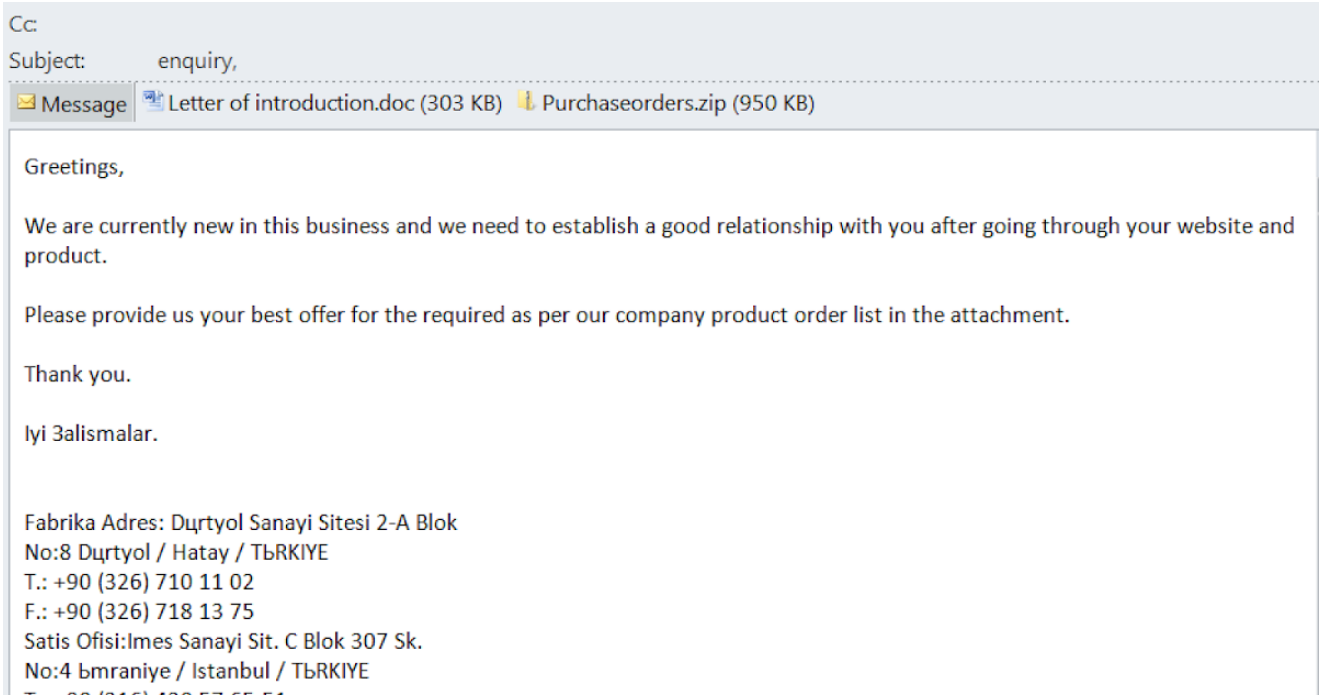
Modified Zyklon and plugins from India

blog.talosintelligence.com/2017/05/modified-zyklon-and-plugins-from-india.html


Cc:
Subject: enquiry,
Message | Letter of introduction.doc (303 KB) | Purchaseorders.zip (950 KB)

Greetings,

We are currently new in this business and we need to establish a good relationship with you after going through your website and product.

Please provide us your best offer for the required as per our company product order list in the attachment.

Thank you.

Iyi 3alismalar.

Fabrika Adres: Dцrtyol Sanayi Sitesi 2-A Blok
No:8 Dцrtyol / Hatay / TЬRKIYE
T.: +90 (326) 710 11 02
F.: +90 (326) 718 13 75
Satis Ofisi:Imes Sanayi Sit. C Blok 307 Sk.
No:4 Ьmraniye / Istanbul / TЬRKIYE

