Dla początkujących

Malware – jak wojna to wojna!

Michał Bucko

stopień trudności

Malware to potoczne określenie złośliwego oprogramowania (ang. *malicious software*). Jest ono szeroko rozpowszechnione w Internecie, infekując tysiące komputerów na całym świecie. Wiedza hakerów jest bardzo duża, a współczesne programy typu malware są coraz bardziej zaawansowane.

W przyszłości ataki z wykorzystaniem malware mogą mieć miejsce na ogromną skalę i przede wszystkim dlatego tak ważna stała się kwestia bezpieczeństwa systemów informatycznych. Jednym z kluczowych zagadnień jest tu analiza malware. Aby obronić się przed rozchodzącym się w danej chwili po sieci malware wymagana jest duża wiedza: nie tylko wykorzystują one zazwyczaj dopiero co odkryte luki, ale też instalują w systemach swoich ofiar rootkity, są spakowane naprawdę wrednymi pakerami oraz wykrywają działające debuggery. W niniejszym artykule chciałbym przedstawić Czytelnikom kilka zagadnień, z którymi często borykają się analitycy bezpieczeństwa; nie jest moją intencją stworzenie słownika czy ujęcia komplementarnego.

Kilka terminów, które musisz zrozumieć

- PE / *portable executable* (przenośne wykonywalne), natywny format plików Windows. W telegraficznym skrócie, pliki takie mają być w założeniu przenośne (pomiędzy platformami zarówno programowymi, jak i sprzętowymi).
- Struktura PE zawiera informacje dostarczające systemowi operacyjnemu wiedzy, jak obsłużyć należy zawarty w pliku kod wykonywalny.

Z artykułu dowiesz się

- wprowadzenie do analizy programów typu malware,
- przegląd niektórych ze stosowanych przez badające malware laboratoria, technik,
- jak radzić sobie z robakami, złośliwymi aplikacjami i wirusami,
- jak wygląda od środka laboratorium badania malware.

Co powinieneś wiedzieć

- podstawowa znajomość zagadnień związanych z bezpieczeństwem technologii informacyjnych,
- terminy takie jak XSS, DNS spoofing, phishing, zero-day, rootkit będą wykorzystywane bez uprzedniego ich wyjaśnienia,
- warto także zapoznać się z narzędziami prezentowanymi w artykule. Pozwoli to na lepsze zrozumienie tematu i będzie z całą pewnością stanowiło bardzo dobre ćwiczenie praktyczne.

- Paker PE / kompresja i/lub maskowanie treści plików PE; w skrócie, używa się ich aby:
- zamaskować zawarte w kodzie adresy sieciowe i URL, utrudnić jego modyfikację, zmniejszyć jego rozmiar, ograniczyć kradzież kodu.

Krótkie szkolenie z pakerów

Fragment pierwszego programu – Listing 1. Jaki wykorzystano paker? Dwa wytłuszczone zapisy sygnalizują zastosowanie narzędzia PECompact2. A w przypadku ukazanym na Listingu 2.? Wytłuszczenie wskazuje na paker ASPACK (obecność sekcji *.aspack*). Bardzo popularny jest również paker UPX:

Listing 1. *Fragment pierwszego programu*

```
!This program cannot be
      run in DOS mode.
Rich
Xx0C
.text
PEC2
.rsrc
,Xh($
ject1
ltFp.7
PECompact2
```

Listing 2. *Fragment programu*

```
This program must be
      run under Win32
CODE
DATA
.idata
.tls
.rdata
.reloc
.rsrc
.aspack
.adata
```

Listing 3. *Informacje pochodzące z analizy statycznej*

```
RCPT TO:<
...
MAIL FROM:<
DATA
X-Mailer
...
steal@############
evil@#########
smtp.##########
```

```
This program must be run under Win32
UPX0
UPX1
.rsrc
1.25
UPX!
```

Wspomniane powyżej pakery to tylko pojedyncze przykłady. W rzeczywistości istnieje bardzo duża ilość różnych narzędzi tego rodzaju, częściowo wyspecjalizowanych pod kątem określonego złośliwego oprogramowania. Przykładowo, popularnym pakerem jest *Themida*; jest on często wykorzystywany, ponieważ umie wykrywać wirtualne maszyny, co utrudnia analizę spakowanego nim malware.

