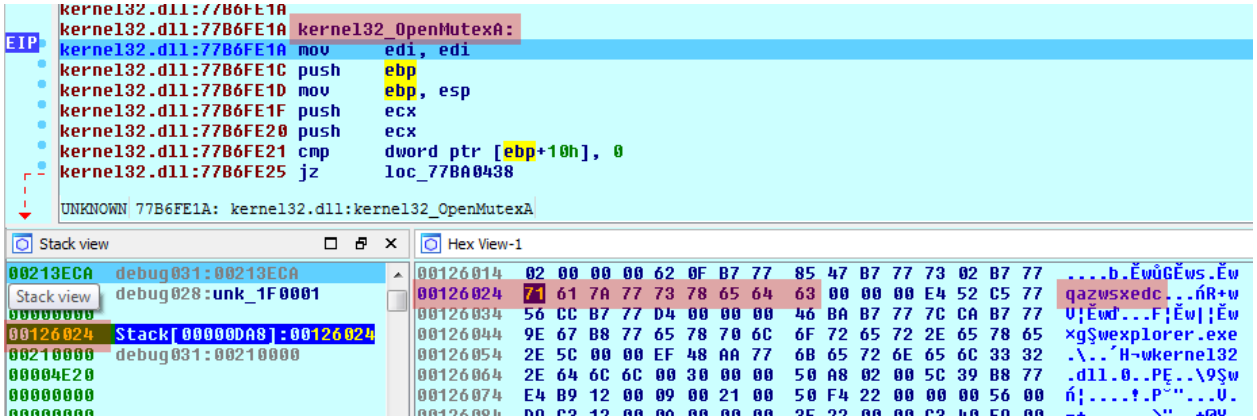
Rysunek 6: podjęcie prób otwarcia własnego pliku

Malware również mierzy prędkości wykonania instrukcji, za pomocą systemowej funkcji GetTickCount, zwracającej liczbę milisekund, które upłynęły od czasu uruchomienia systemu. Jeśli czas ten jest zbyt krótki, lub czas pomiędzy dwoma wywołaniami GetTickCount jest zbyt duży, występuje podejrzenie uruchomienia próbki na platformie analitycznej.

Dość ciekawą techniką jest podjęcie próby otworzenia nieistniejącego mutexu. Wiele gatunków złośliwego oprogramowania wykorzystuje mechanizmy takie jak mutexy do synchronizacji pomiędzy pomiędzy swoimi komponentami. Działają one jak sygnalizacja świetlna. Jeden z komponentów (np. wstrzyknięty do procesu kod) może za ich pomocą zasygnalizować pozostałym komponentom (np. dropperowi), że udało mu się poprawnie rozpocząć działanie.

Automatyczne oprogramowanie diagnostyczne czasami ingeruje w tą sygnalizację, aby odkryć jak najwięcej funkcji badanej próbki. Przykładowo, jeśli dropper czeka na mutex, oprogramowanie automatycznie go sygnalizuje, aby móc dalej obserwować działanie droppera.

Cleopatra zastawia pułapkę na takie narzędzia czekając na sygnalizację mutexu, którego nie tworzyła i który nie powinien istnieć. Próba otworzenia takiego mutexa zakończona powodzeniem wzbudza podejrzenie, że program działa w środowisku analitycznym.