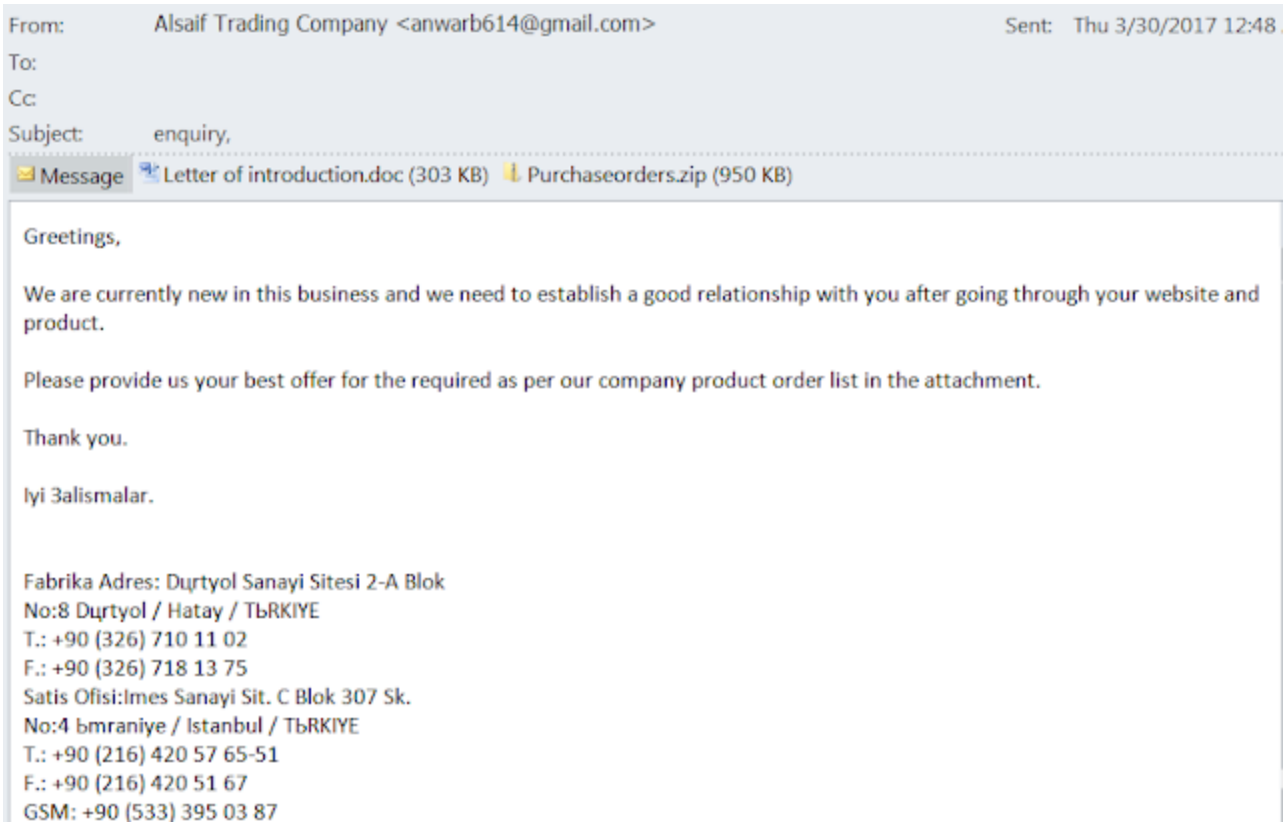
Introduction

Streams of malicious emails Talos inspects every day usually consist of active spamming campaigns for various ransomware families, phishing campaigns and the common malware family suspects such as banking Trojans and bots.. It is however often more interesting to analyze campaigns smaller in volume as they might contain more interesting malware. A few weeks ago I became interested in just such a campaign with a smaller number of circulating email messages. The email, first of them submitted from Middle East, purports to be coming from a Turkish trading company, which might further indicate the geographic area where the attacks were active. Analyzing malware is often like solving a puzzle, you have to do it piece by piece to reach the final image. In this case I spent more time analyzing the campaign than I initially planned. The campaign has many stages of the infection chain and all needed to be unraveled before the final payload level was reached. Furthermore, each of the stages used different development platform and was obfuscated in a different way. But let us start from the beginning.

Stage 1 - email

The email message contains two attachments. The first one is a Word document in the Office Open XML file format while the second is a ZIP file PurchaseOrders.zip, containing an

executable file PurchaseOrders.exe. This is a relatively unusual strategy for email campaigns as it is much more common for malicious emails to contain a single attachment rather than two or more. It seems that the attackers wanted to be double sure that the recipient will open at least one of the attachments.


From: Alsaif Trading Company <anwarb614@gmail.com> Sent: Thu 3/30/2017 12:48
To:
Cc:
Subject: enquiry,
Message Letter of introduction.doc (303 KB) Purchaseorders.zip (950 KB)

Greetings,

We are currently new in this business and we need to establish a good relationship with you after going through your website and product.

Please provide us your best offer for the required as per our company product order list in the attachment.

Thank you.