Jak zidentyfikować paker?

Do identyfikacji ciągów znaków wykorzystuje się programy *strings* (firmy Sysinternals) bądź *BinText*. Bardzo popularnym w środowisku analityków bezpieczeństwa narzędziem jest także PEiD; niestety korzysta on z ograniczonej bazy pakerów, w efekcie czego niektóre pakery muszą być analizowane ręcznie.

Skąd wiemy że to malware, jeżeli go nie otworzyliśmy? Cóż, odpowiedź na to pytanie jest problematyczna: gdyby ocena taka była prosta, odpowiednie rozwiązanie zostałoby zapewne wbudowane we wszystkie istniejące systemy operacyjne. W rzeczywistości zadanie to wydaje się bardzo trudne – można jednak za pomocą wstępnej statycznej analizy pod kątem malware wyszukać informacje sygnalizujące zagrożenie. Przeprowadzając statyczną analizę możemy znaleźć:

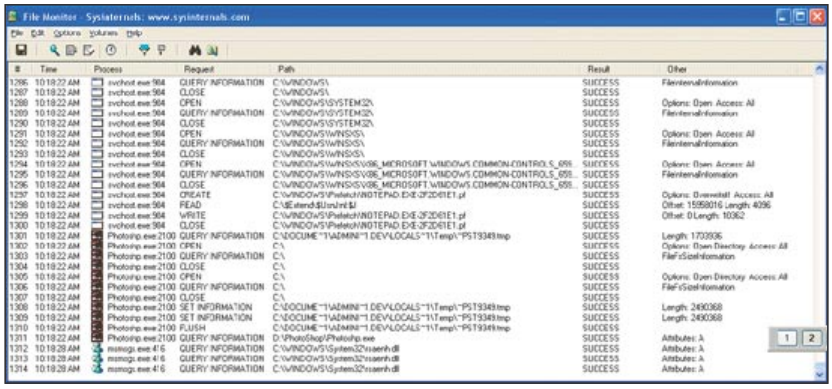
- pewne szczególne adresy URL (na przykład banków, instytucji finansowych, serwisów transakcyjnych, paneli administracyjnych itp.),
- fragmenty wiadomości e-mail, stron HTML, skryptów JS bądź ActiveX, VBS itp.)
- obrazy o treści powiązanej z instytucjami finansowymi,
- numery kart kredytowych,
- typowe nazwy i hasła użytkowników,
- dziwne parametry dla połączeń sieciowych, itd.

Przykładowymi informacjami pochodzącymi z analizy statycznej mogłyby być te zaprezentowane na Listingu 3.

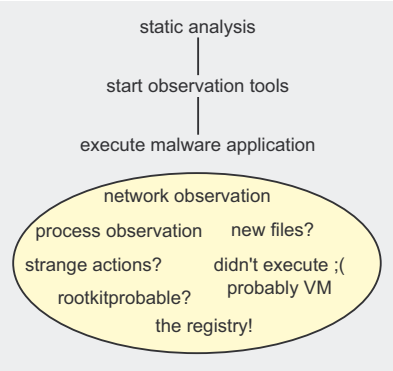
Powyższy fragment kodu nie wymaga, jak sądzę, szczegółowego komentarza. Wyraźnie widać, że jest on powiązany z wysyłaniem na zdalną maszynę listu z jakimiś informacjami. W kontekście analizy malware, listy tego rodzaju służą zwykle kradzieży poufnych informacji bądź wysyłaniu niechcianej korespondencji. Kolejny fragment kodu:

```
www.evilwhatever############
evilhack####################
Internet Banking HELLOWORLD
Mr President
Admin
Admin1234admin
```

Tutaj można domniemywać, iż złośliwy program atakuje infrastrukturę interne-



Rysunek 1. *Monitor plików*



Rysunek 2. *Co mam zrobić?!*

towych usług bankowych; posiadamy jednak zdecydowanie za mało informacji, by wydać jednoznaczny osąd.