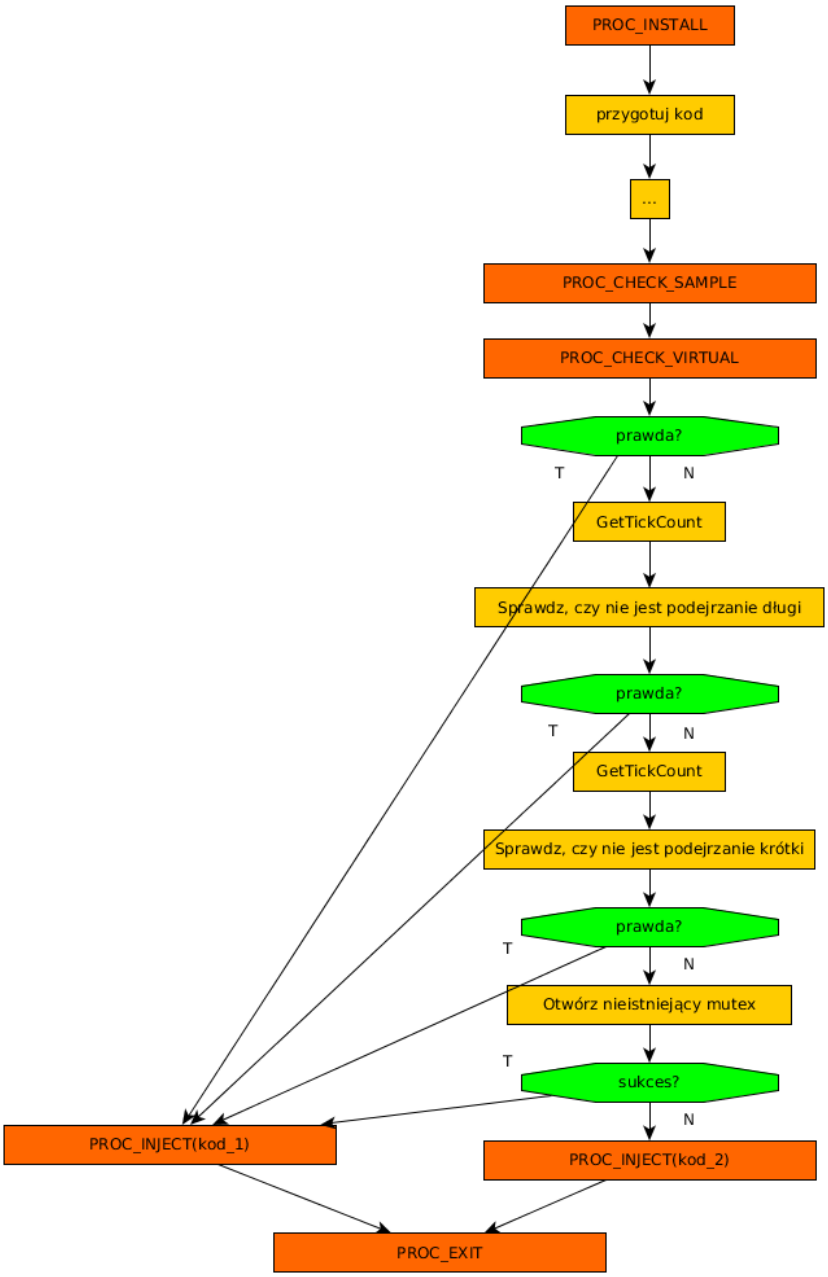


Rysunek 7: Sprawdzanie nieistniejącego mutexu

5. Przykładowa instalacja

Mając świadomość istnienia szeregu tego typu technik próbujących określić, czy Cleopatra działa w istocie w zwykłym systemie ofiary, można zadać pytanie: co dalej robi z uzyskanymi wynikami? Otóż, dropper na podstawie przeprowadzonej diagnostyki podejmuje decyzję, który fragment kodu zostanie wstrzyknięty do procesu w procesie instalacji. Takie działanie otwiera możliwość przedstawienia mało uważnemu analitykowi zupełnie innego kodu niż ten, który jest instalowany na stacji rzeczywistej ofiary.

Przykład algorytmu instalacji stosowanego przez probkę Cleopatra został przedstawiony poniżej.



Rysunek 8: Przykładowy przebieg instalacji - wybór kodu do wstrzyknięcia

CERT Orange Polska radzi:

Pamiętaj, że Twój system jest atrakcyjnym celem dla cyberprzestępców. Dbaj o jego bezpieczeństwo. Jeśli nie posiadasz oprogramowania antywirusowego – zainstaluj je, a jeśli masz zainstalowane – aktualizuj regularnie bazę informacji o zagrożeniach. Nie uruchamiaj programów, które nie pochodzą ze zweryfikowanych źródeł. Zasada ta dotyczy również załączników wiadomości e-mail.