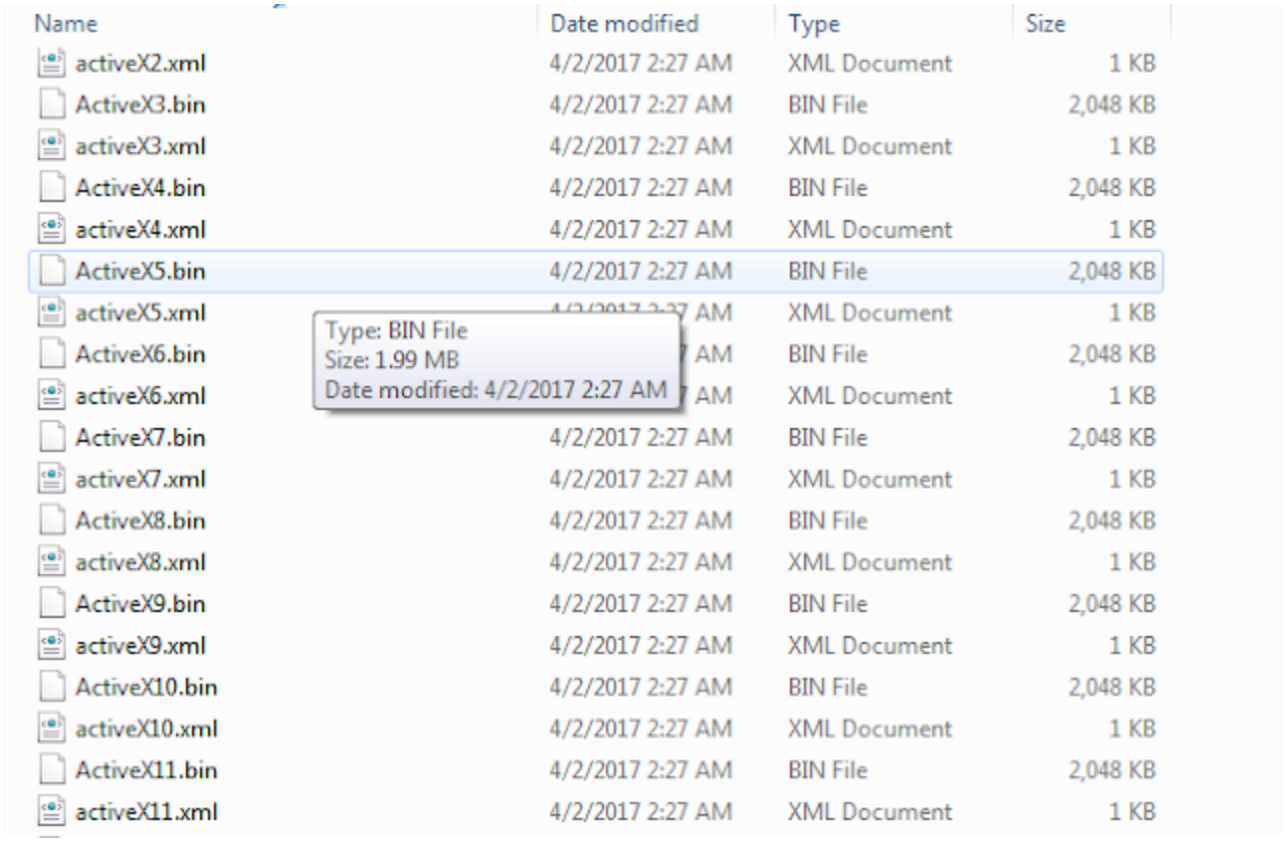
Iyi 3alismalar.

Fabrika Adres: Dųrtyol Sanayi Sitesi 2-A Blok
No:8 Dųrtyol / Hatay / TЬRKIYE
T.: +90 (326) 710 11 02
F.: +90 (326) 718 13 75
Satis Ofisi:Imes Sanayi Sit. C Blok 307 Sk.
No:4 bmraniye / Istanbul / TЬRKIYE
T.: +90 (216) 420 57 65-51
F.: +90 (216) 420 51 67
GSM: +90 (533) 395 03 87

Email campaign

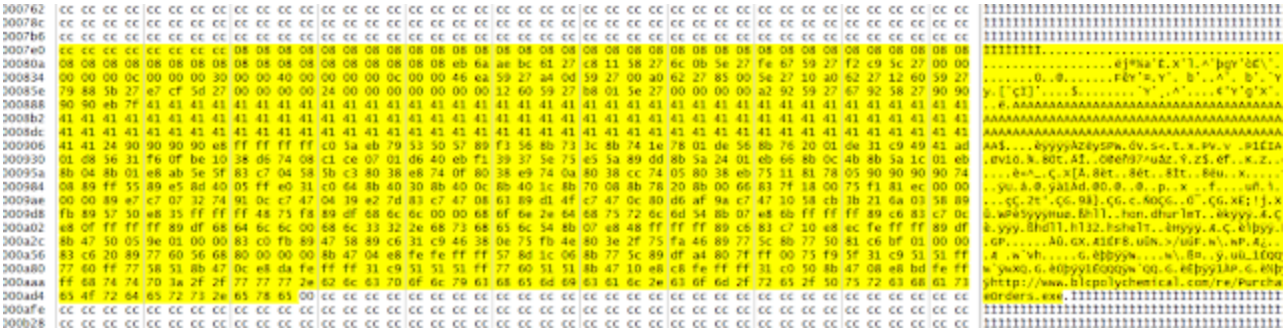
Stage 2a - Word Document - CVE-2013-3906

The Word document attachment, “Letter of introduction.doc” contains an exploit for CVE-2013-3906 tiff image file parsing vulnerability. The document contains multiple TabStrip (classid: {1EFB6596-857C-11D1-B16A-00C0F0283628}) ActiveX controls also used in CVE-2012-1856.



Embedded ActiveX controls used for heap spray

Embedded TabStrip ActiveX controls are used for heap spraying and the embedded TIFF file named image1.jpeg triggers the CVE-2013-3906 vulnerability. There are 40 embedded ActiveX controls and each is mapped in 2MB allocated memory space. In this case, exploitation takes time but the exploit eventually crashes the vulnerable versions of Word and starts the shellcode. The shellcode is immediately visible in the hex dump of the ActiveX OLE2 file and sprayed in the memory of the exploited Word process.



Heap sprayed shellcode from ActiveXn.bin files

Shellcode - hook evasion

The shellcode itself is relatively simple and, give or take, 450 bytes long, excluding the URL used for downloading the payload. As is often the case, the APIs are found by parsing the Process Environment Block (PEB) and traversing the linked list of loaded modules as well as their respective exported functions.