Wciąż nie chcemy próbować uruchamiać programu? Jest on spakowany UPXem, widoczne są ciągi programu WinRAR. Po wnikliwej analizie możemy dojść do wniosku, że złośliwy program zamierza rozpakować z archiwum jakieś pliki. Także i ten sposób działania jest bardzo popularny – malware wyciąga pliki z archiwum, dodaje odpowiednie klucze do rejestru i uaktywnia się tuż po restarcie komputera. Dzięki temu na początku nie jest uruchamiany żaden złośliwy kod.

Wprowadzenie do analizy malware

Czego nam potrzeba? Zdecydowanie powinniśmy uruchamiać złośliwe oprogramowanie pod wirtualną maszyną; z drugiej strony, w wielu przypadkach malware wykrywa maszynę wirtualną i nie uruchamia się (przypadek ten omówimy za chwilę). Dalej, przydadzą nam się monitory plików i rejestru, sniffery, wykrywacze rootkitów i wiele innych narzędzi. Tuż przed uruchomieniem programu zalecane jest zastosowanie technik odzyskiwania danych stosowanych w śledztwach komputerowych. W wielu przypadkach złośliwe oprogramowanie (szczególnie to atakujące instytucje finansowe) zawiera obrazy i logo imitujące bank ofiary i niekiedy można wydobyć zeń te dane. W miarę upływu czasu staje się to coraz bardziej skomplikowane, jako że dane tego rodzaju są niezwykle subtelnie maskowane – nie mamy ich podanych na tacy.

Po co korzystać z maszyn wirtualnych

Zastosowanie maszyny wirtualnej (ang. *virtual machine*, VM) przy analizie malware pozwala na pracę z różnymi systemami operacyjnymi, ułatwia monitorowanie, pozwala na przywrócenie wcześniejszego stanu maszyny oraz zapewnia izolację od właściwego systemu operacyjnego.

Metody wykrywania maszyn wirtualnych

Maszyna wirtualna pozostawia w pamięci ślady, posiada swój wirtualny sprzęt i własne instrukcje. Jej ślady znaleźć można wśród procesów i w systemie plików, a także w rejestrze. Analityk może także znaleźć w pamięci pewne ciągi znaków, względnie rozejrzeć się w różnych miejscach struktur systemu operacyjnego (bardzo ważne jest przyjrzenie się Tablicy Deskryptorów Przerwań – ang. *Interrupt Descriptor Table*). W listopadzie 2004 Joanna Rutkowska przedstawiła tak zwaną czerwoną pigułkę, pozwalającą na dość skuteczne wykrywanie maszyn wirtualnych. Czerwona pigułka wywołuje SIDT (ang. *single machine language instruction*, pojedynczą instrukcję kodu maszynowego) i patrzy na rezultaty; jako że położenie IDT określane jest przez odpowiednie reguły, pierwszy bajt zwracany przez SIDT stwierdza, czy wykorzystywana jest wirtualna maszyna. Jest to jednak dopiero początek wykrywania maszyny wirtualnej, a oprócz SIDT można w tym celu wywołać SGDT i SDLT. Wirtualne maszyny często definiują też swój własny, wirtualny sprzęt, który może zostać zidentyfikowany. To wszystko, co chcieliśmy tutaj powiedzieć o wykrywaniu wirtualnych maszyn – temat ten można drążyć miesiącami, nie jest on jednak przedmiotem niniejszego artykułu.