Notably, before calling required APIs, the shellcode checks for presence of inline hooks, often installed by endpoint security products and jumps over the installed hook code in order to avoid being noticed in their behavior detection windows.

```
.text:0040107F
.text:00401080
.text:00401080 ; =============== S U B R O U T I N E =======================================
.text:00401080
.text:00401080
.text:00401080 EvadeHookCall   proc near               ; CODE XREF: sub_4010A7+69↓p
.text:00401080                                         ; sub_4010A7+8C↓p ...
.text:00401080                 cmp     byte ptr [eax], 0E8h ; Is it a call?
.text:00401083                 jz      short loc_401094
.text:00401085                 cmp     byte ptr [eax], 0E9h ; Or a long jump?
.text:00401088                 jz      short loc_401094
.text:0040108A                 cmp     byte ptr [eax], 0CCh ; Or a breakpoint?
.text:0040108D                 jz      short loc_401094
.text:0040108F                 cmp     byte ptr [eax], 0EBh ; Or short jump?
.text:00401092                 jnz     short loc_4010A5
.text:00401094
.text:00401094 loc_401094:                             ; CODE XREF: EvadeHookCall+3↑j
.text:00401094                                         ; EvadeHookCall+8↑j ...
.text:00401094                 cmp     dword ptr [eax+5], 90909090h ; Legit Windows hook (Win7+ )
.text:0040109B                 jz      short loc_4010A5
.text:0040109D                 mov     edi, edi
.text:0040109F                 push    ebp
.text:004010A0                 mov     ebp, esp
.text:004010A2                 lea     eax, [eax+5]
.text:004010A5
.text:004010A5 loc_4010A5:                             ; CODE XREF: EvadeHookCall+12↑j
.text:004010A5                                         ; EvadeHookCall+1B↑j
.text:004010A5                 jmp     eax
.text:004010A5 EvadeHookCall   endp ; sp-analysis failed
.text:004010A5
.text:004010A7
.text:004010A7 ; =============== S U B R O U T I N E =======================================
```

Evading security hooks

If the user was infected by the attached Word document, the shellcode would download and execute an executable from a legitimate, compromised server. The C2 server for the final payload is extracted from a configuration blob stored encrypted within the downloaded payload body.

Stage 2b - PurchaseOrders.exe

The executable downloaded by the shellcode is identical in its functionality to the executable attached to the email so we are eventually coming to PurchaseOrder.exe which will eventually get executed whether the user opens the attached document or if they immediately go for launching the PurchaseOrder.exe. The executable has a PDF document icon and the user can be forgiven for not recognizing it as an executable, considering the fact that Windows by default hides filename extensions of the known file types.

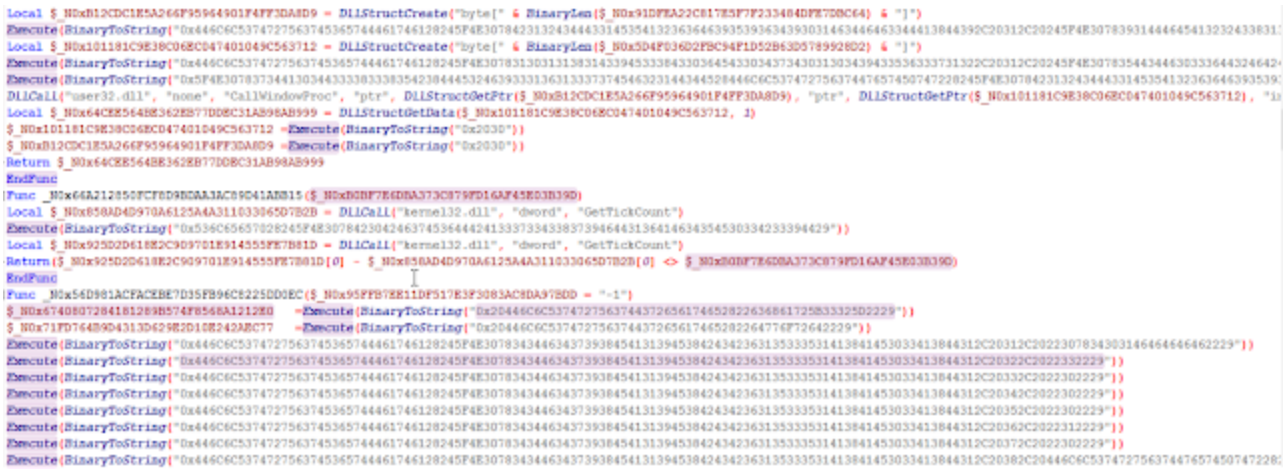
Icon file used by PurchaseOrder.exe.

The executable itself is just over 1.4MB in size, which is rather large for attachments used in email campaigns. The file itself is a self-extractable CAB archive which contains three randomly named files.

Stage 3 - AutoIt Script

The first file is instantly recognisable and it is a legitimate, Autoit script interpreter. The second file is a Unicode file encoded as UTF-16 and is over 110MB in size which is at first almost enough to discourage from analysis. The actual script code starts deep within the file, which provides the attacker with the ability to obfuscate the script code in a way that is not immediately visible by researchers.

Talos has already written about a similar delivery method in the past and it seems that this campaign uses a similar generator of obfuscated Autoit scripts. Thankfully, It was relatively simple to remove all the junk characters and reduce the size of the code to analyze to a much more manageable 41KBs.



Autoit stage deobfuscated

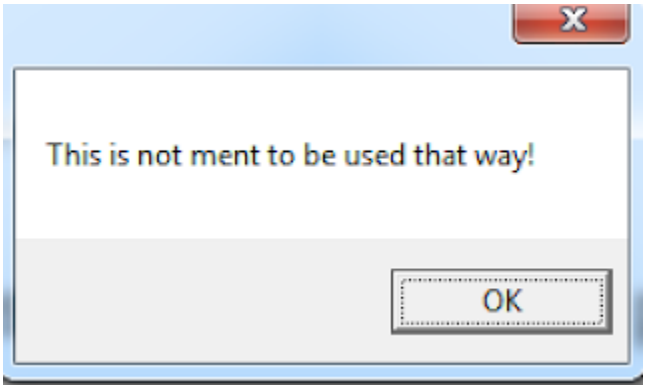
The Autoit script itself creates a directory in the user's profile folder and sets its attributes to system and hidden. It then creates a copy of RegSvcs.exe .NET services installation tool or copies the existing RegSvcs.exe to a filename splwow64.exe to set up the next stage. Regscvcs.exe is used for injecting and launching a remote thread within its process space. The thread uses RC4 to decrypt the third file dropped by the original self-extractable CAB archive and reads it into the process space of regsvcs.exe. This leads us to the next stage, using an executable developed in C/C++. This stage will only exist in its executable format in memory, while it will be an RC4 encrypted data blob on the disk.

Stage 4 - Zyklon injector

The stage injected into RegSvcs.exe is another unobfuscated injector of the final payload. The executable decompresses the payload from the resource section of the PE file, finds and launches Windows Explorer executable that is found in different folders depending on the Windows platform (32 or 64 bit) and launches a remote thread that loads and runs a .NET executable, which is the final payload of the campaign, in this case a sample of the Zyklon HTTP bot.

Loading managed code into an unmanaged space is not entirely simple process. Attempting to cheat the infection chain to launch the Zyklon bot from the command line was apparently anticipated by the campaign author who modified the Zyklon class Main function to display a text message for anybody trying to launch it this way.

The original Zyklon code for the version 1.0.0.0 does not seem to contain this mechanism that ensures that the payload is run by a specific loader that does not call the Zyklon Class Main function but a different entry point.



You are not supposed to run it this way

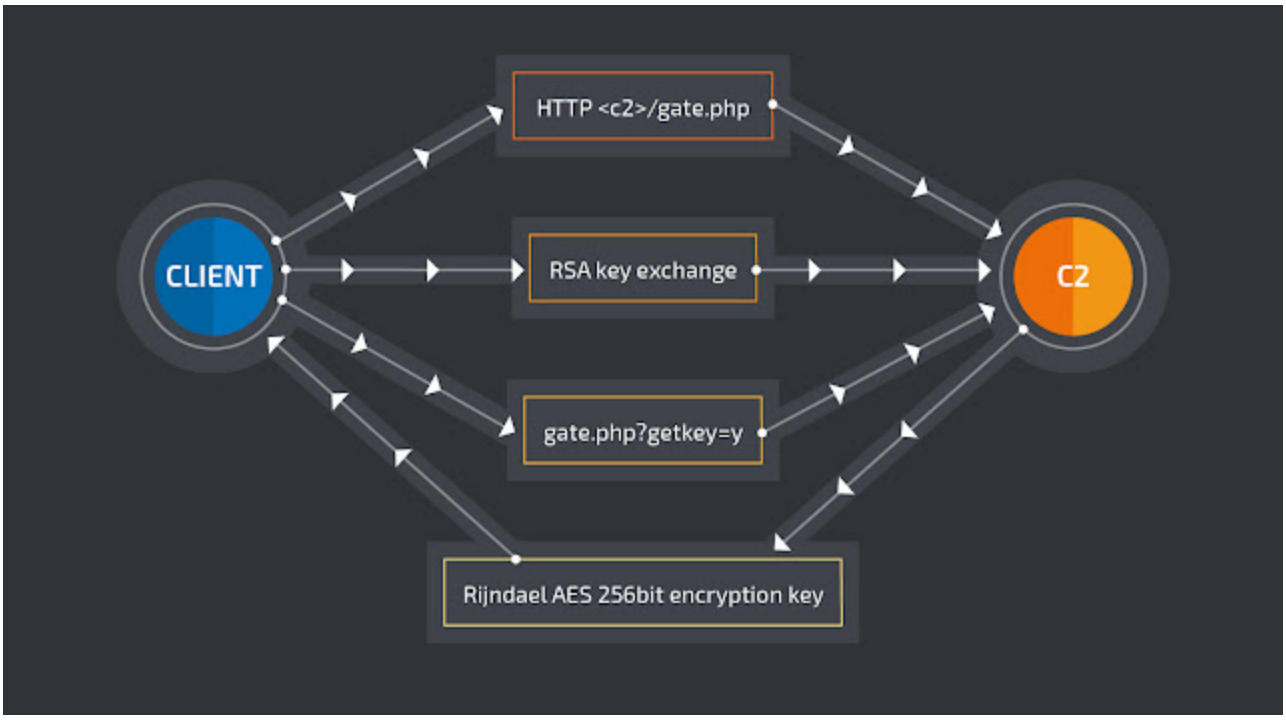
The payload is obfuscated using Crypto Obfuscator and an additional code generator. The code which uses xor operations to set a value of a variable used in a switch statement to direct the program flow is relatively easy to follow once the Crypto Obfuscator code

transformations are removed, which can be done using a very useful .NET deobfuscation utility de4dot. In fact, the Zyklon Builder, found on VirusTotal, uses the same dnlib library, used also by de4dot and dnspy analysis tools, to add the configuration file to the malicious .NET assembly base Zyklon bot embedded in its resource section.