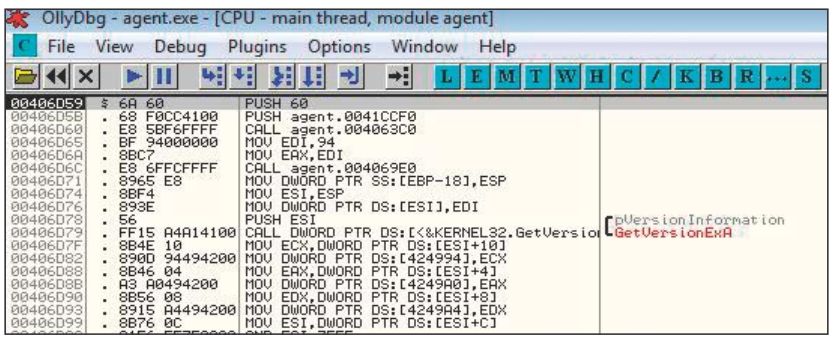
Co robić, gdy malware wykrywa VM

W przypadkach takich, naszym zadaniem jest obejście w jakiś sposób tej detekcji. Czasami korzystamy z rootkitów, by ukryć VM przed złośliwymi programami (przykładowo, było to niegdyś skuteczne wobec pakerów *Themida*); ogółem staramy się uczynić wirtualną maszynę niewidzialną. VMware oferuje pewne nieudokumentowane możliwości, które pozwalają mu ukrywać się przed malware. Istnieje także szereg innych metod pozwalających na ukrywanie VM; część z nich powstaje chałupniczo, jako że ograniczenie wykrywalności wciąż jest w kontekście analizy złośliwych programów zagadnieniem stosunkowo nowym.

Co nam potrzeba, by zacząć?

Przede wszystkim należy zaopatrzyć się w kilka narzędzi:

- *BinText* wydobywa łańcuchy znaków i jest bardzo przydatny podczas statycznej analizy malware (tj. przed uruchomieniem czegokolwiek)
- *Filemon* to jeden z wielu dostępnych monitorów plików – obserwuje on działania na systemie plików.
- *Regmon* monitoruje stan rejestru. Programy sieciowe wykorzystuje się do obserwacji generowanego przez złośliwy program ruchu sieciowego.
- Bardzo przydatnym, w przypadku pracy z rejestrem narzędziem jest



Rysunek 3. *Cześć jestem Olly*

także *Regshot* – pozwala on na porównanie stanu rejestru np. tuż przed i po iniekcji.
- Dzięki *Process Explorerowi* możemy obserwować działające w danej chwili procesy, opcjonalnie w czasie rzeczywistym,
- UPX – paker często stosowany przez twórców malware,
- *OllyDBG* – bardzo popularny debugger.

Chcielibyśmy wiedzieć, jakie porty są otwierane i co jest wysyłane (prawdopodobnie skradzione poufne informacje) z naszej maszyny. Ważne jest także znajdowanie serwerów, z którymi łączy się malware. W efekcie potrzebny jest nam szereg narzędzi do obserwacji sieci, takich jak:

- *TCPView* – wyświetla listę otwartych portów,
- *TDIMon* – obserwuje aktywność w sieci,
- Ethereal, Snort bądź inny sniffer.

Typowe procedury postępowania w analizie złośliwych programów

Nigdy nie powinniśmy ryzykować zarażenia innych komputerów. Analiza powinna mieć zatem miejsce w izolowanym środowisku; popularnym rozwiązaniem są tu maszyny wirtualne. Komputery do analizy bywają często fizycznie odłączone od sieci, by zapobiec przypadkowemu rozprzestrzenianiu próbki; z drugiej strony, w wielu przypadkach (gdy wykorzystywana jest wirtualizacja) stosowana jest jedynie separacja logiczna, co może mieć poważne konsekwencje, jeżeli złośliwy program zdoła obejść wirtualizację i otrzymać nieuprawniony dostęp do systemu.