Once the obfuscator was removed it did not take too long to realize that for the purpose of the analysis it was possible to manually modify the Zyklon class Main function to call the EntryPoint function which contains the bot code and debug the Zyklon using the dnspy debugger.

C2 communication (encryption)

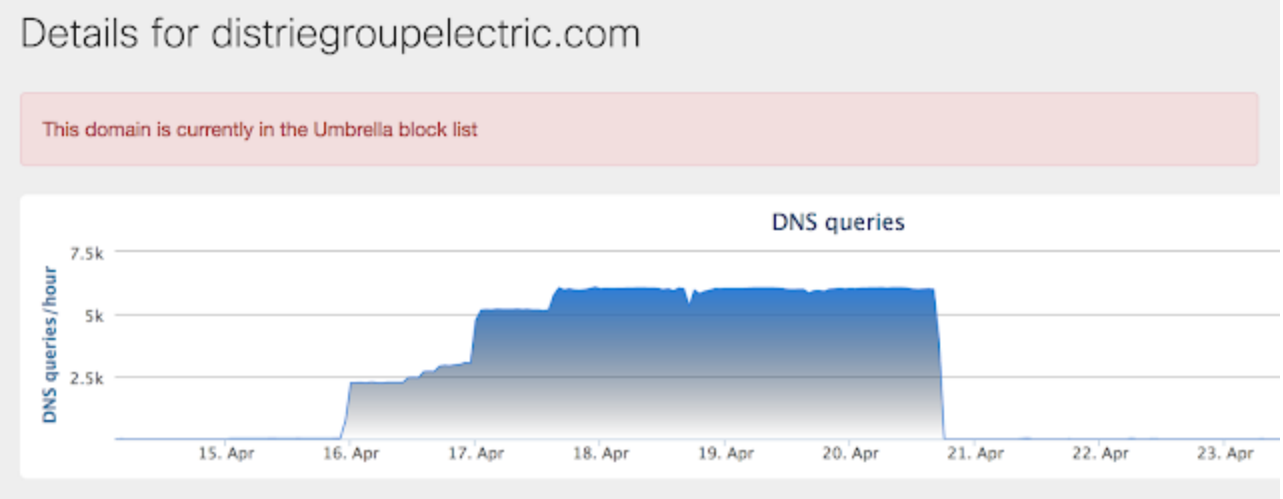
Zyklon's "official" name is "Zyklon H.T.T.P Bot", which is visible in the links to PDB files retained as a remainder of the compilation process. The bot is reasonably well written with precautions for hiding the traffic from network based detection engines, even from intercepting proxies by encrypting all its communications.



Establishing communication with a C2 server

The bot connects to one of the three possible C2 servers, starting from the first one specified in its configuration. The server sends a certificate and the communication is first encrypted with RSA and then with a 256 bit long AES with the initialization vector and the key generated by the server, sent back to the client after the client POSTs a request ending with the query gate.php?getkey=y.

Looking at the DNS requests for one of the C2 servers that remained active throughout the campaign it is possible to see the time when the campaign was active.



C2 DNS domain activity

The initial configuration for the bot is embedded within the resource section of the file, together with the list of user agent strings used by the bot when contacting the C2 server. The malicious .NET assembly also contains an encrypted blob that becomes its persistence module injector. Once decrypted and loaded in memory its function is to make sure that the bot is respawned from a remote thread if the main executable is terminated as a process.

The client then sends a request containing the information about the infected system and receives a configuration string from C2, which sets the internal bot parameters. Several threads are also launched in order to download and execute required additional plugins.

The main command loop sleeps for 60 seconds and sends a request for a command to the C2 server. The main purpose of the bot seems to be conducting DDoS attacks but there are other more or less standard commands available such as downloading and executing additional payloads from a user-specified URL or logging the user keystrokes and sending them back to the C2 server.

Curiously, Zyklon may also attempt to enumerate the usual automatic startup locations in the Windows registry to find potential competitive files and submit them to VirusTotal for scanning. So called cloud malware inspection is used to terminate processes based on the VirusTotal verdict. The bot also executes rudimentary heuristic checks for some of the known competitive bot names and filename extensions and tries to remove them if found on the system. Competition is never welcome by the bad guys.

Zyklon website

The website advertising Zyklon is hosted on a .onion domain which is also accessible from

the clear net through a web to Tor proxy. The owners are advertising two different versions for sale, one that can connect to Tor based C2 servers and the standard one without that capability.

Perhaps the most interesting page of the Zyklon website are its Terms of Service, which the authors seem to believe may free them from potential prosecution. The user, aka the attacker, allegedly has the sole legal responsibility for damage caused by it, at least according to Zyklon creators :

```
YOU UNDERSTAND AND HEREBY ACKNOWLEDGE AND AGREE THAT YOU MAY NOT AND WARRANT THAT YOU WILL NOT:

1. use the Zyklon H.T.T.P Remote Administration Software for any illegal purpose, or in violation of any laws, including, without limitation, laws governing intellectual property, data protection and privacy, and import or export control;

2. remove, circumvent, disable, damage or otherwise interfere with security-related features of the Zyklon H.T.T.P Remote Administration Software, features that prevent or restrict use or copying of any content accessible through the Zyklon H.T.T.P Remote Administration Software, or features that enforce limitations on use of the Zyklon H.T.T.P Remote Administration Software;

3. intentionally interfere with or damage operation of the Zyklon H.T.T.P Remote Administration Software or any user's enjoyment of them, by any means, including uploading or otherwise disseminating viruses, worms, or other malicious code;

4. post, store, send, transmit, or disseminate any information or material which infringes any patents, trademarks, trade secrets, copyrights, or any other proprietary or intellectual property rights;or

5. Install and/or use Zyklon H.T.T.P Remote Administration Software on any computer which you do not have explicit permission to do so on;

6. distribute Zyklon H.T.T.P files over the Internet with the intent of infecting/harming machines of other people;
```

Downloaded credential harvesting modules (email, browser, ftp)

Zyklon creators also advertise a number of useful plugins for harvesting user credentials and stealing confidential information such as details of wallets of various crypto currencies like Bitcoin, LiteCoin and DodgeCoin. For a potential customer, the list of features must be quite impressive. However, not everything is as ideal as it seems at first.

In the analyzed campaign, the Zyklon main executable downloaded only three plugins, as instructed by the C2 server, all of them with a purpose of stealing user credentials from password caches of the most popular web browsers as well as email and ftp clients.

```
CI=False|KT=1|UAC=False|S5=False|ER=False|UPNP=False|RP=True|RW=False|
AK=False|BK_CYCLE=|BK_RUN_ONCE=False|SOCKS_PORT=3128|SOCKS_AUTH=False|
SOCKS_USERNAME=Nothing|SOCKS_PASSWORD=Nothing|KLI=1|KLM=500|EKL=True|
WC=False|BA=MyBtc|LA=MyLtc|KLF=False|BR=True|FTR=True|EMR=True|SFR=False|
GR=False|AU=False|UF=N/A|
```

Configuration command sent to Zyklon from C2 server

The plugin download URL follows the format of plugin/index.php?plugin=<pluginname> with possible plugins being

```
/plugin/index.php?plugin=browser
/plugin/index.php?plugin=email
/plugin/index.php?plugin=ftp!
/plugin/index.php?plugin=software
/plugin/index.php?plugin=games
/plugin/index.php?plugin=cuda
/plugin/index.php?plugin=minerd
/plugin/index.php?plugin=sgminer
/plugin/index.php?plugin=socks
```