Nie należy nigdy zdradzać informacji na temat malware osobom, do których nie mamy zaufania. Nie do uniknięcia jest kontrola dostępu. Pliki powinny być przekazywane innym analitykom w postaci skompresowanej i zabezpieczonej hasłem, a także poprzez bezpieczne kanały komunikacyjne. Najbardziej zalecane jest, by po prostu nigdy nie rozsyłać malware – nawet pojedyncza pomyłka może mieć tutaj poważne konsekwencje.

Ochrona przed podwójnym kliknięciem – należy pamiętać, że pliki mogą przez przypadek zostać uruchomione przez kliknięcie na nie, w związku z czym powinno się zmieniać rozszerzenie na jakieś bezpieczne. Jest to jeden z najpowszechniejszych błędów wśród niedoświadczonych analityków robaków.

Coś prostego na początek

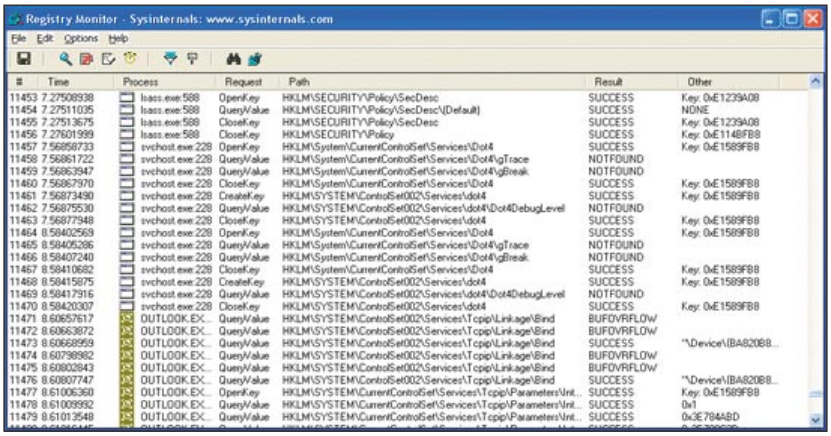
Otrzymaliśmy informację o nowym problemie z bezpieczeństwem. Opisano go tak, jak to widzimy poniżej:

[..] Każda osoba w naszym dziale otrzymała e-mail z odnośnikiem do strony banku, z którym firma współpracuje. [..] Nikt nie spodziewał się jakichkolwiek problemów.[..] Dwa tygodnie później ukradziono nam poufne informacje[..]. Nasze maszyny mają zainstalowane wszystkie dostępne łaty[..]

A zatem pracownik firmy otrzymuje e-mail z odnośnikiem do strony banku. Załóżmy, że nie spodziewamy się zastosowania spoofingu DNS; istotą problemu jest tu więc zapewne podatność serwisu banku na ataki *cross-site scripting* bądź zezwalanie na stosowanie przekierowań. Przykład takiej sytuacji: *www.our-good-bank-site.com/?page=%68%74%74%70%3A//%77%77%77%2e%68%61*%63%6B%2e%70%6C

Chcemy połączyć się z zaufaną (w naszym przypadku) stroną, w dodatku znacznie lepiej zabezpieczoną niż przeciętna; niemniej, podążamy za jakimś innym odnośnikiem. W przypadku luk typu XSS można też po prostu przygotować tego typu fałszywe strony, przeznaczone do oszukania użytkownika.

Co dalej? Podążając za odnośnikiem, trafiamy na odpowiednio przygotowaną stronę. Atak ten nie ma nic wspólnego z phishingiem; strona ta powoduje tylko nagłe zamknięcie przeglądarki, my zaś zapominamy wkrótce o całym incydencie. Dwa tygodnie później następuje kradzież poufnych informacji.



Rysunek 4. *Co mamy w rejestrze?*

Analiza

Jest całkiem prawdopodobne, że w niniejszym przypadku zastosowano malware, wykorzystujące niedawno odkrytą lukę. Przyjrzano się podejrzanemu odnośnikowi; treść przygotowanej pod nim strony WWW jest zamaskowana, po jej zdekodowaniu stwierdzamy, iż atakowała ona nieznaną wcześniej wadę przeglądarki. Wykorzystany w tym eksploicie szelkod należał do kategorii pobierz i uruchom, powodował pobranie ze zdalnej maszyny pliku wykonywalnego. Przyszła pora na pobranie tego pliku i rozpoczęciu nad nim pracy.

Analiza statyczna

Przede wszystkim ważne było oszacowanie możliwości działań złośliwego programu, na podstawie zawartych w nim ciągów znaków. Wydobyliśmy je z pliku za pomocą narzędzia *BinText*. Zastosowanie potem IDA Pro pozwoliło na stwierdzenie, iż plik spakowany jest *UPXem*. Po rozpakowaniu pliku wykonywalnego ponownie wykorzystano *BinText*; tym razem byliśmy

w stanie wydobyć z programu znacznie więcej interesujących informacji.

Łańcuchy znaków

SOFTWARE\Microsoft\Windows\CurrentVersion\Run Jeden z najpopularniejszych wśród malware łańcuchów, pozwala uruchomić złośliwy program zaraz po restarcie.

MAIL FROM Odkryliśmy, że dany program wysyła e-maile zawierające jakieś informacje. Póki co, nie wiemy jednak nic o tym, jakie to informacje.

Znaleźliśmy również inne łańcuchy typowe dla malware, na przykład nazwy plików przypominające te, które stosowane są przez różne rozwiązania antywirusowe. Do czego one służą? Cóż,w końcu jakieś procesy powinny zostać zabite, nieprawdaż? Znaleźliśmy również podejrzane nazwy plików w rodzaju *go2hell2398728.exe* – te z kolei wykorzystywane są do propagacji. Znaleziono także obrazek – logo zaufanego banku, jednak suma kontrolna MD5 tego logo oraz zastosowanego obrazka, różniły się. Najprawdopodobniej miało to na celu ominięcie pewnych zabezpieczeń.