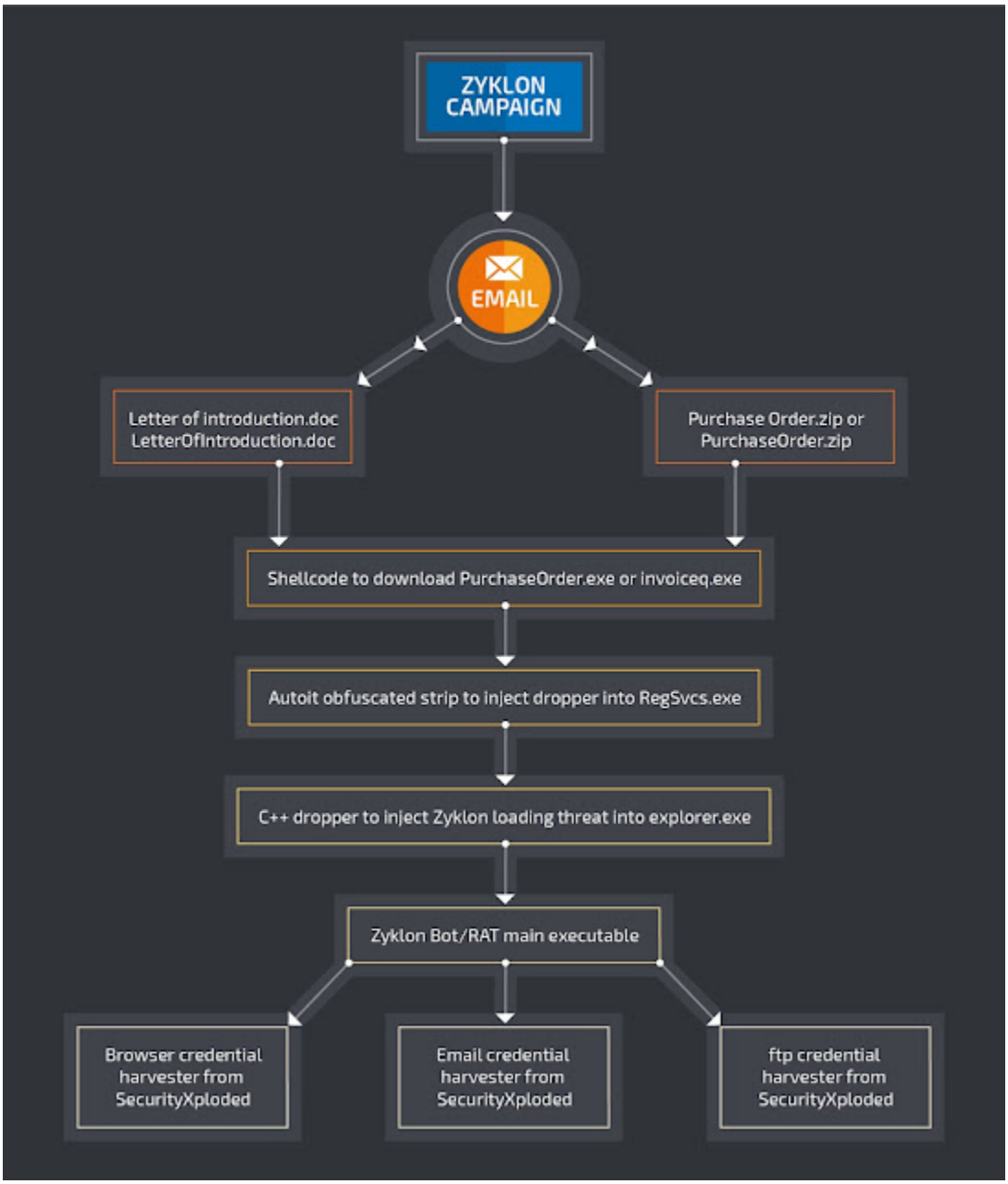
Available Zyklon plugins

Downloaded plugins are injected into a previously launched and hollowed legitimate process name "%windir%\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe" and are in fact just freeware command line tools written in C/C++ available from the website http://www.securityxploded.com.

It is likely that Zyklon author realized it would take quite a long time to fully develop all the features within the main Zyklon bot and decided to include available free password dumping utilities just to make its RAT more competitive in what is quite a cutthroat underground market for remote administration tools.

Conclusion

Zyklon is quite a well known botnet kit and it has been fairly active this year. In this smaller, possibly more targeted campaign we analyzed, it has shown that its users are employing a number of different technologies and obfuscation techniques to be more successful - from exploiting a vulnerability in Microsoft Word over Autoit scripts and .NET executables, all the way to freeware utilities used as plugins for harvesting credentials from browser cache, email and ftp clients.



Zyklon campaign execution flow on an endpoint

Overall, this was a well executed campaign which used compromised hosts as C2 servers. Luckily, there are several weaknesses which can be exploited for detecting its footprint either by inspecting IOCs or tracking the network communications patterns and behavior on endpoints.

Coverage

PRODUCT	PROTECTION
AMP	✔
CloudLock	N/A
CWS	✔
Email Security	✔
Network Security	✔
Threat Grid	✔
Umbrella	✔
WSA	✔

Advanced Malware Protection (AMP) is ideally suited to prevent the execution of the malware used by these threat actors.

CWS or WSA web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such as NGFW, NGIPS, and Meraki MX with Advanced Security can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella prevents DNS resolution of the domains associated with malicious activity.

Stealthwatch detects network scanning activity, network propagation, and connections to CnC infrastructures, correlating this activity to alert administrators.

Iocs

Document exploits
ac944374d5f50ecbdd3b9e7151d5a4b055ec18ea26482c2301ccc439164b25be
996b19658cffedc9395243693c3ca1d12a2c2a2c986e35a877f1ae2a2b595a6d

PE Exes downloaded by the exploit docs
4bce73a29ee1b9840cd82d8c08e107179cd74dc1aed488f6d16772ce12092c69
bcf8dbbc78883b2d84511819123cf39b1c2ffe3cd9763d08fe1544c89084cadf

ZIP Attachments
e67db2e2ebd3c540489dd4844b066b45f31b2d879a085eabda1f63926ddc0688
b1906c1d23f62df7f63a06030f27c3249414d027a9deb62d27f65ec6f3a61adb

PE exe files within ZIPs
b7101462507a8cf5bf91b62b641ef1ac3d268115d6dfca54a1625efb07fccf0d
4bce73a29ee1b9840cd82d8c08e107179cd74dc1aed488f6d16772ce12092c69

Browser plugin
e5d2c3a7ddd219ab361af4a709999a492387e3aaf8380187a7699895fc383e40

FTP plugin
6a32a0d83a5c955822502444833283a3fde8e1893f1490fac1ae5b84a00db5c6

Email plugin
bbcc07baaa00bb30de43a39a04dc66754fe805630f155fde47ab259fdbd03748

Zyklon Builder v1.0.0
682d5d60d6fc0e1d5810e9cd9d8b1c6b6fa154d5a790da944177074d28846d66

Download URLs
http://wszystkozmetalu[.]pl/Invoiceq.exe
http://www.blcpolychemical[.]com/re/PurchaseOrders.exe
http://barkliaytire[.]com
http://distriegroupelectric[.]com
http://extreime-net[.]com
http://distriegroupelectric[.]com:80/plugin/index[.]php?plugin=ftp
http://distriegroupelectric[.]com:80/plugin/index[.]php?plugin=email
http://distriegroupelectric[.]com:80/plugin/index[.]php?plugin=browser

C2

http://distriegroupelectric.com:80/gate[.]php
http://distriegroupelectric.com:80/login[.]php - Control Panel