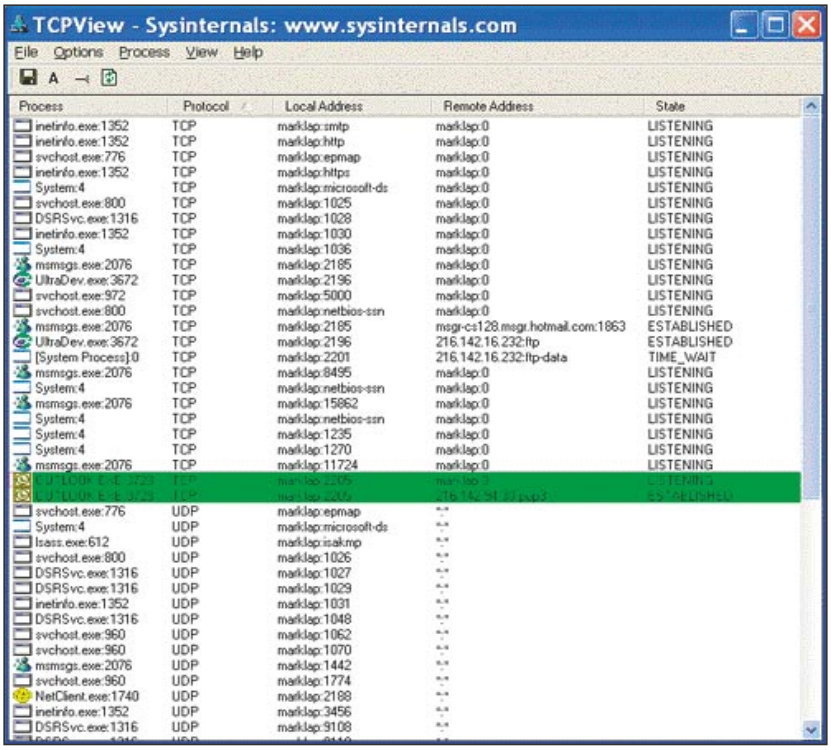
Analiza dynamiczna

Znalezienie istotnych informacji pozwoliło nam na odgadnięcie sposobu działania złośliwego programu: najprawdopodobniej wykorzystuje on połączenia sieciowe do wykradania poufnych informacji z komputerów swoich ofiar. Zamierzamy zaobserwować ten proces, zachodzące zmiany w rejestrze i systemie plików, a także prowadzić analizę ruchu sieciowego.

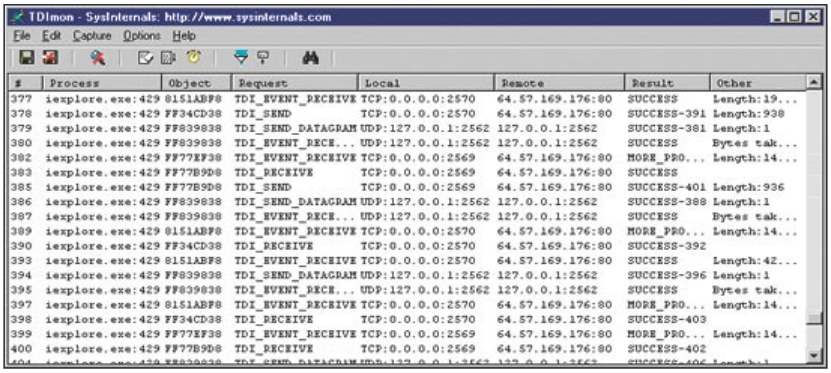
Wyjście programu *Regshot* zasygnalizowało stworzenie kilku plików; kilka innych dodano do plików współdzielonych. Nastąpiło wiele zmian w rejestrze. Analiza ruchu w sieci wykazała, że wysyłane e-maile zawierały treść losowo wybranych plików dokumentów. Złośliwy program generował wiele ruchu, nikt jednak niczego nie spostrzegł. W końcu leżący u podstaw problem został naprawiony.

Najważniejszą częścią analizy malware jest debugowanie. Debugowanie złośliwego programu pomaga nam zrozumieć szczegóły jego konstrukcji. Z drugiej strony, dokładne zdebugowanie takiej aplikacji wymaga czasu – ta dziedzina bezpieczeństwa wymaga niekonwencjonalnego podejścia. Jest ono już obsługiwane przez narzędzia automatycznej wizualizacji, IDA Pro posiada wiele wtyczek ulepszających jego funkcjonalność, jednak wciąż wiele pozostało w tej sprawie do zrobienia.

Powyższy przykład analizy malware jest przykładem prostym. Złośliwy program mógłby zainstalować rootkit bądź wykrywać wirtualne maszyny. W tym przypadku podstawowym problemem było zastosowanie nowo odkrytej luki, którą można było wykorzystać na wiele sposobów.



Rysunek 5. *Połączenia na talerzu(1)*



Rysunek 6. *Połączenia na talerzy(2)*

Podsumowanie

Mam nadzieję, że niniejszy artykuł pomógł wam zrozumieć sposoby radzenia sobie z malware. Zainteresowanych dokładniejszym poznaniem tego zagadnienia odsyłam do technik obchodzenia sygnatur programów antywirusowych, ataków, maskowania, pakowania i wydobywania danych. Artykuł ten nie zawiera żadnych informacji o stosowanych przez malware rootkitach, nie wspomina też o sposobach unikania firewalli. Warto pamiętać jednak, że wiedza na ten temat jest niezbędna. ●

O autorze

Michał Bućko jest niezależnym badaczem z dziedziny bezpieczeństwa informatycznego.